

004
П 523

004.9

Компьютер **на 100%**

Умные книги для умных людей!

Олег Поломошнов

Быстро и легко

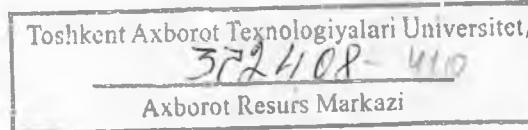
создаем,
программируем
и раскручиваем

Web-сайт

2032438

42809

392
АКН(2)



Москва ЭКСМО 2011

УДК 004.42
ББК 32.973
П 52

004.42

штурвалы

В коллаже на обложке использованы иллюстрации:
Tuiulijumala / Shutterstock.com

П 52

Поломошнов О.

Быстро и легко создаем, программируем и раскручиваем Web-сайт /
Олег Поломошнов. — М. : Эксмо, 2011. — 352 с.

Сегодня мало кто из нас может представить свою жизнь без Интернета, где есть новости, погода, социальные сети, а также масса полезной информации о людях и событиях. Но многим пользователям недостаточно просто лазить по ярким сайтам, им хочется оставить свой след в Сети – создать свой сайт. Если вы один из таких пользователей, то эта книга поможет вам в ваших начинаниях. В нашем самоучителе вы сможете найти все необходимые знания для каждого этапа работы над собственным сайтом – начиная от идеи и заканчивая раскруткой своей страницы. Ключевые аспекты создания сайтов – планирование, выбор хостинга, программирование, создание продуманного дизайна – подробно описаны в этой книге.

УДК 004.42
ББК 32.973

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фоторепродукцию и запись на магнитный носитель, если на это нет письменного разрешения ООО «Издательство «Эксмо».

Поломошнов Олег Владимирович

**БЫСТРО И ЛЕГКО СОЗДАЕМ, ПРОГРАММИРУЕМ
И РАСКРУЧИВАЕМ WEB-САЙТ**

Ответственный редактор *В. Обручев*
Художественный редактор *Н. Биржаков*

ООО «Издательство «Эксмо»
127299, Москва, ул. Клары Цеткин, д. 18/5. Тел. 411-68-86, 956-39-21.
Home page: www.eksмо.ru E-mail: info@eksмо.ru

Подписано в печать 23.08.2011.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 28,52.
Тираж 2000 экз. Заказ 7336.

Отпечатано с готовых файлов заказчика
в ОАО «Первая Образцовая типография»,
филиал «УЛЬЯНОВСКИЙ ДОМ ПЕЧАТИ»
432980, г. Ульяновск, ул. Гончарова, 14

ISBN 978-5-699-51005-4



9 785699 510054 >

ISBN 978-5-699-51005-4

© Поломошнов О. В., 2011
© ООО «Айдиономикс», 2011
© Оформление. ООО «Издательство «Эксмо», 2011

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
ГЛАВА 1. ВВЕДЕНИЕ В ИНТЕРНЕТ И ДОМЕННЫЕ ИМЕНА	8
1.1. Модель OSI/ISO, протоколы и другие непонятные слова	8
1.2. Адресация в Интернете	11
1.3. Регистрация доменного имени второго уровня	13
ГЛАВА 2. ПЛАНИРОВАНИЕ САЙТА	16
2.1. Какие вопросы следует задавать	16
2.2. Визуализация — рисуем несколько страниц сайта	20
ГЛАВА 3. БЫСТРО И КАЧЕСТВЕННО СТРОИМ САЙТ И РАЗМЕЩАЕМ ЕГО НА БЕСПЛАТНОМ ВЕБ-СЕРВЕРЕ	21
3.1. Основные понятия: коротко и быстро	21
3.2. Создание основы сайта: структура, форматирование, наполнение	22
3.3. Списки и таблицы в теле веб-страницы	31
3.4. Организация гиперссылок	38
3.5. Графические объекты на веб-странице	42
3.6. Размещение сайта	46
3.7. Введение в каскадные таблицы стилей CSS	56
ГЛАВА 4. FTP-КЛИЕНТ, ЗАГРУЗКА ФАЙЛОВ НА СЕТЕВЫЕ СЕРВИСЫ	68
4.1. Знакомство с FTP-клиентом	68
4.2. Установка и настройка FileZilla	70
4.3. Размещение файлов сайта на удаленном компьютере с помощью FileZilla	73

ГЛАВА 5. ПРОГРАММИРОВАНИЕ НА JAVASCRIPT	77
5.1. Знакомство с языком	77
5.2. Простейшие сценарии.....	80
5.3. Формы	84
5.4. Более сложные сценарии.....	92
5.5. Функции	97
5.6. Сценарии из библиотек сторонних разработчиков	101
5.7. Обзор библиотеки jQuery.....	105
ГЛАВА 6. ГРАФИКА САЙТА	127
6.1. Основные понятия	127
6.2. Кисти, краски и холсты — обзор инструментов для подготовки веб-графики.....	131
6.3. Начинаем рисовать. Создание графических объектов для веб-страниц.....	133
6.4. GIF-анимация. Создание главного изображения сайта.....	141
6.5. Основы фотомонтажа	150
6.6. Часто возникающие вопросы и ответы на них	158
ГЛАВА 7. ТЕХНОЛОГИЯ FLASH	163
7.1. Введение во Flash-анимацию	163
7.2. Построение изображений в слоеном пироге	170
7.3. Как научить летать. Переход на веб-страницы	182
ГЛАВА 8. DREAMWEAVER – ГЛАВНЫЙ ИНСТРУМЕНТ ВЕБ-МАСТЕРА	188
8.1. Да здравствует прогресс в области сайтостроения!	188
8.2. Сайт не на коленке	195
8.3. Таблицы и графика	201
8.4. Форматирование текста. Организация гиперссылок.....	208
8.5. Тиражирование веб-страниц	213
8.6. Игра со сценариями на веб-страницах	215

8.7. Доделки.....	223
8.8. Работа с хостингом штатными средствами.....	225
ГЛАВА 9. CMS JOOMLA!	230
9.1. О системах управления содержимым	230
9.2. Установка и настройка.....	238
9.3. Структура	248
9.4. Первичные действия по созданию сайта на основе CMS	252
9.5. Расширения	281
9.6. Смена внешнего вида сайта.....	286
9.7. Добавление нетекстовых объектов на веб-страницу.....	302
9.8. Подключение дополнительных расширений к сайту	308
9.9. Что необходимо знать веб-мастеру при управлении сайтом на основе CMS	320
ГЛАВА 10. О ПРОДВИЖЕНИИ САЙТОВ И ИХ РЕГИСТРАЦИИ В ПОИСКОВЫХ СИСТЕМАХ.....	327
10.1. Раскрутка и продвижение сайта	327
10.2. Поисковые системы и регистрация в них.....	331
10.3. Каталоги и регистрация в них.....	335
10.4. Рейтинги.....	336
10.5. Баннерная реклама	337
10.6. Влияние интерактивности сайта на привлечение посетителей	338
10.7. Это волшебное слово SEO	339
ГЛАВА 11. ОБЗОР СЕРВЕРНЫХ ТЕХНОЛОГИЙ.....	349
11.1. Что дальше.....	349
11.2. Язык программирования PHP	349
11.3. ASP-технология.....	351

ВВЕДЕНИЕ

В наше бурное информационное время, когда Интернета становится больше, чем телевидения, у окружающих меня людей все чаще возникает потребность организовать свое представительство в Глобальной сети. Когда подобный разговор ведет предприниматель или руководитель какой-либо бюджетной организации, то обычно поднимаются такие вопросы: «Что надо? Сколько стоит? Как быстро будет создано? Как часто планируется обновлять?» Частные заказчики чаще спрашивают: «Как сделать, потому что нас обязывают это иметь?», на что я советую просто создать свой блог. Они отвечают: «Спасибо, нет, нам нужен именно сайт». В таком случае я даже не знаю, что и сказать.

Появление подобной книги было вопросом времени, и я подумал, почему бы мне не передать другим свой скромный опыт в обучении построению сайтов. Как результат, я оформил в виде книги практические приемы и рекомендации, которые прежде всего необходимы неофиту в постижении глубоких тайн сайтомстроения. На самом деле нет ни тайн, ни секретов, а есть только усидчивость и огромное желание создать. Однако помимо желания необходимо иметь еще что-то, что придется выносить на суд общественности, то есть достойное наполнение сайта — его контент. В том же случае, когда сайт служит лишь для ознакомительных целей, требования к оригинальности содержимого можно рассматривать по другим критериям.

На страницах этой книги описаны некоторые приемы построения буквально с нуля или, можно сказать, с состояния «нарисовано на коленке» сайта фирмы «Сатурн», специализирующейся на организации космических путешествий в пределах Солнечной системы.

К сожалению, а может быть и к счастью, эпоха статических сайтов, которые уже считаются классическими, подходит к концу и все большую часть пространства сетевых ресурсов занимают динамические сайты, где страницы генерируются автоматически, а контент хранится в базах данных на отдельных серверах. По этой причине одна из глав книги посвящена наиболее распространенной в данное время системе управления содержимым, написанной на языке PHP и распространяемой на свободной основе, — Joomla!. Это, конечно, не делает книгу справочником по Joomla!, но дает пользователю представление о том, какой способ можно избрать для существования сайта.

При этом ни в коей мере не принижается классический способ создания сайтов, написанных на языке гипертекстовой разметки HTML, методы использования и основные понятия которого также приведены в одной из глав. Затронута тема оживления веб-страниц путем размещения на них сценариев — маленьких программ, исполненных на языке JavaScript. Помимо этого, я постарался дать краткий обзор расширенных возможностей веб-мастера, используемых при подготовке веб-страниц, таких как каскадные таблицы стилей CSS или подключаемые библиотеки сценариев jQuery. Безусловно, каждая из затронутых тем заслуживает отдельного фундаментального труда. И все же я надеюсь, что краткое представление об этих возможностях у вас будет сформировано. В книге также рассмотрены способы размещения файлов веб-страниц на удаленном сервере с помощью FTP-клиента.

Перед тем как приступить к созданию сайта, необходимо подумать, зачем он вам, каким вы хотите его видеть и как вы себе представляете его в глазах пользователей. Краткие рекомендации, которые помогут вам ответить на эти вопросы, также приведены в одной из глав. А что делать, когда сайт уже создан? И на этот вопрос вы получите ответ.

Глава 1

ВВЕДЕНИЕ В ИНТЕРНЕТ И ДОМЕННЫЕ ИМЕНА

1.1. Модель OSI/ISO, протоколы и другие непонятные слова

Интернет в наше время стал настолько повседневным, что чаще всего мы совсем не задумываемся, как это работает. Ведь никто не знает, что такое электричество, никто не видел электроны, которые согласно школьному учебнику физики якобы бегают по проводам и создают электрический ток, — нам достаточно нажать кнопку, и появляется свет или нагревается чайник. Чтобы найти информацию в Интернете, достаточно к нему подключиться и ввести запрос в строке поисковой машины, а для этого незачем разбираться, почему самолеты крыльями не машут, когда летают. Но для человека, желающего осчастливить человечество своим нетленным творением в виде нового сетевого информационного ресурса, причем созданного самостоятельно, необходимо иметь хотя бы минимальное представление о том, как ларчик открывается.

Итак, сама сеть — это просто провода либо другие каналы связи (оптические, радио и пр.), по которым осуществляется передача электрических импульсов, несущих информацию. Согласно базовой сетевой модели, принятой Международным институтом стандартизации ISO, взаимодействие между компьютерами при обмене информацией можно свести к семи предлагаемым уровням (табл. 1.1).

Таблица 1.1. Схема Интернета согласно эталонной модели для взаимодействия открытых систем

Тип данных	Уровень	Функции
Данные	Прикладной	Доступ к сетевым службам
	Представлений	Представление и кодирование данных
	Сеансовый	Управление сеансом связи
Сегменты	Транспортный	Прямая связь между конечными пунктами и надежность
Пакеты	Сетевой	Определение маршрута и логическая адресация
Кадры	Канальный	Физическая адресация
Биты	Физический	Работа со средой передачи, сигналами и двоичными данными

Уровень первый — физический

Этот уровень представляет собой то, что обычно любят показывать в фантастических фильмах, когда герой путешествует внутри сети. На самом же деле под физическим уровнем понимают перемещение по каналам связи сигналов той природы, для которых эти каналы предназначены. Например, когда вы работаете в интернет-кафе, запрос с вашего компьютера сначала превращается в радиоволну, уходя на точку доступа Wi-Fi, а затем, преобразовавшись в световой лучик, устремляется по оптоволоконному кабелю к серверу провайдера, где превращается в электромагнитную волну и бежит дальше по обычным проводам.

Уровень второй — канальный

Как видно из изменений различной природы сигнала, рассмотренной на физическом уровне, где-то должна быть фея, которая мановением волшебной палочки превращает радиоволну в луч света или наоборот. Вот этой феей на втором уровне и являются различные устройства, модемы, конвертеры и прочие преобразователи природы передаваемого сигнала. Именно они, получая на входе электрический импульс, на выходе выдают радиоволну. Таким образом, только благодаря этим маленьким, но чрезвычайно важным переводчикам различные устройства, например ваш телефон и компьютер, понимают друг друга.

Уровень третий — сетевой

На сетевом уровне безликие сигналы начинают группироваться в пакеты, имеющие одно лицо на всех, строго заданный размер и закрепленный за ними заголовок, определяющий, от какого компьютера данный пакет отправлен и какому конкретно компьютеру предназначен. Именно на третьем уровне указывается адрес для конкретного сигнала, несущего информацию; здесь начинает работать первичный протокол IP, который определяет адрес, куда направляется данный пакет. Страшное слово «протокол», употребляемое в контексте сетей, определяется как способ взаимодействия различных частей системы, то есть договоренность между производителями устройств о характеристиках и параметрах сигналов, которыми эти устройства обмениваются. Если говорить еще проще, то это договоренность о создании одного языка, понятного всем устройствам данного типа.

Чтобы протокол IP (Internet Protocol) отправил пакет по необходимому адресу, этот адрес должен существовать. Беда только в том, что в Сети нет городов и поселков и с адресом на деревню дедушке ничего не получится. Для передачи информационного пакета от компьютера к компьютеру была придумана так называемая IP-адресация, то есть каждый компьютер при подключении к Интернету получает на время сеанса динамический либо

на все время статический IP-адрес. Такой адрес представляет собой простую подборку чисел, лежащих в диапазоне от 0 до 255 и разделенных точками на четыре группы для протокола версии TCP/IPv4 или на шесть групп для версии TCP/IPv6. Например, 87.250.251.3 – IP-адрес сайта ya.ru.

Конечно, из более чем четырех миллиардов адресов для версии TCP/IPv4 не все разрешены – отдельные зарезервированы для локальных либо виртуальных сетей, внутри которых могут быть свои правила адресации. Однако в любом случае подключение отдельных локальных сетей к Интернету производится через шлюз, который и приводит локальную адресацию в соответствие с общепринятой.

Уровень четвертый – транспортный

Можно сказать, что транспортный уровень является пограничным, когда пакеты еще не превратились в файлы. На четвертом уровне отправляемые файлы с данными нарезаются на пакеты, а к ним добавляется контрольная сумма, либо наоборот – полученные пакеты объединяются в файлы. Благодаря четвертому транспортному уровню один монолитный файл разбивается на группу маленьких пакетов, каждый из которых отправляется к адресату своим собственным путем. Может оказаться так, что пакет, содержащий информацию конца файла, придет к адресату раньше, чем пакет, содержащий его начало. Но пользователь этого не ощущает, поскольку именно на транспортном уровне все пакеты проверяются на целостность и исправность путем пересчета закрепленных контрольных сумм, в случае несоответствия которых выполняется повторный запрос. Обеспечивать работу транспортного уровня призваны два протокола. Первый – TCP-протокол управления передачей, при использовании которого полученные пакеты проверяются на исправность, а при необходимости выполняется повторный запрос на пересылку битого пакета. Второй – UDP – универсальный протокол передачи данных, так называемый потоковый протокол, который не запрашивает повторные пакеты при их несоответствии контрольным суммам и поэтому используется в основном при потоковой пересылке данных, например для просмотра на компьютере телепередачи в реальном режиме времени.

Уровень пятый – сеансовый

На пятом уровне, как можно представить из названия, определяются параметры сеанса связи между «договаривающимися» сторонами. На этом уровне информация из электрических сигналов и пакетов превращается в данные, с которыми уже работают приложения более высокого уровня. Здесь определяется начало или завершение сеанса связи, синхронизация выполняемых задач, присвоение прав на передачу, а также обязанность поддержания сеанса связи в то время, когда использующие его приложения неактивны.

Уровень шестой – представлений

На шестом уровне – уровне операционной системы – происходит декодирование полученных из Сети данных в форматы, понимаемые приложениями, или, наоборот, кодирование файлов, с которыми работают приложения, в данные для последующей обработки к пересылке. Например, это сжатие или распаковка, перенаправление к другому информационному ресурсу либо приложению, то есть веб-страница, содержащая Flash-ролик, на этом уровне получит поддержку от Flash-проигрывателя. Обычно подразумевают, что уровень представлений обеспечивает собой промежуточный протокол преобразования информации для смежных уровней. Именно благодаря ему мы можем «порхать» по информационным ресурсам на компьютерах под управлением различных операционных систем или открывать одну и ту же веб-страницу на одном и том же компьютере, но в окнах различных браузеров, чтобы посмотреть, как она там будет выглядеть.

Уровень седьмой – прикладной

На седьмом уровне мы непосредственно работаем с прикладными программами, обеспечивающими наши запросы. Это браузер, позволяющий работать с веб-страницами, почтовые клиенты, радующие нас письмами от друзей, FTP-клиенты, с помощью которых мы можем получать и отправлять файлы, а также огромное множество других программ.

Поскольку каждая из программ говорит на своем языке, то он и обеспечивается собственным протоколом, специфичным для данного приложения. Нам для работы по созданию и обслуживанию сайтов достаточно знать всего два протокола, необходимых для работы с веб-страницами. Первый – HTTP – протокол передачи гипертекста, именно на его основе передается все многообразие содержимого веб-страницы, обеспечиваемое службой WWW. Второй – FTP – протокол передачи файлов, используется для передачи файлов по Сети с помощью службы FTP.

Таким образом, самый верхний прикладной уровень по модели OSI как раз и представляет собой уровень пользователя – человека, то есть именно на этом уровне мы взаимодействуем с Сетью через приложения.

1.2. Адресация в Интернете

Поскольку про модель построения Интернета все относительно понятно, я расскажу, почему нет необходимости знать IP-адрес компьютера при обращении к нему, а всего лишь достаточно указывать имя сайта.

Из модели OSI, рассмотренной выше, видно, что для связи каждый компьютер должен обладать своим адресом, который выражается четырьмя или шестью байтами, в зависимости от версии протокола. Поэтому для

определения адреса компьютера, на котором расположен интересующий нас информационный объект, наш компьютер должен обратиться к специально выделенному серверу — маршрутизатору. Он, в свою очередь, примет на себя работу по просчету маршрута пакетов между нашими компьютерами, осуществляющими связь, и перешлет эти пакеты. Но такая связь в любом случае требует знания номера компьютера, который очень трудно, да и не нужно удерживать в голове. Перед тем как обратиться к маршрутизатору и указать ему IP-адрес нашего визави, мы обратимся к так называемому DNS-серверу. Эти серверы осуществляют работу Системы доменных имен, которая приводит в соответствие название компьютера с его номером — IP-адресом. Иначе говоря, это специальная программа-сервер, сверяющая по таблице доменное имя компьютера с его номером в Сети и возвращающая нашему компьютеру вместо www.ya.ru IP-адрес 87.250.251.3, с которым мы затем и обращаемся к маршрутизатору.

Что же такое доменное имя? Все дело в том, что вся зона Интернета не является сплошной, она разбита на подзоны — так называемые домены. Каждый домен определяет принадлежность информационных ресурсов к той или иной категории. Это сложилось исторически, когда Глобальная сеть, разрастаясь, вышла за пределы государственных учреждений Америки. На данный момент доменные имена в принципе уже не имеют узких политических границ, то есть, физически находясь в России, можно зарегистрировать свой сайт в домене, например, Великобритании, ведь вопрос только в деньгах.

Доменные имена, в свою очередь, принято подразделять на уровни, например домен первого уровня `ru`, который показывает принадлежность всех информационных ресурсов к зоне, отведенной России. Доменное имя второго уровня `yandex.ru` будет показывать, что информационный ресурс «Яндекс» зарегистрирован в зоне Рунета. И, соответственно, надпись `narod.yandex.ru` для DNS-сервера будет означать следующую инструкцию: обратиться в зону `ru`, найти сервер с именем `yandex`, на котором в сетевом доступе расположена папка `narod`, а в ней запустить по умолчанию файл `index.html`. Таким образом, чем больше вложенность по уровням в доменном имени, тем более длинным, а значит, и более неудобным для запоминания оно является, а тут ведь и протоколы указывать надо, благо они автоматически браузерами вбиваются. Иными словами, полное имя, например, создаваемого нами сайта, расположенного на бесплатном хостинге `narod.ru`, будет выглядеть следующим образом: <http://www.saturnkom.narod.ru/>, где:

- `http://` — указание на то, какой протокол использовать для предоставления информации;
- `www.` — указание на то, что это веб-страница и ее обслуживание осуществляется службой WWW;
- `saturnkom.` — имя папки, в которой расположены файлы нашего сайта;

- *narod.* — имя веб-сервера, предоставляющего нашу папку для доступа в Сети;
- *ru* — имя домена, показывающего то, что все это безобразие находится в русскоязычной зоне Интернета.

Для более успешного продвижения сайтов лучше выбирать короткое, запоминающееся имя, и желательно, чтобы оно соответствовало по звучанию своему домену первого уровня, например *mail.ru*. Если нет желания платить хоть небольшие, но все же деньги за доменное имя второго уровня, то можно воспользоваться бесплатным хостингом. При этом имя сайта будет иметь третий или больший уровень вложенности и, как следствие, длинное доменное имя, которое присваивается автоматически при регистрации на сайте хостинга, но об этом я расскажу в следующих главах. Сейчас просто рассмотрим процедуру получения доменного имени второго уровня.

1.3. Регистрация доменного имени второго уровня

Поскольку все домены первого уровня делятся на две большие части: домены общего пользования и национальные домены, необходимо выбрать то доменное поле, в котором вы хотите быть зарегистрированы как владелец. Поэтому следует достаточно внимательно отнестись к следующей информации.

К доменам общего пользования относятся домены *com*, *net*, *org*, *info*, *biz*, *museum*, *name*, *aero*, *coop* и *pro*. К ним также можно отнести домены ограниченного пользования, регистрация в которых проводится на основе фактической принадлежности организации к тому или иному виду деятельности. Это домены *int*, выделенные для международных организаций, *edu* — для высших учебных заведений США, *GOV* — для правительственные организаций США и *mil* — для военных ведомств США.

В свою очередь, национальные домены — это двухбуквенные домены первого уровня, которые назначаются международной организацией IANA (Internet Assigned Numbers Authority) по согласованию с интернет-сообществами стран и в соответствии с кодами стран и территорий по международному стандарту ISO 3166-1. В настоящее время в мире насчитывается 243 национальных домена, среди которых *ru*, *su*, *ua*, *de*, *us* и многие другие.

Домен *ru* является национальным доменом Российской Федерации, и, если ваш сайт ориентирован на Россию, домен в зоне *ru* более всего подходит для адреса сайта.

Доменные имена регистрируются сроком от одного года, и плату за эту процедуру и последующее продление доменного имени взимает регистратор, поэтому регистрировать доменное имя лучше сразу при оформлении

договора на хостинг платного размещения сайта. В этом случае при оформлении платного хостинга компания-провайдер обычно не только берет на себя услуги по оформлению регистрации домена, но и дает определенные бонусные скидки на оформление сайта на своей базе.

Например, рассмотрим процедуру получения платного хостинга и регистрацию доменного имени для сайта компании «Сатурн». Воспользуемся для этих целей сервисом одного широко известного и надежного провайдера, предлагающего из большого перечня услуг в том числе и регистрацию домена, — фирмы Infobox. На сайте компании по адресу <http://infobox.ru/domains/> можно ознакомиться с ценами на имена в доменных зонах (табл. 1.2).

Таблица 1.2. Цены на доменные имена в самых популярных зонах (<http://infobox.ru/domains/>)

Зона	Срок, годы	Регистрация, руб.	Продление, руб.
.ru	1	600	450
.su	1	600	600
.рф	1	590	415
.com, .net, .org, .biz, .info, .us	1	490	490
.name	1	750	750
.eu	1	840	840
.de, .in, .ca	1	680	680
.mobi	1	750	750
.co.uk, org.uk	2	930	930
.bz, .co, .ws	1	980	980
.tv	1	1500	1500
.spb.ru, .msk.ru, .spb.su, .msk.su	1	450	360

Самая приятная информация на этой странице — это подарок, от которого невозможно отказаться, ведь при оформлении хостинга провайдер предлагает зарегистрировать одно доменное имя бесплатно.

Однако пока нам необходимо выбрать доменное имя, ведь не исключено, что оно в интересующей нас доменной зоне занято. Для этого необходимо написать желаемое имя сайта в области проверки домена, указать требуемый домен первого уровня и нажать кнопку Проверить домен. Если домен занят, нужно попробовать изменить имя сайта или домен первого уровня; если же доменное имя свободно, то откроется окно, как на рис. 1.1. В этом окне следует выбрать тариф на обслуживание сайта или только зарегистрировать его доменное имя. В качестве бонуса будет предложено организовать свои электронные почтовые ящики.

После нажатия кнопки Заказать откроется окно формы, где следует аккуратно заполнить предлагаемые поля (рис. 1.2).

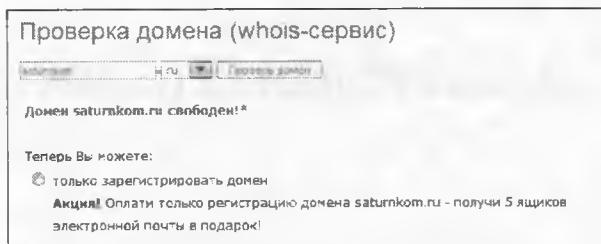


Рис. 1.1. Окно с сообщением о том, что выбранное доменное имя свободно

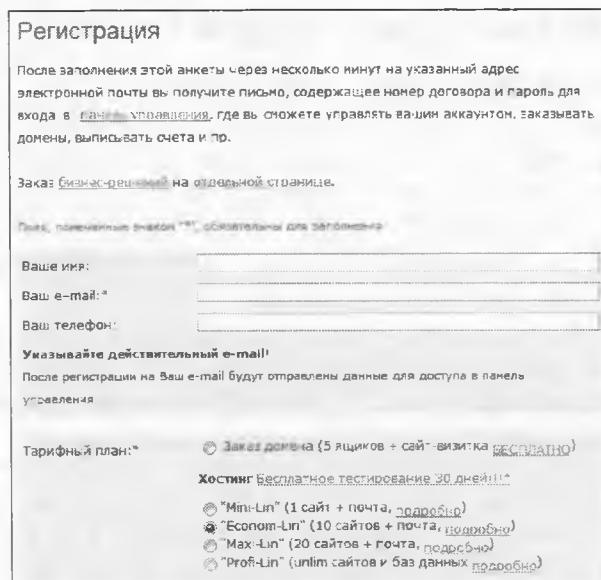


Рис. 1.2. Окно регистрации доменного имени

После заполнения соответствующих полей в форме окна регистрации на указанный вами почтовый ящик в течение минуты придет письмо с паролем доступа и логином к панели управления учетной записью. Войдя в панель управления, вы увидите типовой договор. В нем необходимо заполнить реквизиты владельца доменного имени и счет, который нужно оплатить. Регистрация доменного имени не превышает по времени рабочей недели после оплаты счета, а если вы используете хостинг провайдера, то всю эту неделю можно посвятить отладке своего сайта, поскольку все мощности веб-хостинга будут в вашем распоряжении, за исключением доступа к сайту по доменному имени.

Таким образом, прочитав изложенную в этой главе информацию, вы видите, как и почему мы должны выполнять именно эти, а не другие действия как при регистрации доменного имени сайта, так и при обращении к уже существующему сетевому информационному ресурсу по его доменному имени.

Глава 2

ПЛАНИРОВАНИЕ САЙТА

2.1. Какие вопросы следует задавать

Во многих пособиях и учебниках можно найти различные способы создания сайтов: вручную — по старинке в текстовом редакторе, с помощью визуальных редакторов либо в виде современных блогов, где описаны технические способы размещения различной информации: текстовой, графической, Flash-роликов и прочего на страницах сайта. Однако очень редко в тех же популярных пособиях можно прочесть о необходимости планирования создаваемого информационного объекта.

Между тем, как показывает опыт, именно правильное целеполагание и планирование сайта способно предотвратить на начальном этапе проблемы, которые возникают впоследствии, сэкономить вложенные в сайт средства, а самое главное — время, которое мы затрачиваем не только на создание сайта, но и на последующее его сопровождение. Поэтому, перед тем как сесть за компьютер и начать ваять свой шедевр, не считите за труд уделить полчаса листу обыкновенной бумаги и простому карандашу, ответьте для себя на несколько вопросов и просто нарисуйте то, что вы желаете получить. Все это называется визуализацией и очень помогает не только представить себе желаемое, но и увидеть его во всем многообразии тех предметов и связей между ними, которые и будут составлять ваш информационный объект.

Вопрос первый. Какой сайт я хочу получить?

Дело в том, что в зависимости от содержимого и направленности сайтов их условно можно разделить на несколько групп.

- **Личные страницы** — как правило, это самая массовая категория сайтов, широко применяется как сетевая визитная карточка. Достаточно разместить информацию о себе на личной страничке в Сети и прописывать ее сетевой адрес, например, в резюме, демонстрируя потенциальному работодателю свои наилучшие стороны. Личные страницы обычно просты в создании и по дизайну, который ограничивается только вашим художественным вкусом. Их достаточно разместить на бесплатных серверах в Сети,

да и создать можно, находясь непосредственно в Интернете, с помощью встроенных инструментов мастерской сервера хостера, например на narod.yandex.ru.

- **Некоммерческие сайты** — ресурсы, которые могут представлять некоммерческие объединения, учебные заведения, группы по интересам и т. д. В этом случае сложность и дизайн сайта возрастают. Необходимо предварительно планировать возможность последующего добавления в информационный объект новых функций, например подключения гостевой книги. Сразу возникает вопрос в размещении сайта на платном или бесплатном хостинге и определяются требования к нему. Главенствующая задача, которую необходимо решать: как упростить последующее обслуживание сайта, ведь обычно некоммерческие организации предпочитают не содержать квалифицированных программистов после создания ресурса, а поручают его сопровождение наиболее продвинутому в плане технологий сотруднику организации.
- **Коммерческие сайты** — ресурсы коммерческих организаций. В зависимости от запросов работодателя их сложность и дизайн могут колебаться в широчайших пределах. Обычно коммерческие сайты размещаются на платных хостингах, а для их сопровождения выделяют сотрудника, способного решать широкий круг задач.
- **Контент-сайты (content — содержимое)** — ресурсы — поставщики информации, например поисковые системы, новостные сайты, колонки обозревателей, образовательные ресурсы и пр. Над такими сайтами трудятся команды профессионалов, которые решают не только программные и дизайнерские задачи, но и задачи скрытого характера, например добавляют неявную рекламу или организуют агентов влияния. Соответственно, знаний и вложений информационный объект данного характера требует достаточно весомых.
- **Информационный объект, не размещаемый в сети.** Иными словами, вы создаете какой-либо документ на гипертекстовой основе для использования на персональном компьютере либо в локальной сети предприятия, обладающей высокой пропускной способностью. Это может быть инструкция по использованию чего-либо, интерактивный учебник или презентация, в которых реализованы все возможности гипертекстовых документов, применяемые на стороне клиента. В этом случае снимаются технические ограничения на объем страниц и вес графических объектов, которые критичны при размещении сайта в Интернете, а также можно поступиться привлекательным дизайном в угоду читабельности. Главная задача — донести информацию в наиболее полном виде, максимально использовать возможности организации гипертекста и работы сценариев (маленьких программ, встроенных в тело веб-страницы), например при тестировании сотрудника после изучения им определенной темы.

Вопрос второй. Где я планирую размещать сайт и хочу ли я за это платить?

Вы должны решить для себя проблему хостинга. **Хостинг** (hosting) – услуга по предоставлению вычислительных мощностей для физического размещения информации на сервере, постоянно находящемся в сети.

В связи с этим приведу немного теории. **Сайт** – это просто совокупность веб-страниц, размещенных на жестком диске компьютера. Но если он расположен только на вашем компьютере, то и доступ к его содержимому можете получить только вы. Соответственно, для демонстрации содержимого сайта пользователям Сети необходима специальная программа – веб-сервер, которая получает от сетевого читателя запрос на демонстрацию той или иной страницы вашего сайта и пересыпает эту страницу по запросу. Можно ли поставить эту программу на домашний компьютер? Конечно, можно, даже нужно для создания ресурсов на основе CMS – систем управления содержимым. Но готовы ли вы держать свой компьютер круглосуточно включенным и платить за интернет-канал, которым не пользуетесь? Я думаю, что нет, соответственно, поручите это дело профессионалам, которые зарабатывают деньги тем, что содержат и обслуживают специально предназначенные для этого серверы.

Итак, хостинг бывает платным и бесплатным. С платным достаточно понятно – заключаете договор, оплачиваете дисковое пространство, получаете пароли и сетевые адреса, тут же можно оплатить доменное имя (обычно второго уровня).

Бесплатный сыр, как известно, бывает только в мышеловках. Дело в том, что если у вас не берут денег за хостинг напрямую, то вы обязательно получите какой-либо довесок от хостера. Это может быть рекламный баннер, появляющийся на вашем ресурсе, отдельные страницы, включенные в сайт и содержащие выгодные хостеру сведения, запрет на размещение вашей коммерческой информации, рекламу на ваших же страницах, ограничение на минимальное количество пользователей за определенный промежуток времени и прочие «сырные добавки». Кроме того, бесплатный хостинг обычно подразумевает доменное имя сайта третьего или более уровней, что очень неудобно для пользователя, все-таки есть разница при вводе `mysite.ru` или `mysite.narod.ru`. Бесплатный хостинг также не всегда предоставляет дополнительные серверы для работы более продвинутых динамических сайтов, использующих SQL, PHP или Mail. Достаточно регулярно обновляемый список бесплатных хостеров можно получить по следующим адресам в Сети:

- <http://index-freehosting.narod.ru>;
- <http://www.codenet.ru/hosting/table.php>;
- <http://php5.at.ua/index/0-4>.

Можете также попробовать найти их самостоятельно, используя поисковые системы.

Вопрос третий. Какой топологии сайта лучше придерживаться?

Казалось бы, разве есть богатый выбор? Есть всего две ярко выраженных топологии в конструировании сайтов: **линейная** — когда страницы располагаются последовательно друг за другом, что хорошо применять, например, для фотогалереи, и **древовидная** (иерархическая) — когда содержимое каждой страницы, кроме первой, носящей имя `index.html`, входит в страницу более высокого уровня на правах подраздела. Определив преобладающую топологию размещения информации, вы тем самым сразу же определите меню сайта и связи для перехода с одного информационного уровня на другой. Ведь следует помнить правило двух щелчков: страницу `index.html` — первую страницу сайта — от любой другой страницы должно отделять на более двух щелчков кнопкой мыши.

Вообще, первая страница сайта имеет колоссальное значение, ведь просто серфингующий по Сети пользователь зачастую не идет дальше первой страницы, поэтому главная задача — привлечь и заинтересовать, чтобы пользователь не только остался, но и пошел дальше по страницам именно вашего сайта. Следовательно, из этого требования вытекает следующий вопрос.

Вопрос четвертый. Какую информацию и в каком объеме размещать?

Здесь стоит рассматривать не только содержание — **контент** сайта, но и техническую сторону. Следует помнить о том, что широкополосный выделенный интернет-канал за пределами больших городов — достаточно большая редкость, поэтому и объем загружаемой страницы не должен превышать 100 Кбайт, да и то для обычного модемного соединения это достаточно много, а для лимитированного трафика дорого. Например, для экономии денег некоторые пользователи могут отключать в браузерах функцию просмотра графики, и ваше выстраданное высокохудожественное детище просто не будет оценено, при этом вы не донесете определенную информацию.

Кроме того, мониторы у пользователей разные, и необходимость применять для просмотра содержимого страницы горизонтальную полосу прокрутки при недостаточно заинтересовавшем контенте вызывает раздражение, и зачастую посетитель переходит на другой, менее интересный, но более дружественный сайт. Сюда же можно отнести такой интегрированный параметр, как читабельность — возможность воспринимать информацию без усилий для зрения. Следовательно, при решении этого вопроса имеет смысл рассмотреть контрастность фона и текста, наличие и размер графических объектов и их размещение на страницах. При этом не следует забывать, что навигация по сайту должна быть интуитивно понятна либо четко описана.

2.2. Визуализация – рисуем несколько страниц сайта

На листе бумаги нарисуйте прямоугольник, назовите его `index.html` и в масштабе карандашом схематично изобразите местонахождение элементов меню и при необходимости логотипа, расположите блоки текста и графические объекты в том масштабе, в котором предполагаете их разместить. После этого попробуйте изменить ширину листа и проанализировать, как будет выглядеть страница на узком или широком мониторе, имеет ли смысл жестко задавать размер страницы или использовать для этих целей резиновую таблицу. При этом учтите, что сжиматься и растягиваться могут только текстовые объекты, а графические имеют четко заданные размеры.

Нарисуйте аналогичные прямоугольники и назовите их согласно именам страниц вашего сайта, при этом следует помнить, что имя страницы состоит только из маленьких латинских букв и цифр, чтобы впоследствии не было путаницы. Разместите на них страницочные меню и информационные блоки. Соедините пункты меню страниц с их целевыми объектами и убедитесь, что логика построения интернет-ресурса не нарушена.

При проектировании и создании сайта обратитесь к соответствующему разделу помощи поисковых систем, в которых вы планируете разместить информацию о своем сайте, и внимательно изучите его. Например, просмотрите рекомендации веб-мастеру поисковой системы `yandex.ru`, расположенной по адресу: <http://help.yandex.ru/webmaster/?id=1111316>, либо другой поисковой системы, так как они, как правило, стандартны.

Выполнив описанные выше действия, можете считать, что главное для организации и реализации сайта вы уже сделали, теперь осталась простая техническая работа, требующая лишь усидчивости и внимания.

Глава 3

БЫСТРО И КАЧЕСТВЕННО СТРОИМ САЙТ И РАЗМЕЩАЕМ ЕГО НА БЕСПЛАТНОМ ВЕБ-СЕРВЕРЕ

3.1. Основные понятия: коротко и быстро

Все мы в той или иной степени знакомы с сетевыми ресурсами, ведь именно идея создать свой сайт заставила вас купить эту книгу, поэтому узнать немного теории будет полезно.

Созданная в начале 1990-х годов Всемирная паутина (WWW, World Wide Web) получила столь быстрое по современным меркам распространение потому, что почти мгновенно предоставляла пользователям данные, находящиеся на компьютерах, расположенных в различных местах земного шара. Такое положение вещей стало возможным благодаря тому, что эти документы связаны между собой адресными связями — так называемыми гиперссылками. Таким образом, размещая на странице с текстом различные ссылки за пределы страницы — на другие документы или мультимедийные объекты, хранящиеся даже на других компьютерах сети, то есть используя технологию гипертекста, мы можем получить модель действия Всемирной паутины.

Основной информационной единицей при создании сайта принято считать веб-страницу. Тематически обработанные веб-страницы, размещенные в систематизированных каталогах, будут считаться сайтом, при этом благодаря технологии гипертекста каталоги с веб-страницами могут быть размещены сразу на нескольких компьютерах. Важно только, чтобы данные страницы находились в поле действия веб-сервера — специальной программы, принимающей запрос от посетителя сайта и передающей ему заказанную страницу.

Веб-страница представляет собой обычный текстовый документ, сохраняемый в определенном формате — HTML. Это означает, что страница создана с помощью языка гипертекстовой разметки (HyperText Markup Language) и может использоваться для передачи информации по Сети по протоколу передачи гипертекста <http://> (HyperText Transfer Protocol). В отличие от текстовых файлов, созданных, например, в текстовом процессоре Word, веб-страница не может содержать в себе, в своем теле нетекстовые

элементы: рисунки, ролики и пр. Эти объекты находятся под своими именами в специально отведенных для них каталогах, а на их месте в теле веб-страницы есть указания: откуда, что и как вставлять. Следовательно, веб-страницу можно представить определенной инструкцией, ее читает браузер — программа, с помощью которой просматривают интернет-страницы. Он же выполняет те или иные действия для формирования страницы на экране компьютера, подставляя в определенные места текстовые фрагменты, графические файлы или иные ее объекты. Таким образом, процесс создания сайта можно определить как написание инструкции для браузера по формированию веб-страниц.

Создавать сайты можно различными способами. Один из них — использование подручных средств, например текстового редактора Блокнот или любого другого простейшего текстового редактора. В этом случае необходимо знать элементы языка разметки гипертекста. Другой способ — применение специальных программ-редакторов, позволяющих формировать страницу по принципу «что вижу в поле редактора, то получаю на экране браузера». Третий способ, все более и более популярный в области создания сайтов, — использование систем управления содержимым, так называемые CMS-системы, которые еще часто называют системами модификации контента. В этой книге мы рассмотрим основы создания сайтов всеми перечисленными способами, а какой из них предпочтеть в последующем — решать вам.

3.2. Создание основы сайта: структура, форматирование, наполнение

Шаг первый. Определение задачи. В моей практике был случай, когда после окончания рабочего дня я, одетый и двигающийся к выходу, был остановлен директором и препровожден за компьютер секретаря. Оказывается, пришла директива, по которой необходимо иметь свой сайт, причем созданный еще вчера. Сегодня нужно было срочно представить в компетентные органы его адрес в Интернете, да еще учесть, что вдруг кто-нибудь на ночь глядя решит его посетить, как говорится, чтобы не было стыдно... Естественно, что поздно вечером я не стал писать сайт, но место для него зарезервировал и оформил первую страницу, указав, что ресурс находится на реконструкции буквально с сегодняшнего дня, а до этого был доступен постоянно и работал на всю катушку.

Одним словом, если у вас возникла срочная необходимость оформить свое представительство в Сети и при этом нет специальных средств, то воспользуйтесь подручными. Это дольше и, возможно, менее красочно, а также недостаточно функционально, ведь мы не носим с собой постоянно спецификацию языка HTML 4.0, зато представить себя или свою организацию в Сети вы сможете, буквально сделав несколько шагов.

Итак, шаг первый. Основа создания сайта — планирование, значит, распланируем свой сайт. Он будет состоять из трех связанных гиперссылками страниц, содержащих заранее подготовленные графические элементы. Сайт будет находиться в каталоге `site`, а для графических объектов в нем будет создан подкаталог `graf`. Первая страница ресурса, как и положено, будет носить имя `index.html`, а вторая и третья страницы — `str2.html` и `str3.html` соответственно. При создании сайта помните о простых ограничениях на имена файлов, входящих в него: рекомендуется использовать строчные английские буквы и цифры, а также символы подчеркивания. При этом следует избегать пробелов, точек, кириллических букв и других запрещенных символов. Имеет смысл нумеровать однотипные объекты с указанием типа в имени файла, например `ris1.gif`, `foto3.jpg`.

Итак, приступим к созданию веб-страниц средствами текстового редактора.

Шаг второй. Создание структуры сайта. Первая страница. В первую очередь необходимо создать структуру: создаем папку `site`, в ней дополнительную папку `graf` и три файла с именами `index.html`, `str2.html` и `str3.html`.

Далее открываем страницу `index.html` в простом текстовом редакторе, для этого можно использовать программу Блокнот или любую другую, и начинаем писать код.

Почему для этих целей не рекомендуется использовать сложные текстовые процессоры, как `Word`? Потому что они создают документы в своих собственных форматах, записывая в файлы дополнительный код, который нам абсолютно не нужен. Лучше применять редактор не сложнее Блокнота или специальный редактор для HTML-страниц.

Так называемый код веб-страниц представляет собой обычный текст (контент страницы) в сочетании с элементами разметки — тегами. **Тег** — это инструкция для браузера, прописывающая, каким образом следует оформить на экране объект, который описывается этим тегом. Теги, в свою очередь, заключаются в угловые скобки `< >` и бывают **одиночными** или **парными**. При использовании парного тега первый является открывающим, а второй — закрывающим и поэтому в своем имени содержит символ `/`. Например, тег, определяющий начало веб-страницы, — парный, поэтому, открывая страницу, мы всегда пишем `<html>`, а в конце прописываем тег `</html>`.

Простейшая веб-страница содержит всего несколько тегов:

- `<html>` и `</html>` — определяют начало и конец веб-страницы;
- `<head>` и `</head>` — описывают начало и конец «головы» веб-страницы, не отображаемой в окне браузера;

- `<title>` и `</title>` – определяют заголовок веб-страницы, отображаемый в поле заголовка браузера, в самом верху страницы;
- `<body>` и `</body>` – задают «тело» веб-страницы, то есть ту часть, которая будет отображаться в окне браузера.

Наберем содержимое нашей страницы согласно плану (рис. 3.1). Затем обязательно сохраним страницу: меню **Файл** ▶ **Сохранить**. Не забывайте также сохранять страницу каждый раз, когда внесли в нее какие-либо изменения, в противном случае вы не сможете увидеть их в окне браузера.

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
Компания «Сатурн» предоставит вам возможность провести
отпуск или выходные дни на любой планете ближнего и дальнего
освоенного космоса.
</body>
</html>
```

Рис. 3.1. Начальное содержимое первой веб-страницы, написанное в текстовом редакторе

После сохранения содержимого, не закрывая редактора, открываем с помощью браузера редактируемую страницу. Для этого обычно достаточно дважды щелкнуть кнопкой мыши на ее имени `index.html` либо указать в строке адреса место. В результате в браузере отобразится содержимое нашей страницы (рис. 3.2).

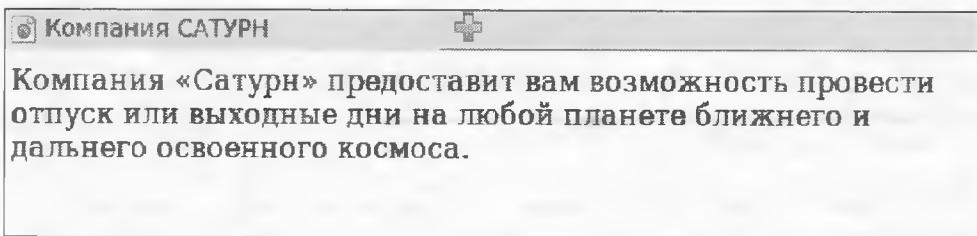


Рис. 3.2. Содержимое веб-страницы в окне браузера

Белый фон, черный цвет букв – это, конечно, читабельно, но очень официально для туристической фирмы, отправляющей своих клиентов в космос. Попробуем изменить цвет букв и фон страницы. Обычно для этих целей присваивают значение параметров тегов, что мы и сделаем. Задавать цветовые параметры можно как словами, так и числовыми значениями (словами более

понятно, а числа дают доступ к большему количеству оттенков). Внесем в текст веб-страницы, расположенной в редакторе, исправления, задав для фона голубой цвет, а для текста — желтый, и не забудем ее сохранить (рис. 3.3).

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body bgcolor=blue text=yellow>
Компания «Сатурн» предоставит вам возможность провести
отпуск или выходные дни на любой планете ближнего и дальнего
освоенного космоса.
</body>
</html>
```

Рис. 3.3. Изменения в теге `<body>`, определяющие цвета фона и текста

После этого обновим страницу в окне браузера и посмотрим, что изменилось. Если полученный результат удовлетворяет, то оставляем его, иначе экспериментируем дальше, явно задавая значения параметров тега.

Шаг третий. Форматирование текста на странице. Если цветовое сочетание текста и фона вас удовлетворило, то пора перейти к форматированию текста на странице. Под **форматированием** подразумевается изменение начертания и размеров символов, а также их выравнивание на странице.

Для изменения размера шрифта текста, являющегося заголовком, используют теги от `<h1></h1>` до `<h6></h6>`, где тег `<h1>` — обозначение самого крупного, а `<h6>` — самого мелкого размера текста в заголовке. Определим заголовком название «Компания „Сатурн“», выделив его нужными тегами (рис. 3.4).

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body bgcolor=blue text=yellow>
<h1>Компания Сатурн</h1>
Компания «Сатурн» предоставит вам возможность провести отпуск
или выходные дни на любой планете ближнего и дальнего
освоенного космоса.
</body>
</html>
```

Рис. 3.4. Выделение заголовка тегами `<h1></h1>`

Сохраним текст в редакторе и обновим содержимое страницы (рис. 3.5).

Поскольку по умолчанию текст всегда выравнивается по левому краю, то для его целевого выравнивания необходимо явно задать некоторые атрибуты тега.

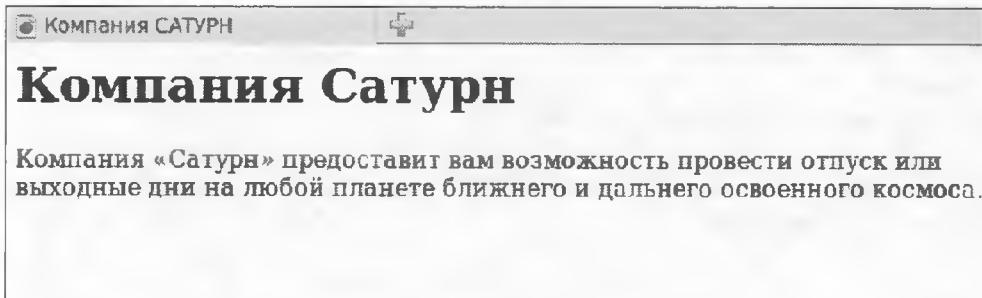


Рис. 3.5. Просмотр заголовка на странице

Атрибут `align` может определять следующие значения: `left` – выравнивание по левому краю, `right` – по правому краю, `center` – по центру, `justify` – по ширине. Внесем изменения, выровняв заголовок на странице по центру (рис. 3.6).

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body bgcolor=blue text=yellow>
<h1 align=center>Компания Сатурн</h1>
Компания «Сатурн» предоставит вам возможность провести отпуск
или выходные дни на любой планете ближнего и дальнего
освоенного космоса.
</body>
</html> |
```

Рис. 3.6. Выравнивание заголовка по центру

Сохраним и обновим страницу в браузере (рис. 3.7).

Для выделения текстовых объектов обычно используют следующие теги:

- `<i></i>` – курсив;
- `` – полужирное начертание;
- `<u></u>` – подчеркивание текста, находящегося между ними.

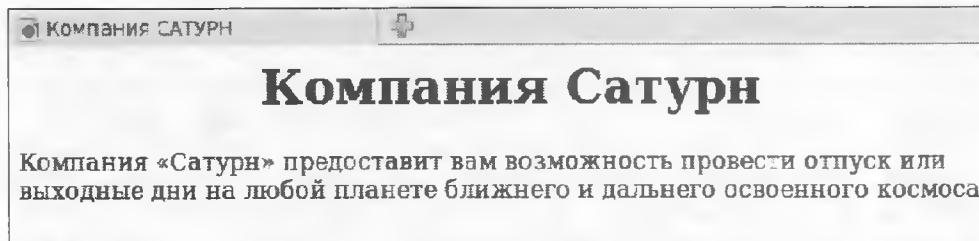


Рис. 3.7. Выравнивание заголовка по центру

Если возникает необходимость выделить текст совокупными тегами, то их можно вложить друг в друга. В этом случае надпись может принять вид `<i>«Сатурн»</i>`. Как видим из примера, нужно учитывать вложенность тегов при закрытии. Применим данные теги для выделения текстовых фрагментов (рис. 3.8).

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
<h1 align=center>Компания Сатурн</h1>
<i>Компания<b><u> «Сатурн»</u></b> предоставит вам
возможность провести отпуск или выходные дни на любой
планете ближнего и дальнего освоенного космоса.</i>|
</body>
</html>
```

Рис. 3.8. Форматирование текста тегами

Сохранив и обновив страницу в браузере, посмотрим, что получилось (рис. 3.9).

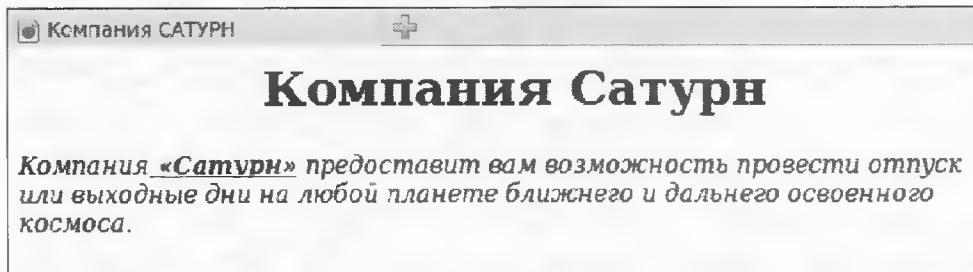


Рис. 3.9. Результат форматирования текста с помощью тегов

Если возникает необходимость изменить размер символов, не указывая теги заголовков, можно использовать следующие способы: теги `<big>` `</big>` и `<small>` `</small>` соответственно увеличивают и уменьшают размер заключенного в них текста по сравнению с текущим, а тег `` `` с атрибутом `size` позволяет задавать размер символов в условных единицах от минимального значения 1 до максимального значения 7. По умолчанию размер символов текста принято считать равным 3 условным единицам, поэтому, явно задавая значение параметра тега со знаком плюс или минус, можно заставлять браузер отображать текст больше или меньше относительно текущего. У тега `` также есть атрибуты: `face` — указывает название предпочтительного шрифта для отображения символов, `color` — задает цвет шрифта. Рассмотрим на примере центрирование и изменение размера шрифта для текста веб-страницы (рис. 3.10).

```

<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
<h1 align=center>Компания Сатурн</h1>
<center><font size=+2><i>Компания<b><u> «Сатурн»</b></u></i></font></center>
предоставит вам возможность провести отпуск или выходные дни на любой
планете ближнего и дальнего освоенного космоса.</i></font></center>
</body>
</html>

```

Рис. 3.10. Пример центрирования и увеличения шрифта с помощью тегов

Не забываем сохранить текст и обновить страницу в браузере, где и наблюдаем результат изменений (рис. 3.11).

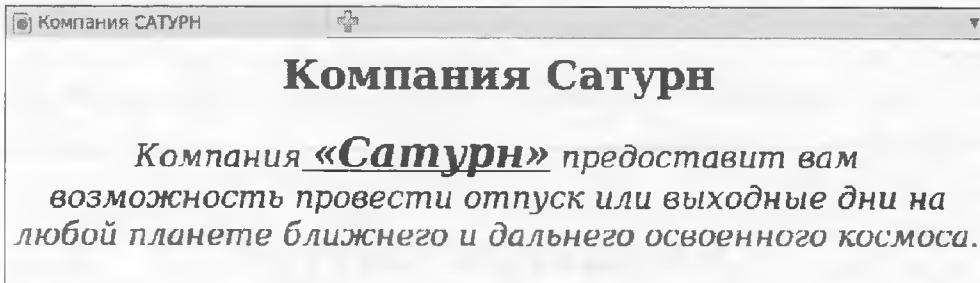


Рис. 3.11. Результат выполнения центрирования и увеличения размеров шрифта

Для придания полной красоты нашему творению выделим первые два слова «Компания „Сатурн“» в отдельный абзац, для чего необходимо воспользоваться непарным тегом `
`, обозначающим конец абзаца (рис. 3.12).

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
<h1 align=center>Компания Сатурн</h1>
<center><font size=+2><i>Компания<b><u> <big><big>«Сатурн»</big></u></b> <br>
предоставит вам возможность провести отпуск или выходные дни на любой
планете ближнего и дальнего освоенного космоса.</i></font></center>
</body>
</html>
```

Рис. 3.12. Использование одиночного тега `
` для разбиения текста на абзацы

Не забываем сохранить текст и обновить страницу в браузере. Так лучше, не правда ли (рис. 3.13)?

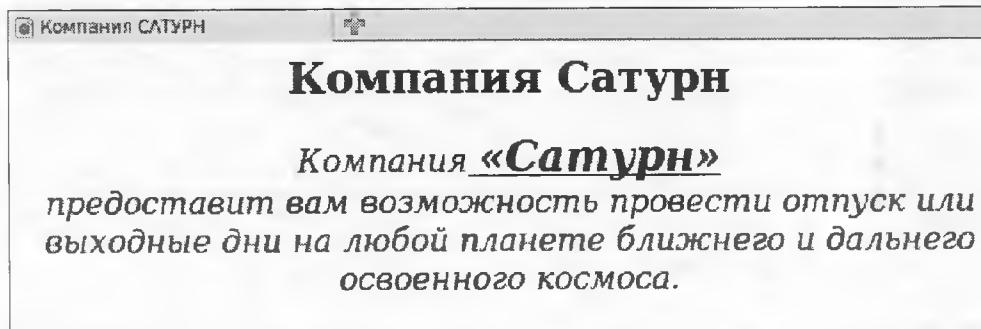


Рис. 3.13. Результат использования тега `
` для разбиения текста на абзацы

Таким образом, оперируя совокупностями тегов, можно выполнять различные операции по форматированию текста на веб-странице. Язык HTML в своей спецификации позволяет использовать не только физическое форматирование текстовых объектов, которое мы сейчас рассмотрели, где в явном виде задаются те или иные значения параметров для форматируемых объектов, но и логическое форматирование.

В случае логического форматирования заключенный в соответствующие теги текстовый фрагмент страницы распознается и форматируется по определенному стилю, с указанием на роль текстового фрагмента в составе документа. Используя эту возможность языка HTML, при оформлении сайта можно применять так называемые каскадные таблицы стилей CSS, которые позволяют управлять форматированием тегов в документе, задавая для каждого заключенного в тег элемента определенный стиль оформления. Например, можно задать синий цвет для заголовков, определяемых тегами `<h1></h1>`, используя параметр `style: <h1 style="color: blue">`, где в качестве значения атрибута указано "свойство: значение".

Каскадные таблицы стилей могут располагаться в файле текущей страницы, определяя для нее стили оформления. Кроме того, они могут находиться в отдельных файлах на различных веб-серверах, определяя стилевое форматирование для всех страниц сайта. Можно также форматировать одну страницу с помощью различных таблиц, из которых выстраивается своеобразная иерархия стилей форматирования.

Продемонстрируем использование форматирования с помощью элемента каскадных таблиц стилей для нашего третьего абзаца.

```
<p style="font-style: italic; font-size: 150%; text-align: center">предоставит вам возможность провести отпуск или выходные дни на любой планете ближнего и дальнего освоенного космоса.</p>
```

Здесь тег `<p></p>` определяет текст как отдельный абзац, а `style` — атрибут, задающий для фрагмента курсивное начертание командой `font-style: italic`. Относительный размер шрифта задан в процентах: `font-size: 150%`. Текстовый фрагмент выровнен по центру: `text-align: center`. При этом размер шрифта можно задавать не только в процентах, но и в явных единицах — сантиметрах (см) или пунктах (pt). Следует учитывать, что явно заданные размеры не позволяют браузеру адаптировать текст под размер раскрытоого окна и, как следствие, могут сделать его нечитаемым. Пример рассмотрен на рис. 3.14.

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
<h1 align=center>Компания Сатурн</h1>
<center><font size=+2><i>Компания<b><u> <big>«Сатурн»</big></u>
<br></i></font></center></b>
<p style="font-style: italic; font-size: 150%; text-align: center">предоставит вам
возможность провести отпуск или выходные дни на любой планете ближнего
и дальнего освоенного космоса.</p>
</body>
</html> |
```

Рис. 3.14. Форматирование абзаца с элементом использования каскадных таблиц стилей

Повторяем операцию сохранения и обновления, открываем в браузере результат (рис. 3.15).

В нашем примере определение стиля располагалось внутри тега элемента, но таким же образом, используя таблицу стилей, в которой заданы свойства для какого-либо элемента, можно устанавливать описанный стиль для нужного элемента, просто указав его.

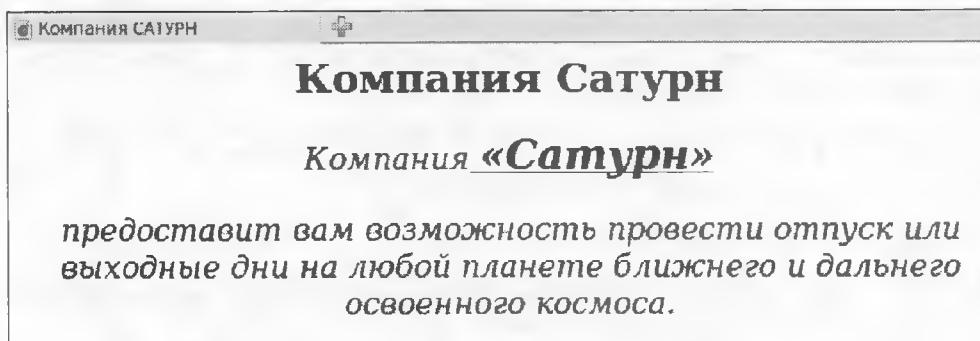


Рис. 3.15. Результат форматирования последнего абзаца с использованием каскадных таблиц стилей

3.3. Списки и таблицы в теле веб-страницы

Шаг четвертый. Использование списков. Перейдем ко второй странице создаваемого сайта и разместим на ней список услуг, которые компания предлагает своим клиентам. Для этого откроем в текстовом редакторе файл с именем str2.html и создадим первоначальную структуру веб-страницы (рис. 3.16). Не стесняйтесь указывать в заголовке страницы название компании — тем самым вы повышаете ее релевантность у поисковых машин.

```
<html>
<head>
<title>Услуги, предоставляемые компанией САТУРН</title>
</head>
<body>
<h1 align=center>Услуги, предоставляемые компанией Сатурн</h1>
</body>
</html>
```

Рис. 3.16. Первоначальная структура второй страницы сайта

Для создания маркированных списков, определяющих равноправные элементы, в спецификации языка HTML применяют следующие теги:

- `` — парный тег определяет начало и конец списка перечислений;
- `` — непарный тег определяет начало элемента списка.

По умолчанию такой список маркируется жирными точками, и при необходимости можно изменить вид маркера, используя атрибут `type` с указанием следующих значений:

- `disc` — точечный маркер, задаваемый по умолчанию;
- `circle` — маркер в виде пустой окружности;
- `square` — маркеры в виде квадратиков.

Рассмотрим примеры использования описанных тегов для организации маркированного списка услуг компании с элементом вложенного списка — выделенной строкой `<ul type=circle>Фобос и Деймос — мифы и легенды. Рекомендуется включить в план посещения планеты Марс.`. Стока выделена исключительно для лучшего восприятия при чтении книги. Пример рассмотрен на рис. 3.17 и 3.18.

```

<html>
<head>
<title>Услуги, предоставляемые компанией САТУРН</title>
</head>
<body>
<h1 align=center>Услуги, предоставляемые компанией Сатурн</h1>

Компания Сатурн предлагает вам провести свои выходные дни или весь отпуск,
совершив следующие увлекательные путешествия:
<ul type=square><li>Романтический ужин на околоземной орбите.
<li>Завтрак с видом на Землю с поверхности Луны.
<li>Луна — известный спутник Земли.
<li>Луна — обратная сторона.
<li>Венера — планета бурь и любви.
<li>Меркурий — то в жар, то в холод.
<li>К Марсу — красной планете на выходные.

<ul type=circle><li>Фобос и Деймос — мифы и легенды. Рекомендуется включить в
план посещения планеты Марс. </ul>

<li>Путешествие к поясу астероидов — привези домой кусочек космоса.
<li>Загадочный гигант Юпитер и его спутники.
<li>Золотые кольца Сатурна.
<li>На краю Ойкумены — комбинированный тур к Урану и Нептуну.
</ul>
</body>
</html>

```

Рис. 3.17. Использование в тексте кода страницы тегов, формирующих маркированный список с элементом вложенности

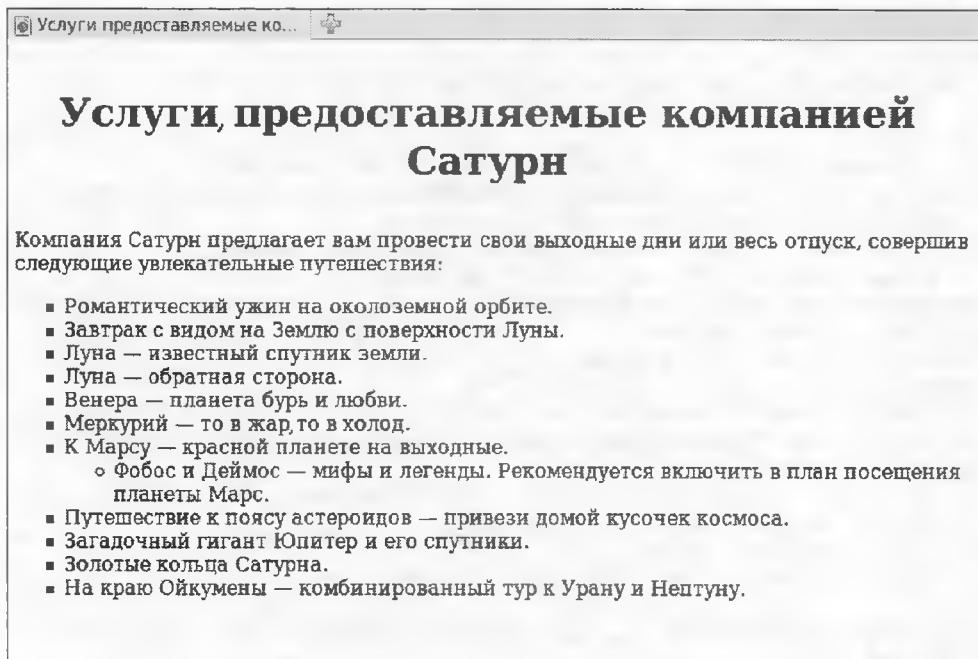


Рис. 3.18. Результат формирования маркированного списка с вложенным элементом списка

Для формирования нумерованных списков перечисления используют аналогичный тег ``, где с помощью параметров его атрибута `type` можно задать тип нумерации:

- 1 — арабскими цифрами (1, 2, 3...);
- I — римскими цифрами (I, II, III...);
- i — малыми римскими цифрами (i, ii, iii...);
- A — прописными латинскими буквами (A, B, C...);
- a — строчными латинскими буквами (a, b, c...).

Поскольку по умолчанию нумерация начинается с первого элемента списка, то с помощью атрибута `start` можно явно задать номер, с которого будет выполняться отсчет элементов списка. Попробуем на примере рис. 3.19 и 3.20 применить теги для формирования нумерованного списка.

Шаг пятый. Таблицы. Мы построили две страницы сайта, и настало время задуматься о навигации, то есть о том, какое меню для передвижения по сайту необходимо создать и как оно будет выглядеть.

```

<html>
<head>
<title>Услуги предоставляемые компанией САТУРН</title>
</head>
<body>
<h1 align=center>Услуги,предоставляемые компанией Сатурн</h1>
Компания Сатурн предлагает вам провести свои выходные дни или весь отпуск,
совершив следующие увлекательные путешествия:
<ul type=square><li>Романтический ужин на околоземной орбите.
<li>Завтрак с видом на Землю с поверхности Луны.
<li>Луна – известный спутник Земли.
<li>Луна – обратная сторона.
<li>Венера – планета бурь и любви.
<li>Меркурий – то в жар, то в холод.
<li>К Марсу – красной планете на выходные.
<ul type=circle><li>Фобос и Деймос – мифы и легенды. Рекомендуется включить в
план посещения планеты Марс. </ul>
<li>Путешествие к поясу астероидов – привези домой кусочек космоса.
<li>Загадочный гигант Юпитер и его спутники.
<li>Золотые кольца Сатурна.
<li>На краю Ойкумены – комбинированный тур к Урану и Нептуну.
</ul>
От совершения увлекательнейшего космического путешествия вас отделяет всего
несколько шагов:
<ol><li>Выберите понравившийся маршрут.
<li>Определитесь со сроками путешествия.
<li>Позвоните нашему консультанту [телефон ***-***-***, круглосуточно].
<li>Оформите путевку.
<li>И вот вы на борту роскошной космической яхты.
</ol>
</body>
</html>

```

Рис. 3.19. Использование тегов для создания нумерованного списка

В простейшем случае для перехода по страницам для создания гиперссылок можно использовать простые текстовые объекты. Проблема будет заключаться в том, что при задании различных настроек окна браузера элементы меню не будут находиться в строго определенных местах экрана, а будут перемещаться по нему, ведь веб-страница – это просто текстовый документ, и появление одного пробела может привести к полной потере форматирования текста. Для разрешения данной проблемы – чтобы было удобно размещать различные элементы на страницах сайта и они не изменили своего положения относительно друг друга при различных настройках окна браузера – применяют таблицы. Рассмотрим применение таблицы для формирования единого стилевого меню навигации, которое можно использовать на всех страницах. Создадим горизонтальную таблицу, состоящую

из одной строки и трех столбцов, в которые и поместим названия пунктов меню нашего сайта в таком виде:

Главная страница	Услуги компании	Вопросы и ответы
------------------	-----------------	------------------

Для формирования таблицы средствами языка HTML нужно помнить следующее утверждение: «Таблица состоит из строк, а строки — из ячеек».

Услуги, предоставляемые компанией Сатурн

Компания Сатурн предлагает вам провести свои выходные дни или весь отпуск, совершив следующие увлекательные путешествия:

- Романтический ужин на околоземной орбите.
- Завтрак с видом на Землю с поверхности Луны.
- Луна — известный спутник земли.
- Луна — обратная сторона.
- Венера — планета бурь и любви.
- Меркурий — то жар, то холод.
- К Марсу — красной планете на выходные.
 - Фобос и Деймос — мифы и легенды. Рекомендуется включить в план посещения планеты Марс.
- Путешествие к поясу астероидов — привези домой кусочек космоса.
- Загадочный гигант Юпитер и его спутники.
- Золотые кольца Сатурна.
- На краю Ойкумены — комбинированный тур к Урану и Нептуну.

От совершения увлекательнейшего космического путешествия вас отделяет всего несколько шагов:

1. Выберите понравившийся маршрут.
2. Определитесь со сроками путешествия.
3. Позвоните нашему консультанту (телефон ***-***-***, круглосуточно).
4. Оформите путевку.
5. И вот вы на борту роскошной космической яхты.

Рис. 3.20. Результат применения тегов ``

Таблица создается с помощью парного тега `<table></table>`, описывающего начало и конец таблицы. Атрибут `border` задает ширину рамки в пикселях. Если он равен 0, то рамка на экран не выводится, но таблица сохраняет свое форматирование, то есть создаются прозрачные границы таблицы. Атрибут `align` в теге таблицы выравнивает саму таблицу относительно экрана, а ее содержимое не трогает. Атрибут `bgcolor` задает цвет фона для всей таблицы. Ширину таблицы определяет атрибут `width`, а высоту — `height`. Ширина и высота могут указываться как явно в пикселях, так и в процентах от размера окна браузера, что более предпочтительно, так как таблица становится резиновой. Иными словами, можно сжимать

или растягивать текстовое содержимое ячейки, избегая тем самым появления в окне браузера полос прокрутки, что улучшает читабельность веб-страницы.

Для формирования строки таблицы используется парный тег `<tr></tr>`, имеющий атрибуты, аналогичные атрибутам тега `<table>`. Этот тег заключает в себе теги ячеек, формирующих строку.

Тег формирования ячейки таблицы — парный тег `<td></td>` с достаточно большим количеством атрибутов:

- `align` — горизонтальное выравнивание содержимого ячейки;
- `valign` — вертикальное выравнивание содержимого ячейки; может принимать следующие значения форматирования:
 - `top` — по верхнему краю;
 - `middle` — по центру;
 - `bottom` — по нижнему краю;
- `width` — ширина ячейки и, соответственно, столбца, в котором находится (в пикселях или в процентах от ширины таблицы);
- `height` — высота ячейки и, соответственно, всей строки, в которой находится;
- `colspan` — количество объединяемых ячеек в строке;
- `rowspan` — количество объединяемых ячеек в столбце;
- `bgcolor` — цвет фона ячейки.

Вернемся на нашу первую страницу `index.html`, открыв ее в текстовом редакторе, и с помощью определяющих таблицу тегов построим навигационное меню. Рассмотрим текст кода:

- `<table width=90% align=center border=0>` — строка задает таблицу шириной 90 % от ширины экрана браузера и с невидимой границей;
- `<tr align=center>` — задает строку таблицы и явное выравнивание ее содержимого по центру;
- `<td>Главная страница</td>` — описывает содержимое ячейки;
- `<td>Услуги компании</td>` — определяет содержимое ячейки;
- `<td>Вопросы и ответы</td>` — описывает содержимое ячейки;
- `</tr>` — закрывает строку таблицы;
- `</table>` — закрывает таблицу;
- `
` — дает инструкцию пропустить пустую строку — символ «возврат каретки»;

-
 — указывает пропустить пустую строку — символ «возврат каретки».

Рассмотрим выполнение задачи на рис. 3.21 и 3.22.

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
<h1 align=center>Компания Сатурн</h1>

<table width=90% align=center border=0>
<tr align=center>
<td>Главная страница</td>
<td>Услуги компании</td>
<td>Вопросы и ответы</td>
</tr>
</table>
<br>
<br>

<center><font size=+2><i>Компания<b><u> <b><big>«Сатурн»</big></u></b>
</i></font></center><b>
<p style='font-style: italic; font-size: 150%; text-align: center'>предоставит вам
возможность провести отпуск или выходные дни на любой планете ближнего и
дальнего освоенного космоса.</p>
</b>
</body>
</html>
```

Рис. 3.21. Использование тегов формирования таблицы при оформлении навигационного меню сайта

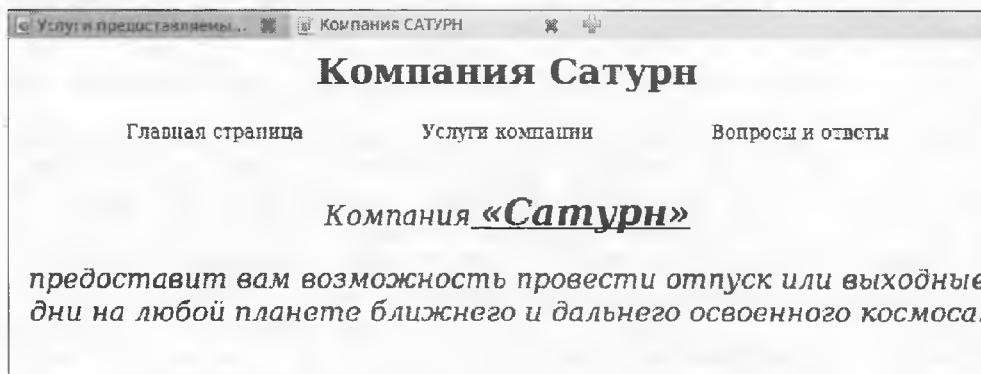


Рис. 3.22. Результат использования скрытой таблицы при оформлении меню сайта

3.4. Организация гиперссылок

Шаг шестой. Гиперссылки. Меню навигации оформлено, осталась самая малость — заставить его работать, то есть при выборе пункта меню должен осуществляться переход на страницу, указанную в меню. Это свойство гипертекста и превращает простую совокупность обычных текстовых страниц в гипертекстовый документ.

Преобразование текстового или графического объекта в гиперссылку возможно с помощью парного тега `<a>`, его обязательным атрибутом `href` является документ, на который и указывает ссылка, дополнительный атрибут `title` отображает текст подсказки, появляющейся при наведении указателя мыши на область гиперссылки.

Применим полученные знания, оформив надлежащим образом пункты нашего меню.

- `<td>Главная страница</td>` — в этой строке переход осуществляется на страницу с именем `index.html`, всплывающая подсказка показывает текст **Переход на главную страницу сайта компании Сатурн**, а гиперссылкой является надпись **Главная страница**.
- `<td>Услуги компаний</td>` — здесь переход осуществляется на страницу с именем `str2.html`, всплывающая подсказка показывает текст **Предоставляемые компанией Сатурн услуги**, а гиперссылкой является надпись **Услуги компаний**.
- `<td>Вопросы и ответы</td>` — в этой строке переход осуществляется на страницу с именем `str3.html`, а всплывающая подсказка показывает текст **Часто задаваемые вопросы по услугам компании Сатурн**, гиперссылкой является надпись **Вопросы и ответы**.

Рассмотрим на примере рис. 3.23 применение тегов для оформления гиперссылок.

Осталась последняя задача — скопировать созданную таблицу с навигационным меню на все три страницы нашего сайта. Для этого выделяем текстовый фрагмент, начиная с тега `<table>` и заканчивая тегом `</table>`, выбираем пункт меню **Правка ▶ Копировать**, копируя выделенный фрагмент в буфер памяти. Открываем в текстовом редакторе страницы `str2.html` и `str3.html` и, установив курсор после тега заголовка `<h1>`, выбираем пункт меню **Правка ▶ Вставить**, тем самым вставляя текстовый фрагмент из буфера памяти в тело страницы. Таким образом, содержимое файла `str2.html` будет выглядеть, как на рис. 3.24.

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
<h1 align=center>Компания Сатурн</h1>
<table width=90% align=center border=0>
<tr align=center>

<td><a href=index.html title="Переход на главную страницу сайта компании Сатурн">Главная страница</a></td>
<td><a href=str2.html title="Представляемые компанией Сатурн услуги">Услуги компании</a></td>
<td><a href=str3.html title="Часто задаваемые вопросы по услугам компании Сатурн">Вопросы и ответы</a></td>

<br>
<table>
<br>
<br>
<center><font size=+2><i>Компания<b><u> Сатурн</u> <big><big>Сатурн</big></big></u></i></font></center><b>
<p style="font style: italic; font size: 150%; text align: center">предоставит вам возможность провести отпуск или
выходные дни на любой планете ближнего и дальнего освоенного космоса.</p>
</body>
</html>
```

Рис. 3.23. Применение тегов для оформления гиперссылок при создании навигационного меню сайта

```
<html>
<head>
<title>Услуги предоставляемые компанией САТУРН</title>
</head>
<body>
<h1 align=center>Услуги, предоставляемые компанией Сатурн</h1>

<table width=90% align=center border=0>
<tr align=center>

<td><a href=index.html title="Переход на главную страницу сайта компании Сатурн">Главная страница</a></td>
<td><a href=str2.html title="Представляемые компанией Сатурн услуги">Услуги компании</a></td>
<td><a href=str3.html title="Часто задаваемые вопросы по услугам компании Сатурн">Вопросы и ответы</a></td>
</tr>
</table>

Компания Сатурн предлагает вам провести свои выходные дни или весь отпуск, совершив следующие
увлекательные путешествия:
<ul type=square><li>Романтический ужин на околоземной орбите.
<li>Звездоплав с видом на Эмиле с поверхности Луны.
<li>Луна – известный спутник Земли.
<li>Луна – обратная сторона.
<li>Бенгера – пляжиста бурь и любви.
<li>Меркурий – то в жар, то в холод.
<li>К Марсу – красной планете на выходные.
<li type=circle><li>Фобос и Деймос – мифы и легенды. Рекомендуется включить в план посещения планеты
Марс. </li>
<li>Путешествие к поясу астероидов – привези домой кусочек космоса.
<li>Эзотерический гигант Юпитер и его спутники.
<li>Золотые кольца Сатурна.
<li>На краю Ойкумены – комбинированный тур к Урану и Нептуну.
</li>
От совершения увлекательнейшего космического путешествия вас отделяет всего несколько шагов:
<ol><li>Выберите понравившийся маршрут.
```

Рис. 3.24. Содержимое страницы str2.html
со вставленным навигационным меню

Содержимое файла str3.html будет выглядеть по-другому, ведь над ним мы еще не работали (рис. 3.25).

```
<table width=90% align=center border=0>
<tr align=center>
<td><a href=index.html title="Переход на главную страницу сайта компании Сатурн">Главная
страница</a></td>
<td><a href=str2.html title="Предоставляемые компанией Сатурн услуги">Услуги компании</a></td>
<td><a href=str3.html title="Часто задаваемые вопросы по услугам компании Сатурн">Вопросы и
ответы</a></td>
</tr>
</table>
```

Рис. 3.25. Содержимое страницы str3.html
с неоформленной структурой страницы

Несмотря на то что структура страницы str3.html не оформлена, меню навигации страницы работает. В этом можно убедиться, сохранив все три файла и вызвав их в окне браузера, а затем использовав для перехода между страницами созданное навигационное меню (рис. 3.26).

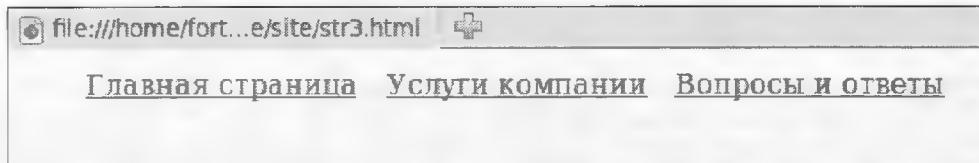


Рис. 3.26. Иллюстрация рабочего меню, расположенного
на неоформленной странице str3.html

Продолжим работать со страницей str3.html и для начала все-таки структурируем ее согласно правилам, введя в ее содержимое необходимые теги (рис. 3.27).

```
<html>
<head>
<title>Вопросы и ответы по услугам компании Сатурн</title>
</head>
<body>
<table width=90% align=center border=0>
<tr align=center>
<td><a href=index.html title="Переход на главную страницу сайта компании Сатурн">Главная
страница</a></td>
<td><a href=str2.html title="Предоставляемые компанией Сатурн услуги">Услуги компании</a></td>
<td><a href=str3.html title="Часто задаваемые вопросы по услугам компании Сатурн">Вопросы и
ответы</a></td>
</tr>
</table>
</body>
</html>
```

Рис. 3.27. Структурированное оформление страницы str3.html
с описанным навигационным меню

Поскольку страница посвящена часто задаваемым вопросам и на ней будут размещены достаточно объемные ответы, которые явно не поместятся в окне браузера, оформим ее с помощью полностью отображаемого на экране списка вопросов. Эти вопросы будут гиперссылками на ответы к ним. Выбирая необходимый вопрос, пользователь щелкает на нем кнопкой мыши, активизируя гиперссылку. Ссылка перемещает его к ответу, который может находиться в любом месте страницы.

Для реализации данной задачи сначала определяются так называемые якоря — точки, с которых начинаются ответы. Они представляют собой такие места в тексте страницы, на которые необходимо перенести пользователя при активизации им гиперссылки. Это выполняется с помощью все того же тега `<a>`, только записанного без атрибута `href`. Вместо этого атрибутом `name` явно задается имя якоря, на который и будет указывать гиперссылка. В нашем случае для первого вопроса это будет выглядеть так: ``. Затем обычным способом прописываем гиперссылку с указывающим объекта с помощью атрибута `href`, обязательно поместив в параметрах префикс `#`, говорящий о том, что ссылка действует внутри документа. В нашем случае она будет выглядеть следующим образом: `Вопрос 1.
`. Таким образом, код веб-страницы `str3.html` будет выглядеть так, как показано на рис. 3.28.

```

<td><a href=str2.html title="Представляемые компанией Сатурн услуги">Услуги
компании</a></td>
<td><a href=str3.html title="Часто задаваемые вопросы по услугам компании
Сатурн">Вопросы и ответы</a></td>
</tr>
</table><br><br>
<a href="#vopr1">Вопрос 1.</a><br>
<a href="#vopr2">Вопрос 2.</a><br>
<a href="#vopr3">Вопрос 3.</a><br>
<a href="#vopr4">Вопрос 4.</a><br>
<a href="#vopr5">Вопрос 5.</a><br>
<br><br><br><br>
<a name=vopr1></a>Ответ на вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1. Ответ на
вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1.
Ответ на вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1. Ответ на
вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1. Ответ на вопрос 1.

```

Рис. 3.28. Пример фрагмента содержимого страницы `str3.html`.

Демонстрация организации внутренних ссылок на веб-странице

Кроме того, с помощью гиперссылок можно организовать и автоматический запуск почтового клиента, если пользователь решит задать вопрос письменно. Для этого в теге организации гиперссылки используется значение `mailto` атрибута `href`, в котором явно задается адрес электронной почты получателя, в нашем случае компании «Сатурн». С помощью

дополнительного атрибута `?subject` можно задать тему письма, а атрибут `body` позволит оформить его шаблон. Рассмотрим это на примере.

- `<a href="mailto:saturn@saturn.ru` — строка задает вызов почтовой программы, установленной по умолчанию, с явным указанием почтового адреса `saturn@saturn.ru` в поле Кому.
- `?subject=Вопрос%20в%20компанию%20Сатурн` — строка, явно заполняющая поле Тема текстом Вопрос в компанию Сатурн, а совокупность символов `%20` обозначает пробел.
- `&body=Здравствуйте%20уважаемые%20сотрудники%20обращаюсь%20к%20вам%20со%20следующим%20вопросом: ">` — строка, явно прописывающая начало письма следующим текстом: Здравствуйте, уважаемые сотрудники, обращаюсь к вам со следующим вопросом:
- активировав данную ссылку `Спасибо.` — строка, определяющая гиперссылку с текстом активировав данную ссылку.

Рассмотрим данный код непосредственно в теле страницы (рис. 3.29).

```
<td><a href="str3.html" title="Часто задаваемые вопросы по услугам компании Сатурн">Вопросы и ответы</a></td>
<br>
</table><br><br>

Уважаемые посетители сайта компании Сатурн, если Вы не нашли ответ на интересующий Вас вопрос, то, пожалуйста,
обратитесь к нам письменно.
<a href="mailto:saturn@saturn.ru
?subject=Вопрос%20в%20компанию%20Сатурн
&body=Здравствуйте%20уважаемые%20сотрудники%20обращаюсь%20к%20вам%20со%20следующим%20вопросом:>">
активировав данную ссылку. </a>Спасибо.
<br><br>

<a href="#Vopr1">Вопрос 1.</a><br>
<a href="#Vopr2">Вопрос 2.</a><br>
```

Рис. 3.29. Пример расположения кода гиперссылки, организующей вызов и заполнение установленной по умолчанию почтовой программы

В результате выполнения всех манипуляций страница `str3.html` в окне браузера будет выглядеть следующим образом (рис. 3.30).

При активизации гиперссылки откроется почтовая программа, в которой будет заполнено поле с текстом письма (рис. 3.31).

3.5. Графические объекты на веб-странице

Шаг седьмой. Вставка графических объектов на веб-страницу. Вставка графических объектов в тело страницы не вызывает трудностей, так как используется все тот же принцип — применение инструктирующего тега, которым является непарный тег ``. Графические объекты, призванные украшать веб-страницы, должны сохраняться в файлах строго определенных форматов: JPG, GIF, PNG. Иными словами, подготовить небольшую картинку можно в любом графическом редакторе, главное — сохранить ее в надлежащем формате.

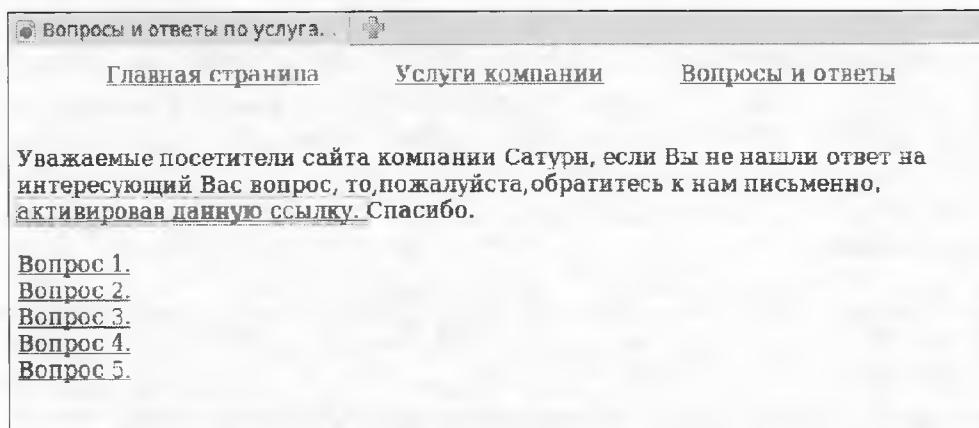


Рис. 3.30. Вид оформленной страницы вопросов и ответов

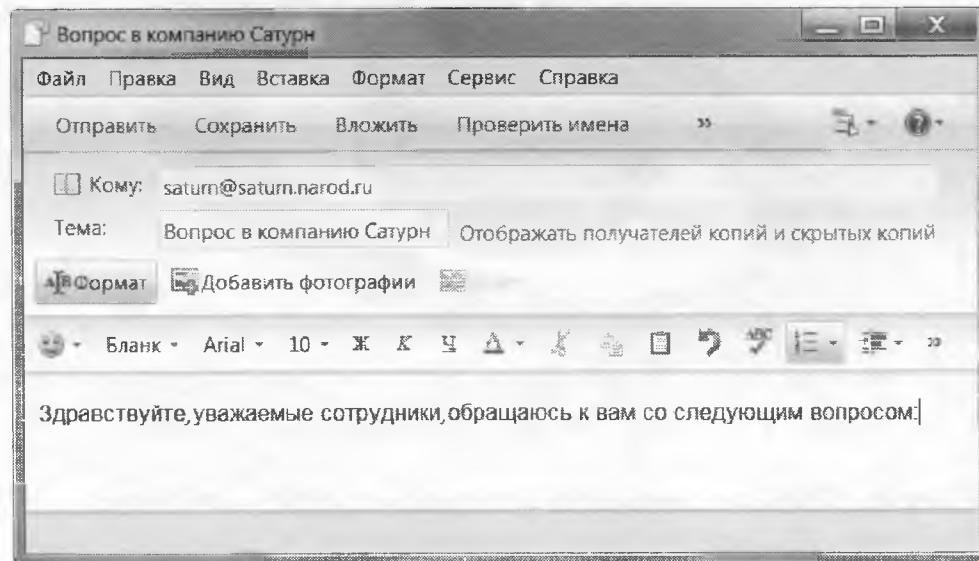


Рис. 3.31. Окно почтовой программы, активизированное атрибутом mailto

Нарисуем эмблему нашей компании — опоясанный кольцами Сатурн — и сохраним в отведенной под рисунки папке *graf* под именем *saturn* (рис. 3.32).

У тега вставки графических изображений ** довольно много атрибутов.

- *src* — обязательный атрибут, указывающий местоположение графического файла, при этом необходимо задавать либо абсолютную ссылку на файл — полный URL, либо относительную — путь к текущей папке, в которой содержится веб-страница.

- `align` — определяет способ обтекания рисунка текстом: `left` — рисунок слева, а текст справа, `right` — рисунок справа, а текст слева.
- `width` и `height` — высота и ширина рисунка в пикселях, задаваемые явно и определяющие место, выделяемое браузером.
- `alt` — альтернативный текст, появляющийся, пока рисунок не загрузился или вместо него, если режим отображения графики отключен в браузере посетителя.
- `border` — определяет толщину рамки вокруг рисунка.
- `hspace` и `vspace` — определяют горизонтальное и вертикальное поле, отделяющее рисунок от текста.

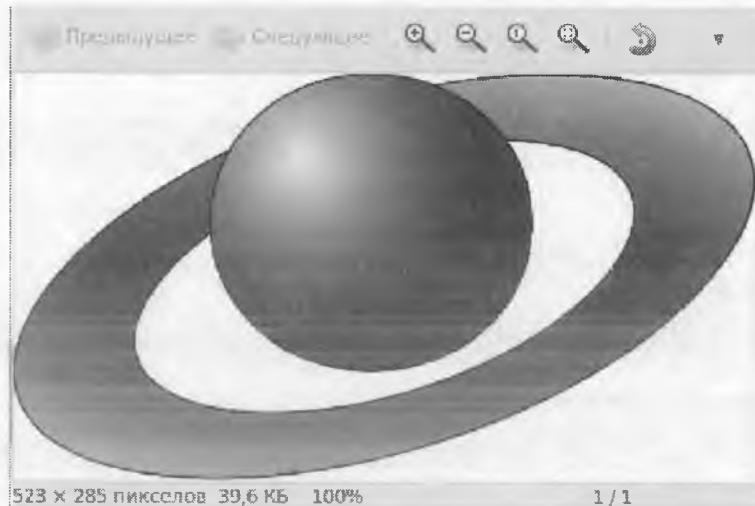


Рис. 3.32. Графический объект, вставляемый на веб-страницу

Поместим эмблему компании на главной странице нашего сайта, прописав следующие параметры тега ``:

```

```

Расшифровать их можно следующим образом: поместить рисунок с полным именем `graf/saturn.png`, объявив замещающий текст как Эмблема компании Сатурн, зарезервировав ширину 250 пикселов, а высоту — 150 пикселов, а также расположив рисунок по правому краю страницы. Пример фрагмента кода представлен на рис. 3.33.

Результат выполнения данного кода можно наблюдать на рис. 3.34.

Если слегка уменьшить наше изображение и «притушить» яркость, то его можно использовать в параметрах тега `<body>` для задания фона страницы.

Продемонстрируем эту возможность формирования фонового изображения и замостим им страницу str2.html, записав следующий код:

```
<body background=graf/saturnfon.png>
```

```
<html>
<head>
<title>Компания САТУРН</title>
</head>
<body>
<h1 align=center>Компания Сатурн</h1>
<table width=90% align=center border=0>
<tr align=center>
<td><a href=index.html title="Переход на главную страницу сайта компании Сатурн">Главная страница</a></td>
<td><a href=str2.html title="Предоставляемые компанией Сатурн услуги">Услуги компании</a></td>
<td><a href=str3.html title="Часто задаваемые вопросы по услугам компании Сатурн">Вопросы и ответы</a></td>
</tr>
</table>
<br>
<img src=graf/saturn.png alt="Эмблема компании Сатурн" width=250 height=150 align=right>
<center><font size=+2><b><u>Компания</u></b><big>«Сатурн»</big></font></center></b>
<p style="font-style: italic; font-size: 150%; text-align: center">предоставит вам возможность провести отпуск или
выходные дни на любой планете ближнего и дальнего освоенного космоса.</p>
</body>
</html>
```

Рис. 3.33. Фрагмент кода вставки рисунка на веб-страницу

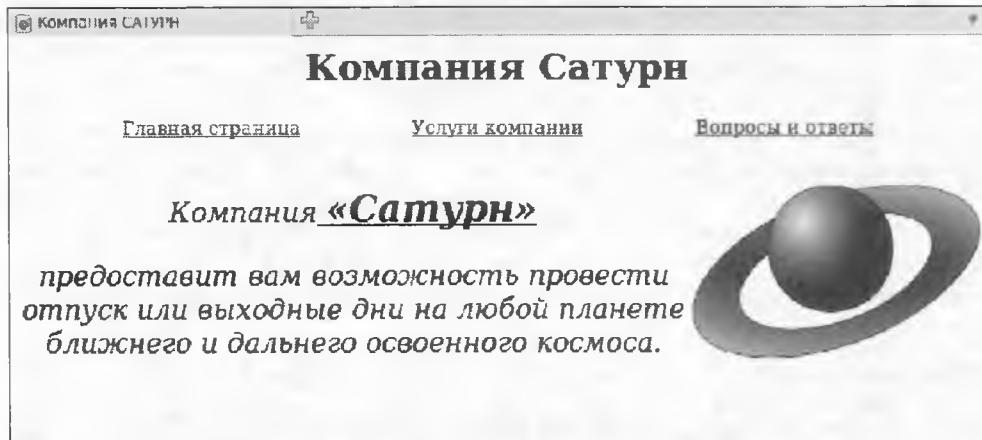


Рис. 3.34. Пример размещения графического объекта на веб-странице

Здесь saturnfon.png — имя фонового файла. Результат выполнения представлен на рис. 3.35.

Применение изображений в качестве фона накладывает следующие требования: либо маленькие по размеру изображения, дублируясь браузером, мостят все пространство открытого окна, как в нашем случае, либо под конкретное фоновое изображение необходимо жестко задавать размеры

страницы, которые будут четко соблюдаться для всех возможных настроек экрана пользователя.

Услуги, предоставляемые ко...

Услуги, предоставляемые компанией Сатурн

Главная страница

Услуги компании

Вопросы и ответы

Компания Сатурн предлагает вам провести свой выходные дни или весь отпуск, совершив следующие увлекательные путешествия:

- Романтический ужин на Роколоземной орбите.
- Завтрак с видом на Землю с поверхности Луны.
- Луна — известный спутник земли.
- Луна — обратная сторона.
- Венера — планета бурь и любви.
- Меркурий — то в жар, то в холод.
- К Марсу — красной планете на выходные.
- Фобос и Цеймос — мифы и легенды. Рекомендуется включить в план посещения планеты Марс.
- Путешествие к ядру астероидов — привези домой кусочек космоса.
- Загадочный гигант Юпитер и его спутники.

Рис. 3.35. Использование изображения в качестве фона веб-страницы

3.6. Размещение сайта

Шаг восьмой. Размещение сайта на бесплатном хостинге. Мы вплотную приблизились к размещению в Интернете созданного нами сайта. Что подразумевается под словами «размещение сайта»? Созданный нами рабочий сайт, находящийся в папке site, пока доступен только нам и только с нашего компьютера. Для представления его всему сетевому сообществу необходимо поместить все содержимое папки site (обратите внимание — не саму папку site, а ее содержимое!) в область действия веб-сервера.

Веб-серверы, на которых осуществляется размещение сайтов — так называемый хостинг, подразделяются на платные и бесплатные, но об этом более подробно будет рассказано в последующих главах. Мы же в критических условиях срочности и недостатка времени и денег на оплату хостинга остановимся на бесплатных серверах размещения сайтов. Бесплатных серверов в Сети много. Укажите в поисковике запрос список серверов бесплатного хостинга или что-то подобное и выбирайте по душе. Но, как известно про бесплатный сыр в мышеловке, ничего бесплатного не бывает, и в любом случае вы получите какой-либо довесок от фирмы-хостера. Не мудрствуя лукаво, обратимся к старейшему бесплатному хостеру Рунета narod.ru, так как он давно работает и особых нареканий не вызывал. Кроме того, он достаточно демократичный — из рекламы выдает только маленький модуль в верхнем углу

страницы, который можно обыграть так, чтобы он не закрывал важный материал. Хостер предоставляет специальные инструменты для создания или исправления страниц, находящихся непосредственно на хостинге, — так называемую мастерскую, и позволяет загружать объекты средствами хостинга. Для этого введем в адресной строке браузера <http://narod.yandex.ru> и получим приглашение к публикации (рис. 3.36).



Рис. 3.36. Начальная страница бесплатного хостинга <http://narod.yandex.ru>

Главное, что предстоит сделать на этом этапе, — выбрать доменное имя сайта. Дело в том, что доменное имя, предоставляемое данным сервисом, будет третьего уровня: `имя.narod.ru`, и может быть уже занято, поэтому необходимо проявить фантазию, чтобы выбрать запоминающееся и в то же время не длинное имя сайта для последующего удобства пользователей, набирающих его в адресной строке браузера. Щелкаем на ссылке **Создайте свой сайт** и попадаем в поле регистрации (рис. 3.37).

Перейдем по ссылке **Зарегистрироваться** и начнем саму процедуру регистрации, заполняя соответствующие поля (рис. 3.38).

Нажмем кнопку **Дальше**, что притаилась в самом низу экрана, и перейдем к следующему окну регистрации (рис. 3.39).

Проверив пароли, не поленитесь и заполните дополнительные поля, поскольку вместе с сайтом сервис организует бесплатный почтовый ящик, имя которого совпадает с именем сайта. Снова нажмите расположенную в самом низу кнопку **Зарегистрировать**, тем самым переходя к следующему окну (рис. 3.40).



Рис. 3.37. Начало регистрации в сервисе narod.yandex.ru

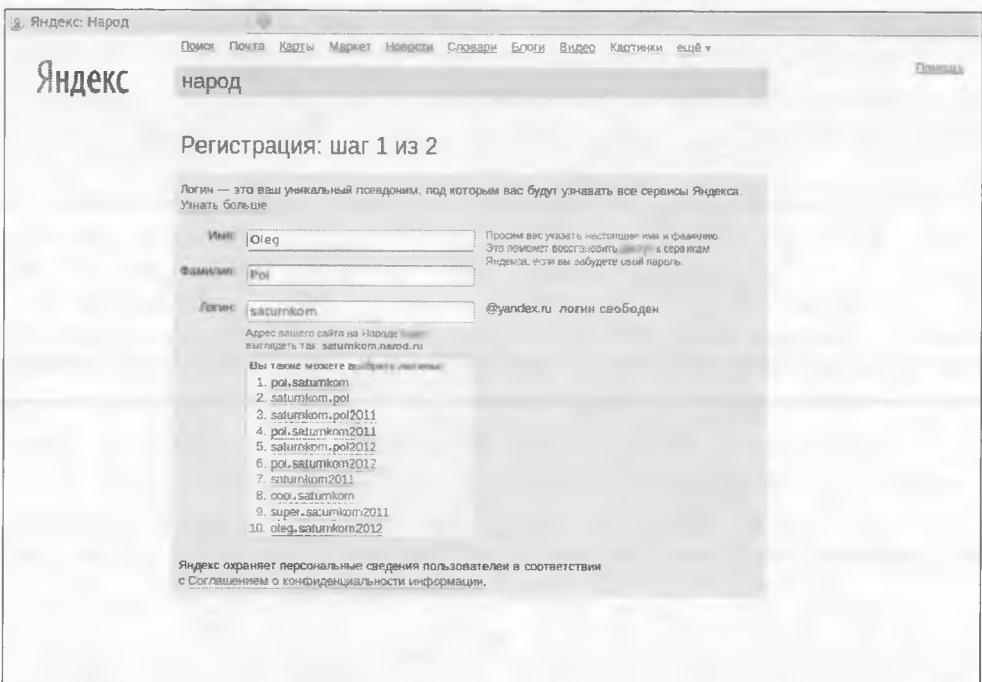


Рис. 3.38. Первый шаг регистрации на сервисе narod.yandex.ru

Регистрация: шаг 2 из 2

Вы выбрали логин: **saturnkom**

Придумайте пароль: [Как выбрать пароль](#) [Показать текст пароля](#)
Несколько 12 символов

Подтвердите пароль: [введенно верно](#)

Секретный вопрос: [Если вы забудете пароль, то можете](#)
получить доступ, ответив
на этот вопрос.

Ответ:

Другой e-mail: [Важно](#)
Если вы забудете пароль, мы сможем
отправить вам SMS с кодом
для его восстановления. На следующий
стадии будет показана инструкция

Мобильный телефон: [Помощь](#)
в формате +7 938 323 11 22

Индекс: Введите символы с картинки слева:
→ **damrlld**
Ребят не сможете ввести символы с картинки, а вы сможете.

[показать другую картинку](#)

Рис. 3.39. Второй шаг регистрации на сервисе narod.yandex.ru

Яндекс народ

Поздравляем, регистрация успешно завершена!

Ваш новый e-mail: **saturnkom@yandex.ru**

Логин: **saturnkom**

Имя и фамилия: **Oleg Pol**

Секретный вопрос и ответ: **██████████**

Держите ваш пароль в секрете и помните, что сотрудники Яндекса никогда и ни под каким предлогом не спросят пароль по телефону или электронной почте.

Распечатать или сохранить в файл
на случай, если вы забудете пароль

Что дальше

[Начать пользоваться сервисом Народ](#)



[Рассказать о себе еще немного](#)

Пол: мужской женский

День рождения: [Помощь](#)

[Сохраните](#)

Рис. 3.40. Завершение регистрации на сервисе narod.yandex.ru

По завершении регистрации очень рекомендуется либо распечатать, либо сохранить паспортные данные вашего сервиса в отдельный файл, используя для этого специальные ссылки на странице. Активируем

ссылку Начать пользоваться сервисом Народ и перейдем к странице, позволяющей начать создание или размещение сайта на сервисе narod.yandex.ru (рис. 3.41).

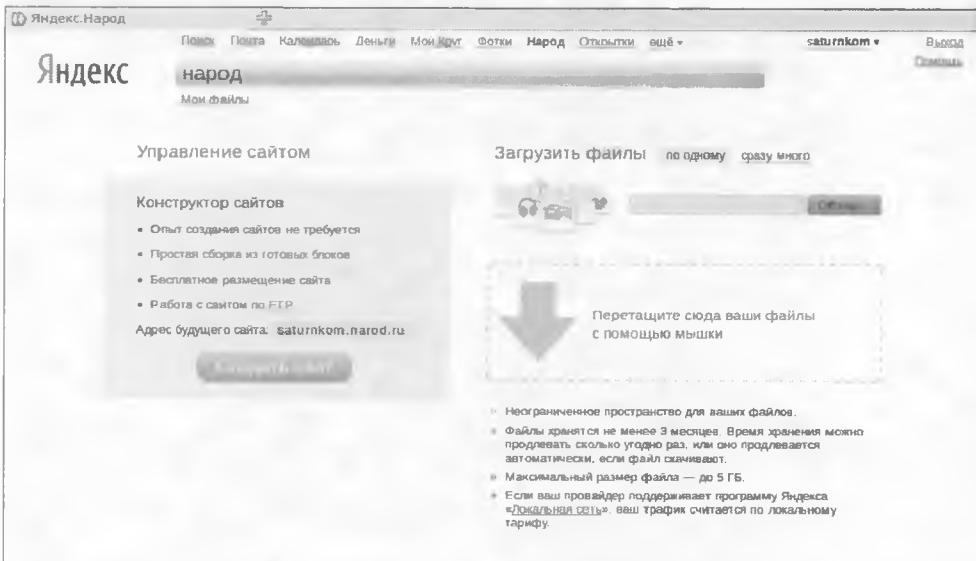


Рис. 3.41. Страница, определяющая возможность создания сайтов или загрузки файлов на сервис narod.yandex.ru

Если нажать кнопку Создать сайт, то будет представлена последовательность из нескольких шагов, позволяющая с помощью шаблонов создать свой сайт, например, как на рис. 3.42.

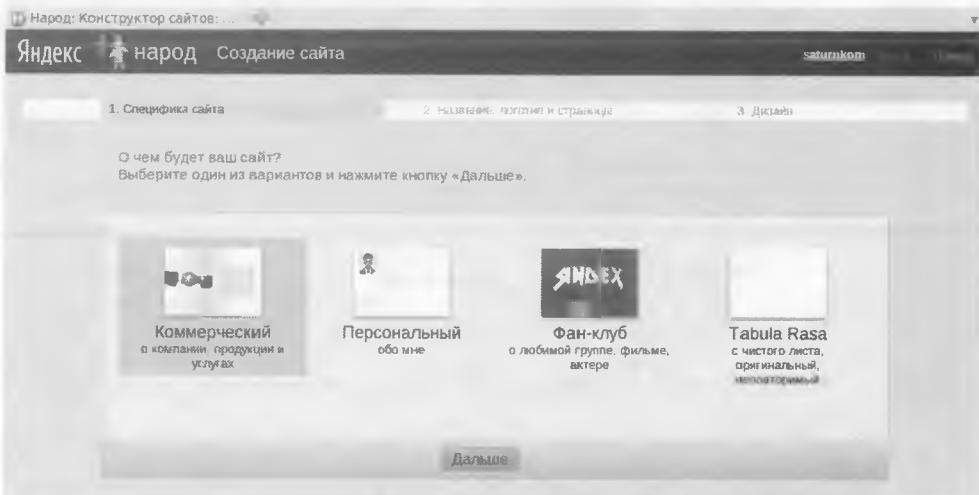


Рис. 3.42. Первый шаг по формированию сайта из имеющихся шаблонов непосредственно на хостинге сервиса narod.yandex.ru

Выбираем **Tabula Rasa** с чистого листа и жмем кнопку **Дальше** (рис. 3.43).

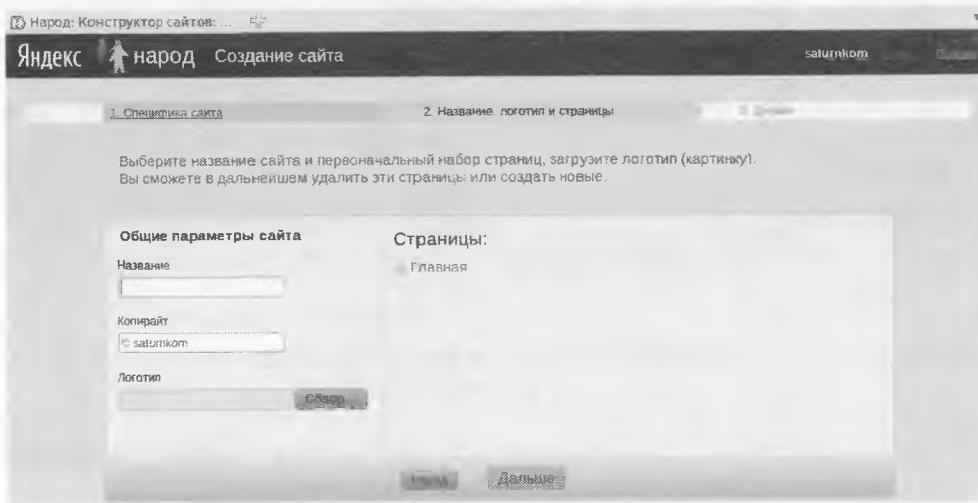


Рис. 3.43. Второе окно формирования сайта по шаблону на сервисе narod.yandex.ru

Ничего не заполняем и жмем кнопку **Дальше** (рис. 3.44).

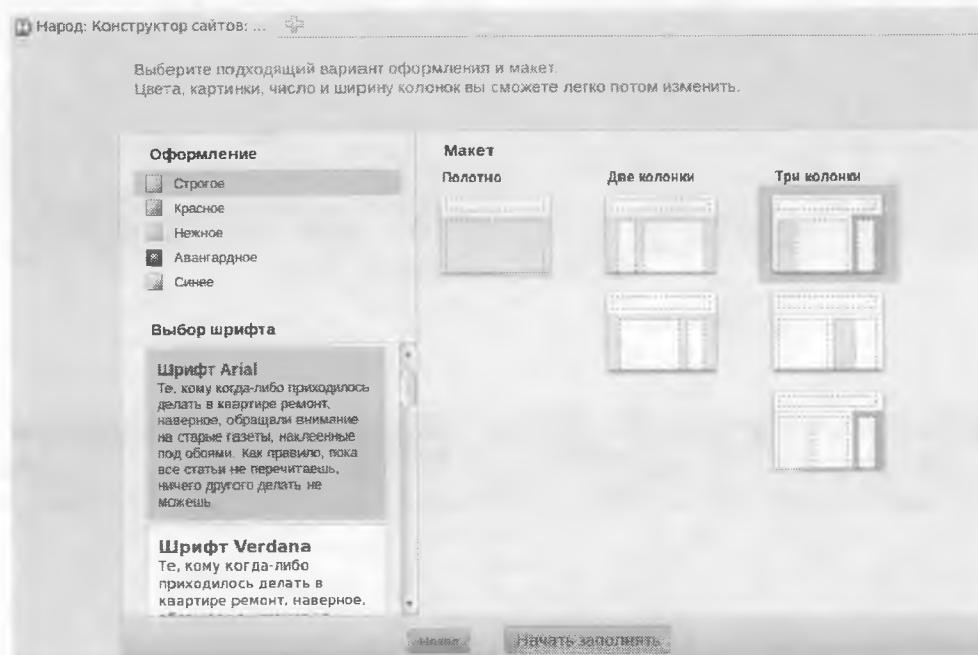


Рис. 3.44. Третий шаг в формировании сайта средствами шаблонов сервиса narod.yandex.ru

Нажимаем кнопку Начать заполнять и переходим к следующему экрану, где отображается сервис управления сайтом — так называемая мастерская (рис. 3.45).



Рис. 3.45. Мастерская управления сайтом на сервисе narod.yandex.ru

Выбираем в верхнем раскрывающемся меню конструктор сайтов пункт управление файлами (рис. 3.46).

Переходим к окну, в котором отображаются существующие файлы. Из них, как видим на рис. 3.47, присутствует только главная страница, генерированная автоматически, и файл robots.txt, записанный поисковой системой.

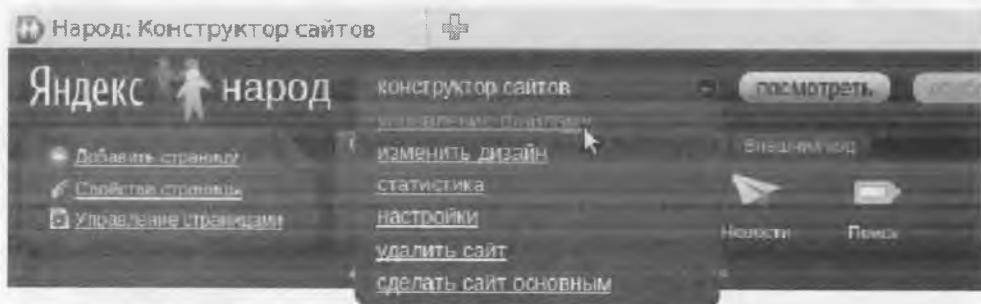


Рис. 3.46. Выбор пункта меню для управления файлами



Рис. 3.47. Окно, отображающее загруженные файлы

Щелкаем на ссылке Загрузить файл и в появившемся окне Обзор указываем на расположение нашей первой страницы index.html для ее загрузки на сервер. На это система управления выдаст предупреждение (рис. 3.48).

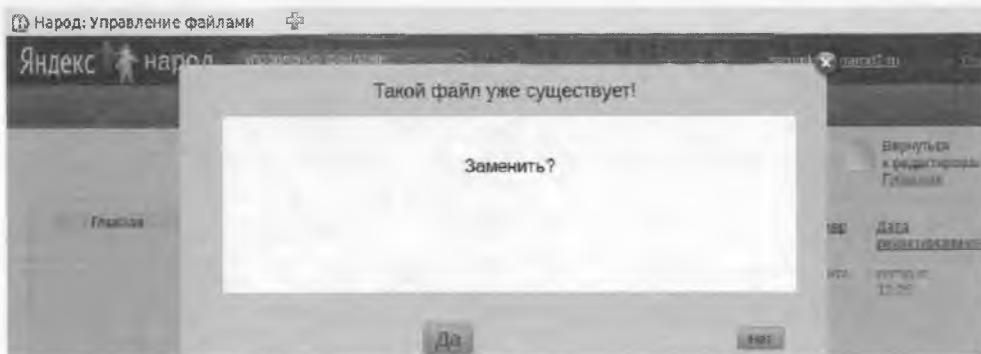


Рис. 3.48. Предупреждение системы о имеющемся на сервисе файле с аналогичным именем

Необходимо ответить Да, поскольку сгенерированная сервисом главная страница пуста и мы ее меняем на свою, созданную ранее. После загрузки страницы index.html окно браузера будет выглядеть, как на рис. 3.47. Однако, введя в адресной строке браузера доменное имя сайта <http://saturnkom.narod.ru/>, мы увидим созданную ранее страницу (рис. 3.49).

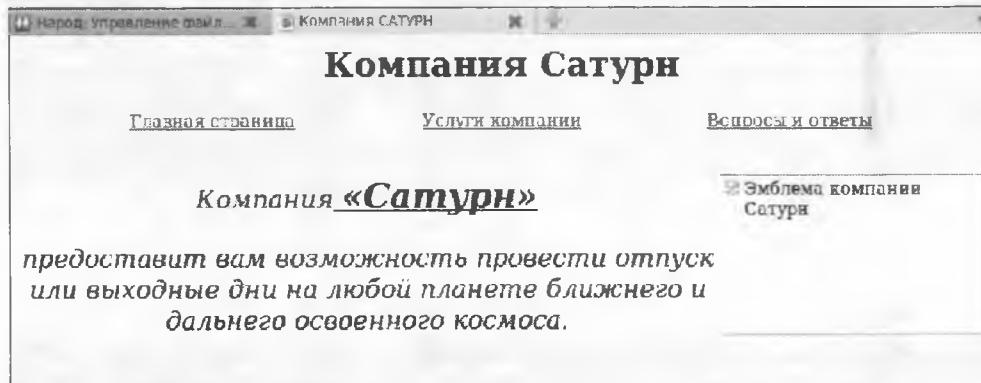


Рис. 3.49. Главная страница сайта, размещенная на веб-сервере

Обратите внимание, что вместо графической эмблемы нашей компании в зарезервированном для нее месте стоит замещающая надпись Эмблема компании Сатурн. Это естественно, поскольку на сервер загружена только одна страница, а нам необходимо загрузить всю структуру сайта. Выполнив описанные выше действия, загрузим страницы str2.html и str3.html, в этом случае окно браузера примет вид, как на рис. 3.50.

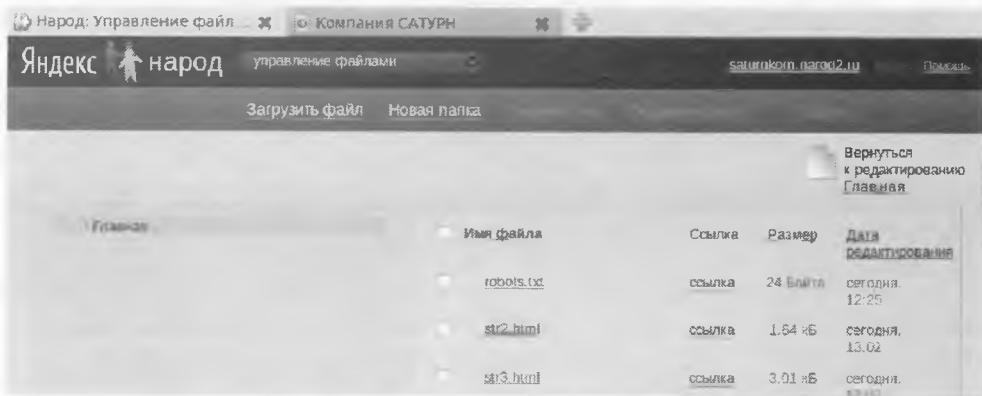


Рис. 3.50. Демонстрация загруженных страниц str2.html и str3.html

Для загрузки изображений необходимо создать папку на сервере с таким же именем, как и на нашем компьютере, — *graf*, для чего выберите ссылку Новая папка и введите ее имя в появившемся окне (рис. 3.51).

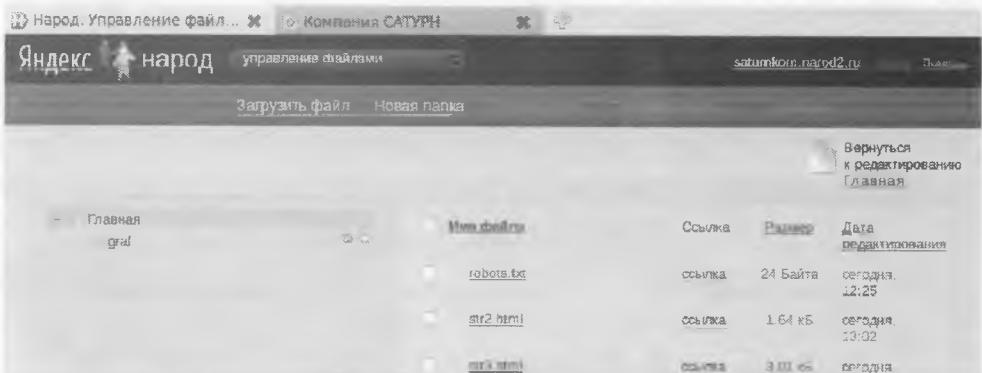


Рис. 3.51. Создание папки *graf*

Щелкнув на имени папки, откройте ее (рис. 3.52).

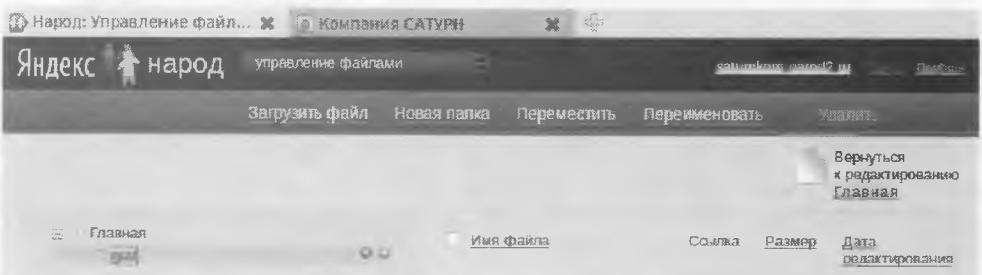


Рис. 3.52. Открытая папка *graf*

Выполнив вышеописанные действия по добавлению файлов, внесите в нее созданные файлы saturn.png и saturnfon.png, должно получиться, как на рис. 3.53.



Рис. 3.53. Демонстрация загруженных графических файлов saturn.png и saturnfon.png

Проверяем работоспособность сайта, переходя по ссылкам его навигационного меню (рис. 3.54).

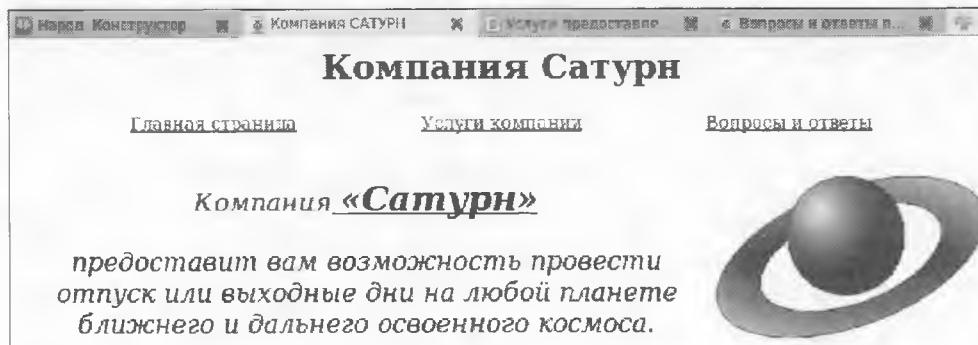


Рис. 3.54. Созданный сайт по адресу saturnkom.narod.ru

Таким образом, в течение двух часов с помощью книги и при наличии низкоскоростного доступа к Интернету создан и размещен на бесплатном хостинге трехстраничный сайт, при построении которого были использованы различные возможности языка HTML.

Проверив работоспособность сайта в Сети, можно закрыть мастерскую сервиса narod.yandex.ru, нажав кнопку Выход, и доложить начальству: «Задание выполнено, представительство в Сети есть». Потом можно закрыть эту книгу и поставить ее на полку до следующего аврала. А можно просто перевернуть страницу и продолжать знакомиться с возможностями сайтомостроения.

3.7. Введение в каскадные таблицы стилей CSS

С корабля на бал: рассмотрим примеры

На самом деле на современном этапе развития веб-технологий все меньше и меньше людей для создания и сопровождения своих сайтов идут тем путем, что был описан выше, то есть мало кто пишет сайт в текстовом редакторе вручную. Это не высокопроизводительный способ, занимающий очень много времени, которое просто уходит на набор и отладку кода. Такая же ручная работа по созданию стилистического оформления веб-страниц тоже малопроизводительна, ведь в визуальном HTML-редакторе весь набор, определяющий тот или иной стиль объекта, делается несколькими щелчками кнопкой мыши. Но было бы неправильно не дать краткого представления о каскадных таблицах стилей и возможности их применения при создании веб-страниц.

Итак, основная концепция каскадных таблиц стилей состоит в том, чтобы определить набор средств для описания внешнего вида отображения любых HTML-документов. Благодаря языку описания каскадных стилевых таблиц можно полностью управлять стилем и расположением каждого элемента веб-страницы, что является более простым и функциональным по сравнению с применением обычного набора HTML-тегов.

Рассмотрим простейший пример, чтобы понять, как это работает (рис. 3.55).

```

1  <html>
2  <head>
3
4  <style type="text/css">
5  <!--
6  mystyle :font-family:arial, Helvetica, sans-serif;
7  font-size: 16pt; font-style: italic;
8  font-weight: bold; color: #6699CC;
9  -->
10 </style>
11
12 </head>
13
14 <body>
15
16 <span class="mystyle"> текст оформленный согласно стилю mystyle</span>
17
18 </body>
19 </html>

```

Рис. 3.55. Фрагмент кода веб-страницы, содержащий описание стиля mystyle

На рис. 3.55 представлен следующий фрагмент кода веб-страницы:

```

<html>
<head>
<style type="text/css">
<!--

```

```
.mystyle {font-family:Arial, Helvetica, sans-serif;  
font-size: 16pt; font-style: italic;  
font-weight: bold; color: #6600CC}  
  
-->  
</style>  
</head>  
<body>  
<span class="mystyle"> текст оформленный согласно стилю mystyle</span>  
</body>  
</html>
```

Здесь нас интересуют только неизвестные нам строки, описывающие сам стиль и способы его применения, поэтому рассмотрим их более детально. Начальный тег определения стиля выглядит следующим образом:

```
<style type="text/css">
```

Он содержит атрибут `type="text/css"`, значение которого говорит браузеру о том, что далее будет следовать текст, оформленный согласно правилам языка CSS. Само же определение стиля помещается в скобки комментария языка HTML `<!--...-->`. Здесь в самом начале обязательно должен присутствовать селектор стиля, в нашем случае `.mystyle`, с обязательной точкой перед именем селектора. Далее в фигурных скобках перечисляются конкретные свойства данного стиля с заданными для них значениями в виде `свойство: значение`, где до двоеточия прописывается имя свойства, а после двоеточия — его конкретное значение. Таким образом, данный фрагмент кода можно описать так:

- `<!--` — начало описания стиля;
- `.mystyle` — селектор стиля;
- `{font-family:Arial, Helvetica, sans-serif;` — семейство шрифтов, которые наиболее предпочтительны для отображения данной страницы;
- `font-size: 16pt;` — размер шрифта, заданный в абсолютных единицах;
- `font-style: italic;` — стиль или начертание шрифта, в нашем случае это курсив;
- `font-weight: bold;` — «вес» шрифта, в нашем случае полужирный;
- `color: #6600CC}` — цвет шрифта, заданный в числовом виде в системе RGB;
- `-->` — конец описания стиля;
- `</style>` — завершающий тег описания стиля.

Для применения стиля `mystyle` к тексту, который необходимо отформатировать, например размещенному в элементе `span`, нужно непосредственно указать в коде данного элемента атрибут `class`, которому и присвоить значение в виде созданного нами стиля.

```
<span class="mystyle"> текст, оформленный согласно стилю mystyle</span>
```

Как видно из вышеописанного, присваивать значение созданного стиля можно любым текстовым фрагментам, включенным в текст веб-страницы. Далее при необходимости изменить, например, размер или начертание шрифта следует просто проделать это в самом коде описания стиля, и внесенные изменения отразятся на всех записях веб-страницы, оформленных в данном стиле.

Таким же образом, сгруппировав в единый файл стили, используемые для оформления сайта, можно одним росчерком пера изменить вид всего сайта.

Связывание стилевых таблиц с документом

Чтобы браузер узнал, откуда брать соответствующую данному файлу таблицу CSS, ему следует явно указать это. Всего различают четыре способа встраивания таблиц CSS в документ.

- **Внешние таблицы.** Таблица располагается в обычном текстовом файле, имеющем расширение CSS. В этом случае стилевой файл будет подключаться с помощью атрибута `href` тега `<link>`. Например, запись `<link rel="stylesheet" type="text/css" href="style.css">` определяет для браузера подключение файла `style.css` в качестве стилевого, а атрибуты тега `rel` и `style` указывают на то, что это таблица.
- **Внутренние таблицы.** В этом случае таблица располагается внутри веб-страницы, внутри тега `<head>` и, соответственно, отвечает за стилистическое оформление только данной страницы. Стиль записывается внутри парного тега `<style></style>`, где желателен атрибут `type="text/css"`, указывающий на то, что браузеру приходится иметь дело с каскадными таблицами стилей.
- **Строчное указание стиля.** Применяется, когда необходимо установить стилевое оформление для конкретной небольшой части документа, например надписи или заголовка. Этот вариант мы уже рассматривали, когда оформляли абзац текста (см. рис. 3.14 и 3.15).
- **Импортирование внешних таблиц стилей, находящихся на веб-сервере.** В этом случае в теге `<style>` с помощью свойства `@import` можно указать адрес расположения таблицы стилей на удаленном веб-сервере: `@import: URL (style.css)`. Главное — помнить, что запрос на импортируемую таблицу стилей необходимо задавать в самом начале стилевого блока.

Понятие класса

Как видно из приведенных выше примеров, обычно стиль задается для конкретных тегов, определяющих вид и структуру веб-страницы, но это возможно не всегда. Например, такой вариант не подойдет при необходимости по-другому выделить различные элементы или выровнять весь текст по ширине, а для абзацев предусмотреть красную строку и т. д. В этом случае прибегают к созданию классов (рис. 3.56).

```

<style>
H1,H2,H3 { text-align: center;
            color:#886600;
            font-family:Arial;
        }
.just {   text-align:justify;
            text-indent:3 em;
        }
.term {   color:red;
            font-size:110%;
            letter-spacing:0.1 em;
        }
</style>

```

Рис. 3.56. Определение классов объектов и их стилей

На примере рис. 3.56 видно, что помимо определения стилей для заголовков первого, второго и третьего уровней были заданы два дополнительных класса и определены их стили:

```

H1,H2,H3 { text-align: center;
            color:#886600;
            font-family:Arial;
        }
.just {   text-align:justify;
            text-indent:3 em;
        }

```

Второй класс задает выравнивание по ширине и создает отступ красной строки в 3 единицы (за ширину единицы принимают ширину латинской буквы *e*).

```

.term {   color:red;
            font-size:110%;
            letter-spacing:0.1 em;
        }

```

Этот класс выделяет термин красным цветом и шрифтом, увеличенным на 110 % от номинала, с разрядкой символов в объекте в 0,1 единицу.

Применим созданные классы стилей, например, как в этом листинге:

```
<h1>просмотр заголовка первого уровня</h1>  
<div class="just">просмотр применения стилистического оформления для объ-  
екта подвергшегося стилистическому оформлению класса just</div>  
<div class="just">поговорим про <span class="term">выделенные</span> сло-  
ва в дополнение обработанные стилем term</div>
```

На рис. 3.57 можно увидеть результат стилистического оформления различных элементов веб-страницы с помощью классов стилей.

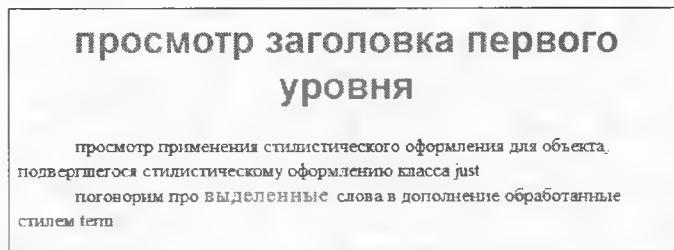


Рис. 3.57. Применение различных классов стилей для оформления элементов документа

Динамический HTML. Модель DOM

Из показанных примеров видно, что при большом количестве веб-страниц в информационном ресурсе технология применения CSS является более мощным инструментом стилистического управления сайтом, чем оформление элементов веб-страниц средствами чистого HTML. Но, как и любая другая технология, она требует более детального и глубокого изучения для последующего использования.

Казалось бы, зачем создавать сложности со всеми этими каскадными таблицами стилей?

Дело в том, что использование CSS в совокупности с языком гипертекстовой разметки HTML и языком программирования JavaScript, о котором речь пойдет в пятой главе, даст новую смесь возможностей для веб-программирования — динамический HTML (DHTML).

Динамический HTML на самом деле не является каким-либо языком, это просто термин, применяемый для обозначения HTML-страниц с динамически изменяемым содержимым окна браузера в ответ на какое-либо событие. Обычно DHTML рассматривают на основе следующей понятийной модели (рис. 3.58).

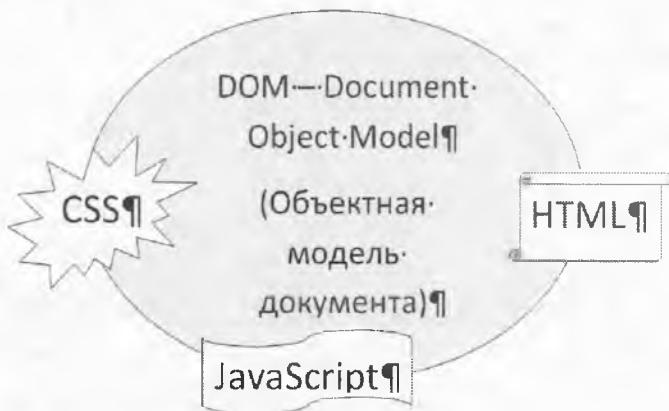


Рис. 3.58. Модель связи компонентов в динамическом HTML

Итак, как видно из рисунка, все три составляющих компонента, входящих в DHTML, связываются между собой объектной моделью документа — DOM. Она, в свою очередь, придает классическому HTML-документу новое качество — возможность изменять свое содержимое без перезагрузки страниц. Из приведенной модели также понятно, что каждый из входящих в DHTML элементов вносит свою лепту в это качество: HTML определяет так называемый базис веб-страницы — наполняет контентом и создает структуру, JavaScript, выполняя программный код, изменяет свойства ее элементов, а каскадные таблицы стилей устанавливают эти изменения для конкретного структурного элемента веб-страницы.

Таким образом, видно, что объектная модель документа DOM делает все элементы веб-страницы программируемыми объектами, то есть, получив через нее доступ к элементам страницы, можно управлять всем содержимым документа с помощью сценариев — маленьких программ на языке JavaScript. Попробуем понять работу DHTML на примере.

Представьте, что существует веб-страница, на которой находится определенное меню с пунктами и подпунктами. Если создавать такое меню методами, которые были рассмотрены в этой главе, все эти пункты и подпункты должны быть видимы постоянно, что загружает экран лишней информацией. Применение же технологии DHTML позволяет спрятать подпункты меню до тех пор, пока не будет выделен необходимый главный пункт, а затем они будут появляться каким-либо заранее предусмотренным способом. При этом пункты и подпункты меню, в которых вы уже побывали, стилем, цветом или начертанием шрифта будут отличаться от тех, которые еще не посещались. Стиль пунктов меню будет изменяться с помощью каскадных таблиц стилей, где веб-программистом заранее определено состояние стиля для каждого созданного класса. Применять же классы стилей для данного фрагмента HTML-кода будет язык программирования JavaScript согласно уже тем

сценариям, которые были написаны этим же или другим веб-программистом либо вообще взяты из сторонних библиотек.

Если с рассмотренным примером все достаточно понятно, попробуем углубиться в теорию дальше.

Как было сказано ранее, любое правило, создаваемое для каскадных таблиц стилей, состоит из двух частей: селектора и определения. Селектором может быть любой тег HTML-кода, для которого задаются способы форматирования его содержимого. Само же определение также состоит из двух частей: свойства и его значения. Они обязательно разделяются двоеточием, а группы «свойство — значение» отделяются друг от друга точкой с запятой. Обычно назначение свойства очевидно из его названия, а его значение задается в явном виде, например:

```
h3 {color: red;  
     font-size: 14pt}
```

Здесь `h3` — элемент, определяемый как селектор; `{color: red;` — свойство, задающее красный цвет; `font-size: 14pt}` — свойство, устанавливающее размер шрифта 14 пунктов для нашего селектора — заголовка третьего уровня. Но это мы рассматривали чуть выше, поэтому опишу следующие правила для селекторов — группирование и наследование.

Группирование и наследование в каскадных таблицах

Группирование выполняется достаточно просто, например:

```
h3 {color: red}  
h2 {color: red}  
h1 {color: red}
```

В свою очередь, это можно заменить на:

```
h3, h2, h1 {color: red}  
и  
h3 {color: red;}  
h3 {font-size: 14pt}
```

Что также можно заменить на:

```
h3 {color: red; font-size: 14pt}
```

Из приведенных примеров видно, что можно как задать определенное правило для группы селекторов, так и сгруппировать определения. Соответственно, можно группировать и селекторы, и определение вместе:

```
h3, h2, h1 {color: red; font-size: 14pt}
```

Благодаря такому подходу в группировке правил количество кода резко уменьшается, а управление оформлением сайта значительно упрощается.

С наследованием правил дело обстоит следующим образом: поскольку некоторые элементы веб-страницы могут содержать в себе другие, вложенные в них элементы, то вложенные дочерние элементы наследуют стилистическое оформление родительских элементов, если им не присвоено свое значение стиля. Это очень удобно при формировании общего стиля, задаваемого по умолчанию, например:

```
body {  
    color: blue;  
    font-family: Arial;  
    background: yellow;  
}
```

Поскольку определение относится к тегу `<body>`, то по умолчанию для содержимого страницы будут заданы следующие стилевые установки: голубой цвет шрифта страницы, тип шрифта Arial и желтый фоновый цвет самой веб-страницы. Здорово, правда? С одной стороны — да, а с другой? Например, определим для селектора A следующее правило:

```
A { text-decoration: none; }
```

Этим мы запретим подчеркивать все ссылки в документах, к которым подключен данный стиль. А что делать, если некоторые ссылки в определенных местах документа следует оформлять чуть по-другому? Если здесь отменить, то опять отменится во всем документе... Поэтому нам необходимо либо непосредственно прописать стиль в теге для конкретной ссылки, либо обратиться к параметру CLASS.

Параметр CLASS принимается спецификацией языка HTML в качестве стандарта для всех тегов, а его значением является ссылка на определенный класс, задаваемый в таблице стилей. Мы уже рассмотрели применение параметра CLASS на примере в начале раздела, сейчас же более подробно познакомимся с ним как с селектором.

Селектор CLASS

Определяемый класс позволяет задавать различные правила форматирования для одного элемента какого-либо типа или для всех элементов документа. Для этого имя создаваемого класса указывается в селекторе правила после имени тега и отделяется от него точкой. В нашем примере он ставится перед селекторами `.just` и `.term` определяющего имени тега, а это значит, что данные классы могут применяться ко всем элементам документа вне зависимости от тега.

```
H1,H2,H3 { text-align: center;  
            color:#886600;  
            font-family:Arial;  
        }  
  
.just { text-align:justify;  
         text-indent:3 em;  
        }  
  
.term { color:red;  
         font-size:110%;  
         letter-spacing:0.1 em;  
        }
```

Внесем изменения в оформление страницы, приведенной на рис. 3.57. Стилевое оформление текста будет выполняться следующим кодом:

```
<h1>просмотр заголовка первого уровня</h1>  
  
<div class="just">просмотр применения стилистического оформления для объ-  
екта подвергшегося стилистическому оформлению класса just</div>  
  
<div class="just">поговорим про <span class="term">выделенные</span> сло-  
ва в дополнение обработанные стилем term</div>
```

Видно, что применение правил относится только к выделенным тегам вне зависимости от того, являются объекты вложенными или нет. Главное, как был применен селектор CLASS.

Селектор ID

Один из важнейших параметров в HTML – ID, который, как и CLASS, не влияет на отображение браузером элемента веб-страницы, но позволяет задать для него уникальное имя в пределах всего сайта. Это имя будет использоваться как в сценариях, так и в каскадных таблицах стилей для индивидуального обращения к нему. Как и CLASS, селектор ID можно применять к любому элементу веб-страницы. Необходимо только предварять идентификационное имя, выбранное в качестве селектора, символом #. Например, рассмотрим уже известный пример с выделением определенного слова селектором term:

```
#term { color:red;  
         font-size:110%;  
         letter-spacing:0.1 em;  
        }
```

Но, как видно, теперь он у нас не будет относиться к классу, а будет иметь четко определенное идентификационное имя. Соответственно, вызов его в следующей строке:

```
<p>поговорим про <span id="term">выделенные </span> слова в дополнение  
обработанные стилем term</p>
```

даст результат, как на рис. 3.58, то есть слово «выделенные» будет написано красным цветом с увеличенным межбуквенным интервалом и более крупными буквами относительно общего текста фразы.

Применение таблиц стилей

Основные сведения о каскадных таблицах стилей изложены, и, в принципе, этого материала достаточно для умения прочитать имеющуюся таблицу, а при необходимости даже внести какие-либо изменения. Теперь, пожалуй, стоит задержаться на объяснении того, как именно применяются таблицы стилей, причем сохраняемые не в одном файле, а в нескольких, ведь недаром они названы каскадными. Данный материал будет полезен при изменении шаблонов сайтов, созданных на основе систем управления содержимым, которые описаны в главе, посвященной CMS Joomla!.

Как уже упоминалось выше, к одному документу можно присоединить несколько таблиц стилей, которые, в свою очередь, одновременно будут определять стилевое оформление данного документа. Это основополагающий принцип применения каскадных таблиц стилей, ради которого и затевался этот сыр-бор. Рассмотрим, что он позволяет сделать.

- Можно реализовать модульность оформления сайта, то есть для создания сайта можно написать несколько таблиц, каждая из которых будет отвечать только за свой фронт форматирования: одна таблица определяет стили шрифтов, другая описывает размещение объектов на веб-страницы, третья регламентирует состояние ссылок и т. д.
- Пожалуй, более важный пункт — определение равновесия между тем, как свой сайт видит автор и как его хочет видеть пользователь. Дело в том, что пользователь в состоянии самостоятельно определить правила отображения элементов сайта в таблицах стилей, применяемых браузером. Браузер, в свою очередь, на основании механизма приоритетности будет использовать те или иные правила стилистического оформления страницы, взяв их из таблиц автора сайта или настроек пользователя.

Следует знать, как взаимодействуют правила, определяющие стилистическое форматирование одинаковых элементов, но заданных в разных таблицах. Эти правила называются принципом приоритетности и заключаются в следующем: все таблицы располагаются друг за другом, образуя так называемый каскад приоритета — от низшего к наивысшему. По такому каскаду проходит обрабатываемый документ, постепенно изменяя свое форматирование с учетом правил, записанных в таблицах стилей.

Для определения приоритетности вводится понятие веса таблицы стилей. Соответственно, чем больше вес таблицы, тем выше ее приоритет. По умолчанию браузер принимает следующую приоритетность: вес каскадных таблиц стилей пользователя всегда меньше веса каскадных таблиц разработчика. Таким образом, в случае возникновения конфликта при определении стилистического оформления того или иного элемента веб-страницы последнее слово за веб-мастером, поэтому не следует этим злоупотреблять. Пользователю только и остается, что изменять настройки браузера по умолчанию.

В случае импортируемых стилевых таблиц их вес зависит напрямую от порядка импортирования в документ, каждая последующая импортируемая таблица имеет вес больший, чем у предыдущей. В свою очередь, импортируемые каскадные таблицы стилей сами могут импортировать другие таблицы.

Любые правила, определенные внутри документа, имеют наибольший вес по сравнению с любыми правилами, расположенными в таблицах за его пределами.

Разрешение внутренних конфликтов при определении приоритета правил, описанных в различных стилевых таблицах, осуществляется с помощью следующего алгоритма.

- Находятся все определения, относящиеся к данному правилу «элемент – свойство», которые рассматриваются только в том случае, если селектор правила соответствует рассматриваемому элементу. Если ни одно из определений невозможно применить, то используется наследуемое значение свойства, а если же наследуемое значение не устанавливается, применяют начальное значение данного свойства.
- Конфликтные определения сортируются по явно заданным весам.
- Определения подвергаются сортировке по происхождению: авторские, читательские, таблицы стилей, установленные по умолчанию в браузере.
- Далее происходит сортировка по специфичности селектора. Специфичность селектора определяется как сумма параметров CLASS, ID и название тегов в селекторе. Соответственно, чем эта сумма больше, тем выше специфичность селектора.
- И в заключение: определения сортируются по порядку их задания в таблице стилей – при конфликте двух правил с одинаковым весом наибольший приоритет имеет правило, заданное позже.

Если все это достаточно сложно, то попробуем проще расставить приоритетность правил по возрастанию.

- Связанная таблица стилей.
- Импортируемая таблица стилей.
- Правило с элементом языка HTML в качестве селектора.

- Правило с параметром CLASS в качестве селектора.
- Правило с параметром ID в качестве селектора.
- И как вершина — правило, встроенное в тег кода HTML.

Используя полный алгоритм либо упрощенную схему поиска приоритета правил, всегда легко найти в нескольких таблицах то правило, которое почему-то мешает оформлению элемента веб-страницы.

Данный краткий экскурс в каскадные таблицы стилей ни в коей мере не претендует даже на обзор всех возможностей CSS, но дает представление о том, что это такое и в каком направлении следует идти, чтобы сделать свой сайт более привлекательным для посетителей.

Глава 4

FTP-КЛИЕНТ, ЗАГРУЗКА ФАЙЛОВ НА СЕТЕВЫЕ СЕРВИСЫ

4.1. Знакомство с FTP-клиентом

Сегодня многие сетевые сервисы и компании, предоставляющие хостинг, предлагают пользователям не только услуги по размещению сайтов на их площадках, но и дисковое пространство для хранения файлов, вообще не имеющих отношения к размещенному на хостинге сайту.

На примере предыдущей главы в процессе размещения созданного сайта на сервисе narod.yandex.ru была использована возможность загрузки файлов на сервер с помощью HTTP-протокола, то есть непосредственно со страницы администрирования сайта. Следует отметить, что данный способ загрузки файлов на веб-сервис хотя и является приемлемым, но достаточно неудобен как с точки зрения возможностей загрузки, так и с точки зрения затрат времени на сам процесс загрузки и расходования трафика при загрузке абсолютно ненужного контента веб-страниц, сопровождающих данный процесс.

Можно ли решить задачу загрузки файлов на сервер без использования сервиса веб-страниц? Не только можно, но и нужно. Для этого вместо протокола HTTP, специально предназначенного только для передачи по Сети содержимого веб-страниц, необходимо перейти к использованию специального протокола FTP (File Transfer Protocol). Это протокол, созданный для передачи любых файлов между компьютерами вне зависимости от их содержимого. Любой современный сетевой сервис поддерживает данную клиент-серверную технологию, то есть на любом современном сетевом сервисе установлена специальная программа FTP-сервер для обслуживания потребностей пользователей через FTP, так как это экономит время и трафик не только пользователям, но и самому серверу, не перегружая его канал связи. Для работы с этим сервисом пользователю необходимо только установить специальную программу FTP-клиент, которая будет выполнять клиент-серверное взаимодействие с FTP-сервером.

Сегодня FTP-клиентов очень много: одни платные, другие бесплатные, третьи встроенные в различные файловые менеджеры или браузеры – пользователю предоставлен большой выбор этих незаменимых программ. Мы остановимся на бесплатном, мультиязычном – поддерживающем работу на многих языках, кроссплатформенном – работающем на базе различных

операционных систем FTP-клиенте FileZilla, который является проектом с открытым кодом. Скачать русскоязычную версию программы можно с официального русскоязычного зеркала программы <http://filezilla.ru/> (рис. 4.1).

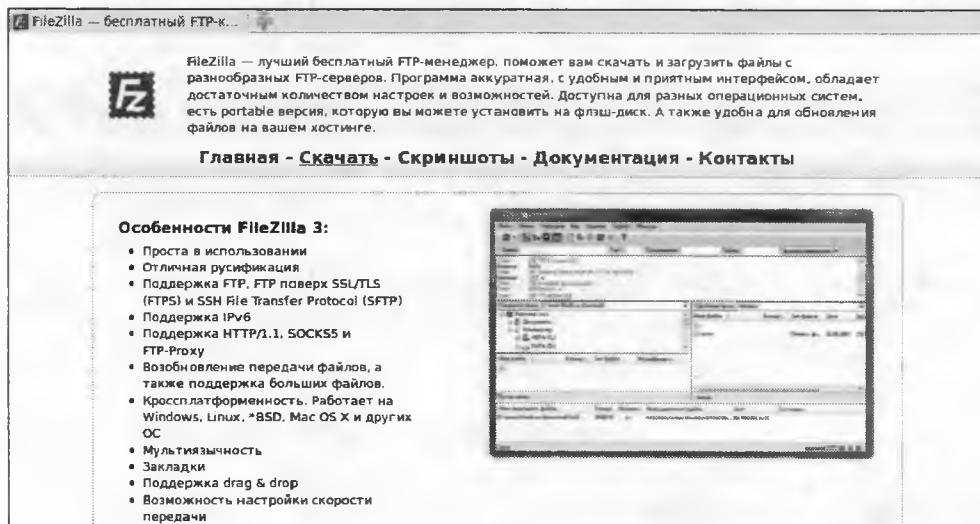


Рис. 4.1. Официальное русское зеркало проекта FileZilla

Перейдя по ссылке Скачать, окажемся в окне браузера, где будет предложено выбрать программу для той операционной системы, в которой и будет произведена установка (рис. 4.2).

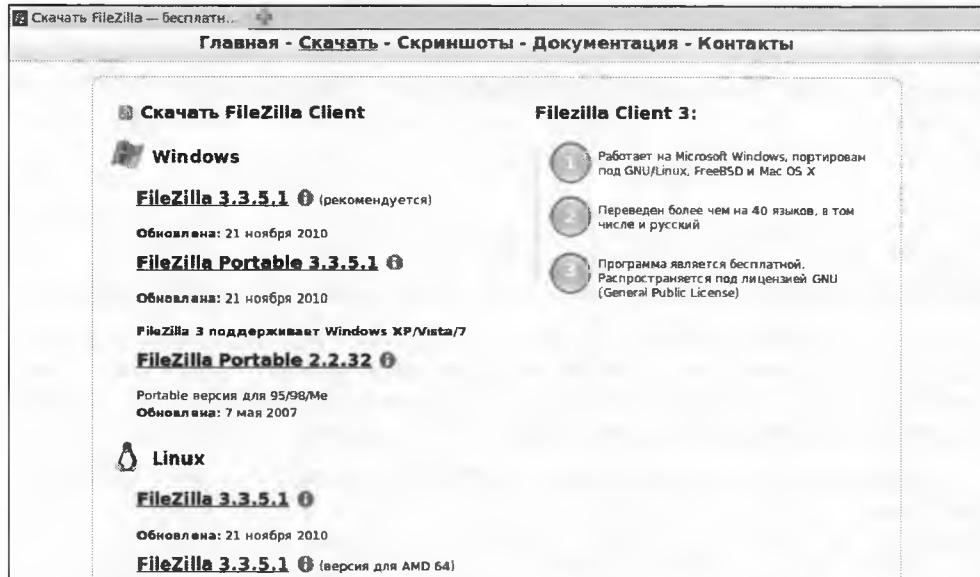


Рис. 4.2. Окно выбора версии программы под установленную операционную систему

4.2. Установка и настройка FileZilla

Выбрав версию, скачайте и установите программу стандартным для вашей операционной системы способом. После установки запустите ее на выполнение, и откроется примерно такое окно, как на рис. 4.3.

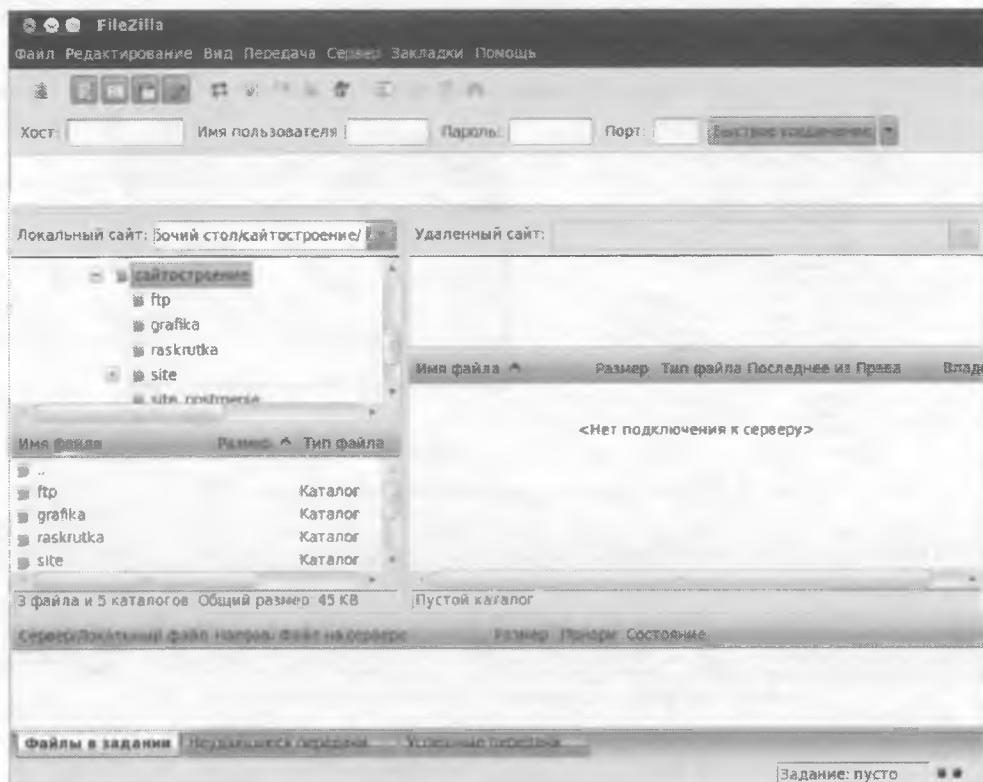


Рис. 4.3. Окно FTP-менеджера FileZilla

Почему примерно? Пока книга готовится к печати, могут произойти различные изменения в графическом интерфейсе программы, и, кроме того, в различных операционных системах программа также выглядит не совсем одинаково.

Программа установлена, и, прежде чем начать разбираться с ее интерфейсом, необходимо подключиться к сетевому ресурсу, с которым будет происходить взаимодействие. После подключения данный интуитивно понятный интерфейс программы принять и освоить будет гораздо проще.

Для подключения FTP-клиента к FTP-серверу необходимо знать следующее.

- Доступ к FTP-серверу, или хосту.
- Порт, по которому осуществляется подключение.

- Тип установленного сервера.
- Тип входа на сервер.
- Имя (логин) пользователя.
- Паспорт (пароль доступа к сайту) пользователя.

Откуда же взять эти данные, необходимые для подключения? В справочной системе компании-хостера, в нашем случае на странице <http://help.yandex.ru/narod/?id=1112370&ncrnd=7437> (рис. 4.4).

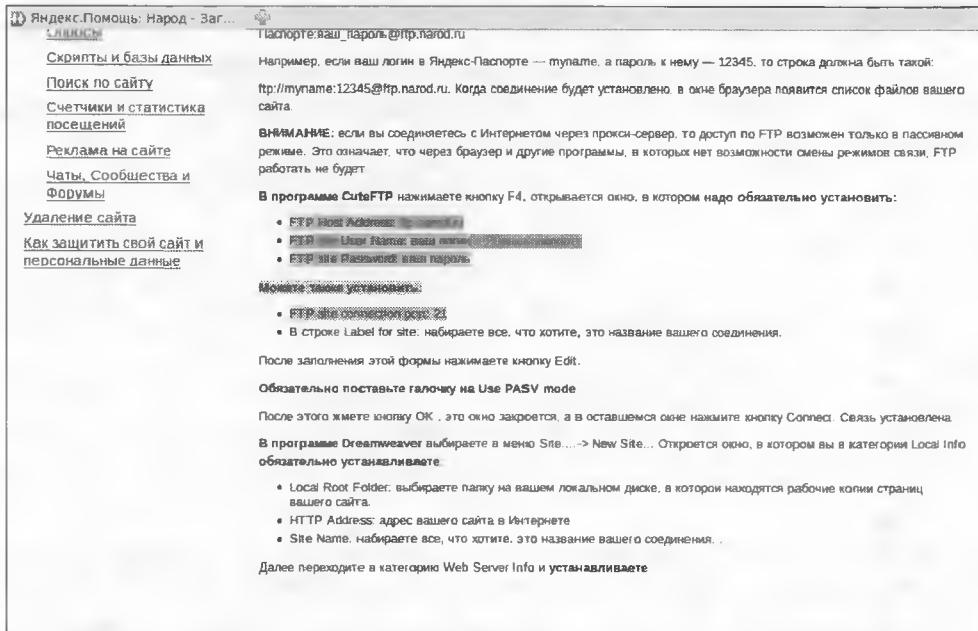


Рис. 4.4. Страница помощи. Описание подключения FTP-клиента (содержимое страницы и ее адрес могут изменяться в процессе модификации сервиса)

Данные, которые необходимы для подключения к серверу, прописаны хостером на этой странице. Если что-то из перечисленного в списке не указано на странице помощи, оставляйте настройки клиента по умолчанию.

Из справочной информации мы получили следующее.

- Доступ к FTP-серверу, или хосту, — ftp.narod.ru. Этот хост действителен для сайтов, созданных до 2010 года. Современный, на момент написания главы, хост будет выглядеть так: saturnkom.ftp.narod.ru. Как видим, появилось требование установки логина перед хостом.
- Порт, по которому осуществляется подключение, — 21.
- Тип установленного сервера — ftp.

- Тип входа на сервер – не указан, значит, выбираем **Нормальный**.
 - Имя (логин) пользователя – берем из регистрационных данных – **saturnkom**.
 - Паспорт (пароль доступа к сайту) пользователя – тоже знаем – *********.
- Пора приступать к настройке клиента FileZilla.

Откроем в программе FileZilla меню **Файл** и выберем пункт **Менеджер сайтов**, в результате чего должно появиться окно менеджера сайтов (рис. 4.5).



Рис. 4.5. Окно менеджера сайтов программы FileZilla

Нажмем кнопку **Новый Сайт** и заполним предложенные поля теми данными, которые знаем. Результат заполнения представлен на рис. 4.6.

Осталось только нажать кнопку **Соединиться**, и если есть подключение к Интернету, то получим следующее состояние окна программы (рис. 4.7).

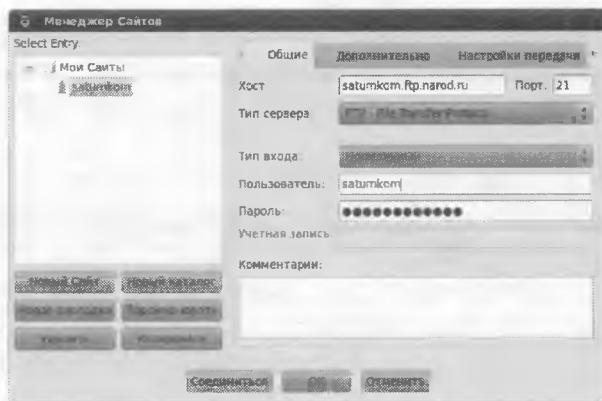


Рис. 4.6. Заполненное окно менеджера сайтов программы FileZilla

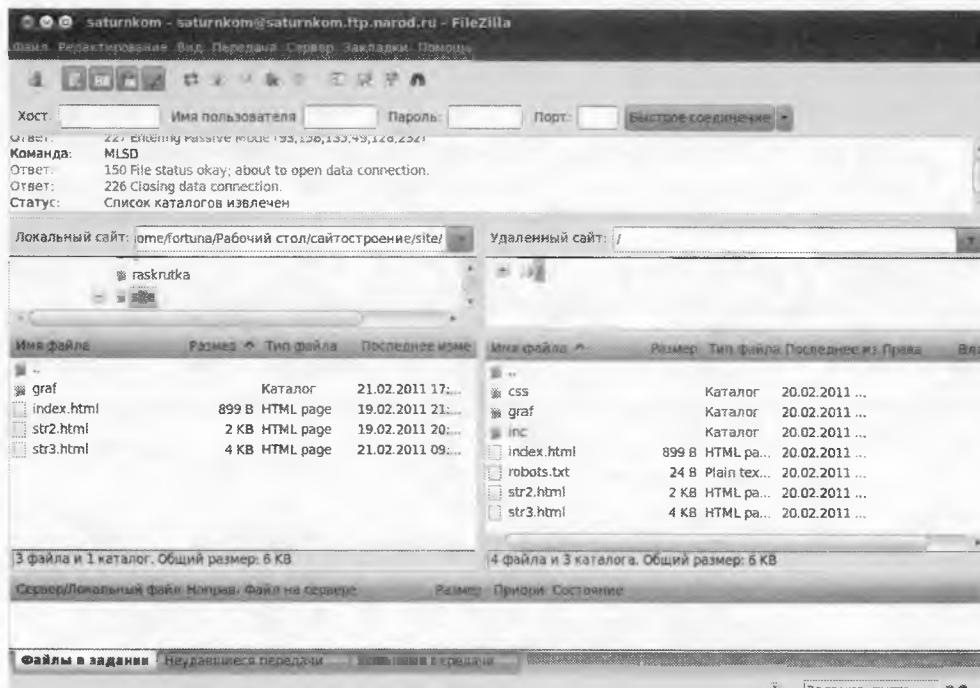


Рис. 4.7. Окно FileZilla при установленном соединении с веб-сервером по FTP-протоколу

4.3. Размещение файлов сайта на удаленном компьютере с помощью FileZilla

Как легко заметить, интерфейс программы интуитивно понятный. Окно, находящееся под строкой меню, показывает процесс соединения с сервером, левая панель **Локальный сайт** отображает каталоги, расположенные на локальном компьютере, а правая панель **Удаленный сайт** показывает содержимое каталогов на веб-сервере. Нижняя область активизируется в процессе приема-передачи файлов и демонстрирует ход выполнения процесса. Переиная вкладки нижней панели **Файлы в задании**, **Неудавшиеся передачи** и **Успешные передачи**, можно отследить ход выполненных заданий по перекачке файлов.

Как видно из рис. 4.7, слева прописаны файлы, находящиеся на нашем компьютере, и файл `str3.html` имеет более позднее время модификации, чем файл, находящийся в папке веб-сервера. Чтобы заменить файл на веб-сервере, необходимо сделать следующее.

1. Выделить с помощью мыши файл, находящийся на компьютере.
2. Щелкнув правой кнопкой мыши на выделенном файле, открыть его контекстное меню.
3. В контекстном меню выбрать пункт Закачать на сервер либо Добавить в задание, если связь еще не установлена. Процесс выполнения начнет демонстрироваться в нижней части окна программы (рис. 4.8).

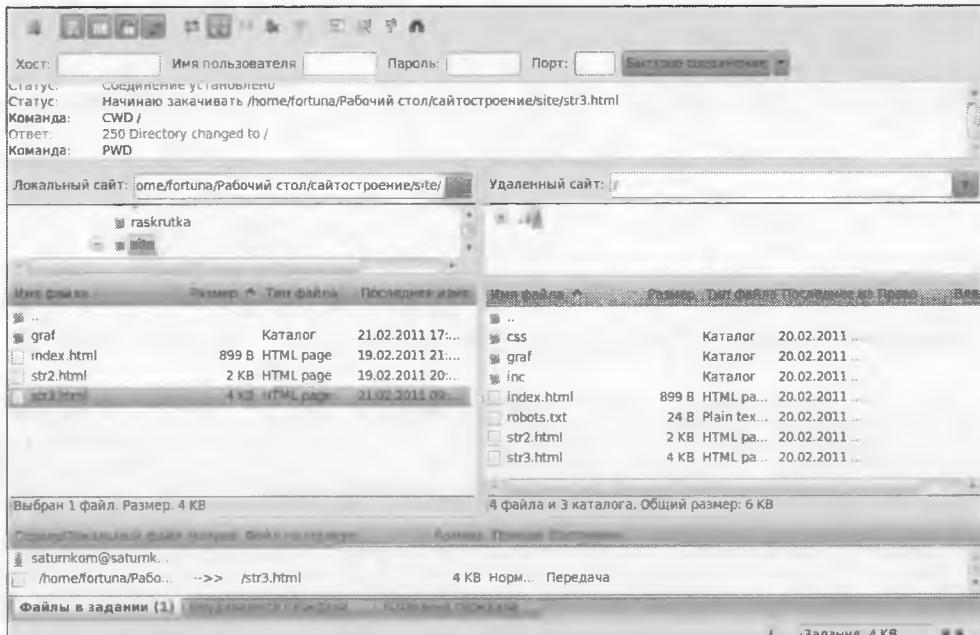


Рис. 4.8. Выполнение задания по размещению файлов на веб-сервере

Поскольку файл с именем `str3.html` на веб-сервере уже существует, то программа резонно поинтересуется у вас, что же делать с этими двумя файлами (рис. 4.9).

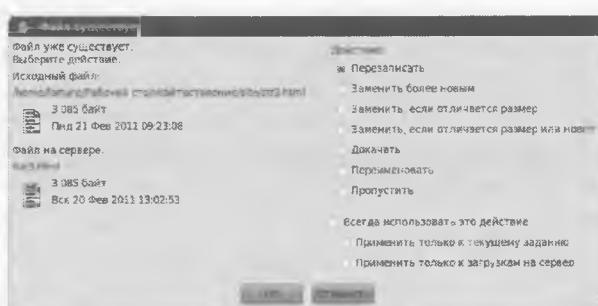


Рис. 4.9. Запрос программы на выполнение действий с одноименными файлами

Ответим утвердительно на этот запрос и нажмем кнопку **OK**. При этом после завершения передачи на вкладке **Успешные передачи** появится цифра 1 — как информация об успешно выполненном задании (рис. 4.10).

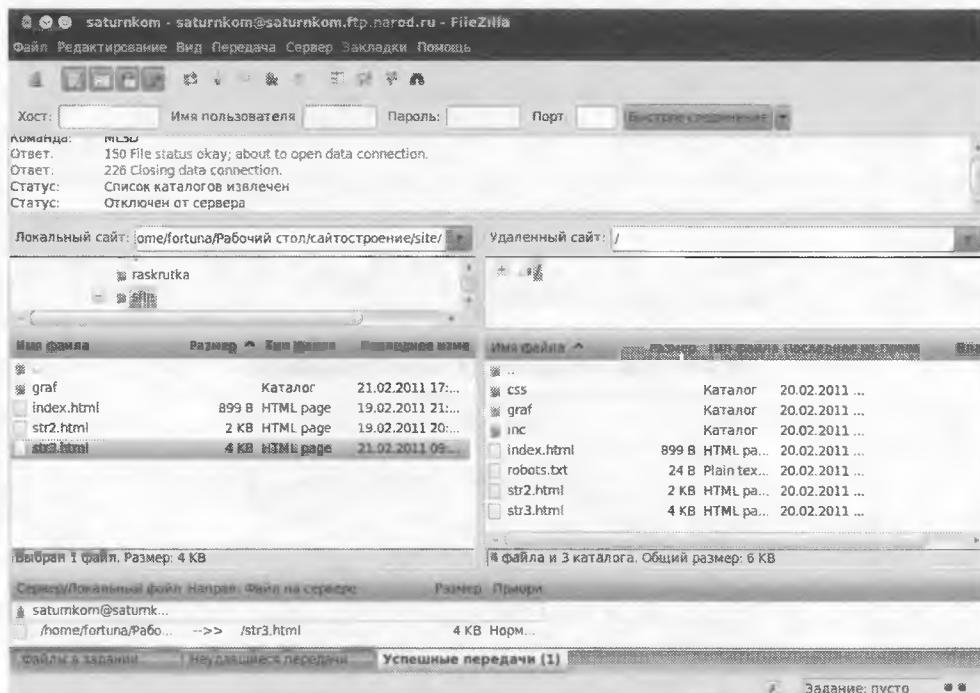


Рис. 4.10. Итоговое окно программы при успешной загрузке файла на веб-сервер

Как видите, все очень просто и достаточно быстро. С помощью программы можно не только размещать файлы на удаленных серверах, но и скачивать их оттуда. Во втором случае описанные выше действия выполняются уже над файлами, находящимися на правой панели программы.

Следует заметить, что достоинства FileZilla этим не исчерпываются. Познакомимся с еще одной замечательной способностью программы, которая понадобится нам в скором времени, — это особенность задавать права доступа к файлам.

Для пользователей операционных систем UNIX все понятно, а для пользователей Windows поясню. Дело в том, что подавляющее количество веб-серверов в Сети работает под управлением UNIX или Linux, которые принципиально отличаются в подходе к безопасности по сравнению с Windows. В связи с этим, если вы желаете, чтобы ваш файл, содержащий, например, прайс-лист продукции, был скачан пользователем с сайта и беспрепятственно прочитан, необходимо задать права доступа пользователей на этот файл.

Эти права сделают его доступным для всех, то есть равным 777 при цифровом задании прав доступа. Программа FileZilla позволяет устанавливать права доступа к файлам, даже не зная, что это такое.

Для установки прав доступа к файлу, находящемуся на сервере, необходимо сделать следующее.

1. Выделить нужный файл на правой панели программы.
2. Щелкнув на файле правой кнопкой мыши, вызвать его контекстное меню.
3. Выбрать пункт меню Права доступа к файлу.

В результате выполнения описанной последовательности действий появляется окно Изменить атрибуты файла (рис. 4.11).

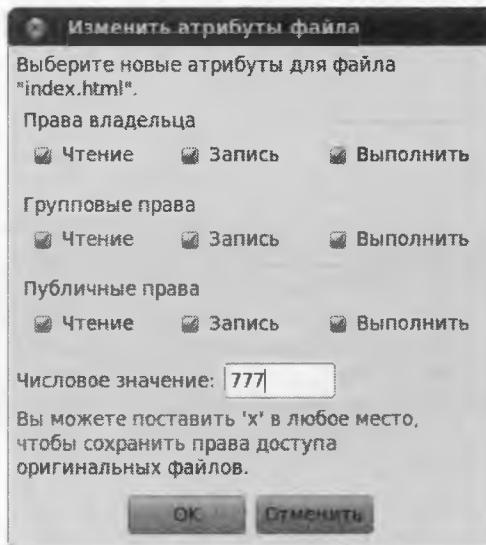


Рис. 4.11. Окно изменения атрибутов файла программы FileZilla

В появившемся окне можно задать необходимые права доступа к файлу, просто устанавливая соответствующие флагшки либо прописав числовое значение в поле. Рекомендация на будущее — если вы не уверены в правильности задания прав доступа к файлу, то не изменяйте их просто так. Если же файл должен быть скачан и открыт, то установите полный доступ, прописав 777 в поле ввода числового значения.

Использование FTP-клиентов не только ускоряет и упрощает работу с файлами, находящимися на удаленных компьютерах, но и позволяет избежать множества ошибок, появляющихся при пересылке файлов по HTTP-протоколу, и в целом делает работу строителя сайтов более комфортной.

Глава 5

ПРОГРАММИРОВАНИЕ НА JAVASCRIPT

5.1. Знакомство с языком

Обычно под выражением «оживление сайта» подразумевают приятие ресурсу интерактивности, чтобы было ощущение общения, а не простого просмотра картинок на экране. Таким образом, для решения данной задачи необходимо найти какие-то элементы, которые и будут создавать у пользователя иллюзию того, что сайт — не просто статический набор страниц, а активно и постоянно работающая система, готовая либо тут же автоматически выполнить пожелание гостя, либо комфортно для него сообщить о его желаниях администратору.

Элементами интерактивности сайта занимаются маленькие программы — так называемые сценарии. Они, в свою очередь, могут выполняться двумя способами. В первом случае пользователь дает задание браузеру, браузер формирует запрос серверу, сервер подготавливает ответ и отправляет его браузеру, который и демонстрирует его пользователю. Недостатки очевидны: на сервере должна быть программа, обрабатывающая запросы, то есть транслятор языка программирования, обычно это PHP; время пересылки запросов и ответов увеличивается вместе с увеличением трафика Сети и ее загрузкой на пересылку служебной информации.

В нашем же случае на данной стадии развития проекта «Компания „Сатурн“» при его размещении на бесплатном сервисе yandex.narod.ru мы лишены права использовать язык PHP, так как его интерпретатор не установлен на веб-сервере, соответственно, и запросы, написанные на PHP, сервер обрабатывать не сможет. Поэтому мы пойдем другим путем: будем использовать сценарии, написанные непосредственно в теле страницы и исполняемые самим браузером в процессе ее обработки. Такой способ оживления веб-страницы требует применения языка JavaScript, разработанного в 1995 году и в нашем случае как нельзя кстати подходящего.

JavaScript, несмотря на простоту, — все-таки язык программирования. Работа с ним подразумевает определенные, но в то же время обилие в Интернете готовых сценариев позволяет пользователю, не вдаваясь в суть программирования, вставлять их в тело страницы. Поскольку данная книга не учебник по программированию, то вдаваться в глубины языковых конструкций мы не будем, а рассмотрим основные понятия, позволяющие прочитать и использовать в теле веб-страницы готовые сценарии.

Редактор Notepad++

Перед тем как приступить к работе со сценариями, написанными в JavaScript, рекомендуется немного улучшить свое рабочее место, то есть заменить малофункциональный текстовый редактор Блокнот чем-либо более удобным, например редактором Notepad++. Проект с открытым кодом Notepad++ является бесплатным, и страница его загрузки располагается по адресу <http://notepad-plus-plus.org/download>. Хотя Notepad++ не является кроссплатформенным, поскольку разрабатывается специально под операционную систему Windows, он без особых проблем устанавливается и на Linux под Wine. Достоинств у Notepad++ достаточно много: это и подсветка синтаксиса, и работа с кодировками, и многое другое, позволяющее облегчить жизнь начинающему веб-разработчику. Имеет смысл скачать его установочный файл с сайта проекта и установить как обычную программу согласно требованиям вашей операционной системы. При первом запуске программа выдаст окно, как на рис. 5.1.

Это значит, что программа успешно функционирует. Теперь можно приступить к освоению построения сценариев на JavaScript.

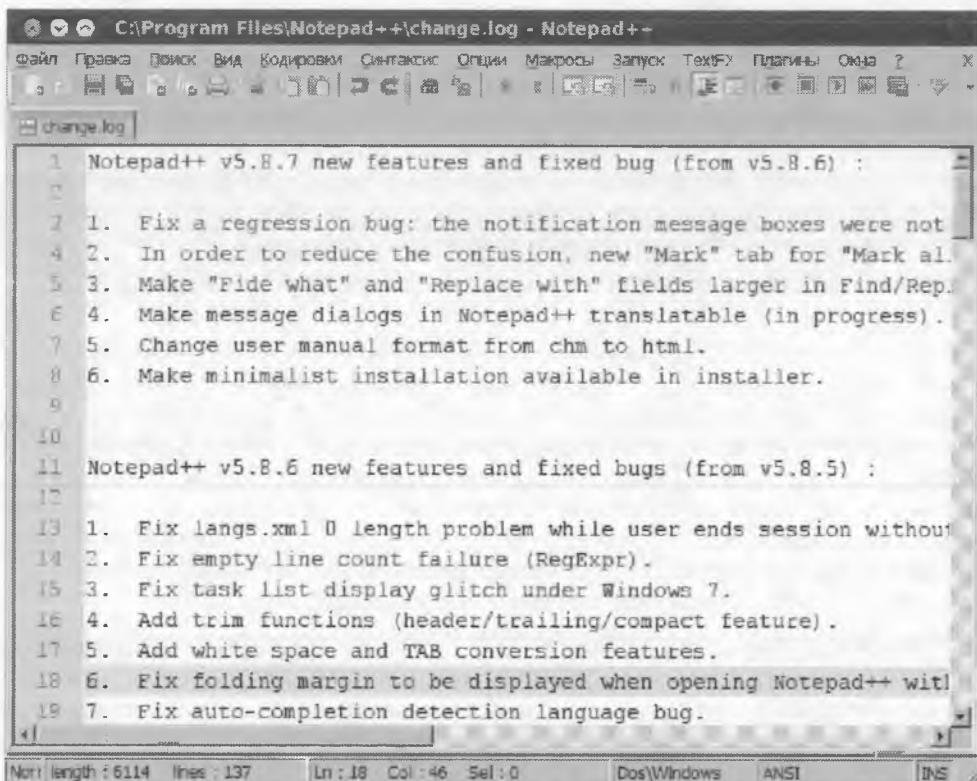


Рис. 5.1. Окно программы Notepad++ при первом запуске

Основные моменты при создании сценариев на JavaScript

Для начала необходимо разобраться с понятием объектно-ориентированного событийного программирования, поэтому, не углубляясь в дебри, определим для себя следующее.

- Любая программа, а в нашем случае сайт, как инструкция браузеру состоит из программных объектов: страницы, таблицы, рисунка и т. д.
- Программный объект, в свою очередь, обладает свойствами: имеет размер, цвет и пр., что отражается в описывающих его значениях переменных.
- Помимо свойств программный объект имеет еще и методы, определяющие то или иное его поведение и меняющие значения свойств, например изменение размеров окна, цвета текста, перемещение над объектом указателя мыши и т. д. Таким образом, свойства и методы объекта объединены вместе — это называется инкапсуляцией, то есть методы как бы окружают свойства объектов, не позволяя изменять их напрямую, а только через использование самих методов.
- И наконец, событие — это какое-то сообщение программному объекту, что необходимо изменить с помощью определенных методов значения переменных в теле объекта, чтобы изменились его некоторые свойства. Приведу пример: обычно указатель мыши имеет вид стрелки, но при перемещении на область ввода текста (текстовые поля) он принимает вид текстового курсора. Иначе говоря, при совершении события — перемещении указателя в определенную область экрана — были задействованы определенные методы, которые изменили свойства объекта «указатель», изменив его внешний вид.

Из приведенной информации необходимо понять следующее: при построении в теле веб-страницы программных объектов нужно описывать дополнительные инструкции для конкретных объектов и их поведения. Это могут быть размеры, цвет и форма самих страниц или их областей, поведение указателя мыши в определенных частях объекта или за его пределами, формирование каких-либо опросов на основе форм и организации процедур их обработки.

Язык JavaScript чувствителен к регистру, значит, при написании кода страницы этому нужно уделить повышенное внимание, так как строчные и прописные буквы алфавита считаются разными символами.

Конструкции JavaScript — это набор операторов языка в теле страницы, последовательно обрабатываемых встроенным в браузер интерпретатором. Операторы, в свою очередь, могут располагаться в одной строке — при этом они должны разделяться точкой с запятой (;) — либо отдельно на каждой строке по одному оператору — в этом случае наличие ; необязательно, но желательно, так как облегчает чтение кода сценария для пользователя.

Текстовое содержимое — так называемые строковые литералы — должно быть размещено на одной строке либо разбито на несколько строк. Во втором случае для объединения текста используют операцию объединения (конкатенации) нескольких строк в одну.

Символом // обозначаются комментарии к коду, которые игнорируются интерпретатором в ходе выполнения сценария. В случае многострочных комментариев используют символы /* и */, то есть комментарий выглядит так: /* комментарии */.

Возможны следующие способы размещения сценариев на веб-странице.

- Использование тега <script>тело сценария</script>. Это парный тег-контейнер, внутри которого и расположено само тело сценария. Применение такой методики встраивания сценария в любое место веб-страницы допустимо, только если вы не используете какие-либо глобальные переменные. В ином случае их необходимо описать в разделе <head> веб-страницы, ведь браузер интерпретирует страницу сверху вниз. В том же разделе <head> следует описывать процедуры и функции обработки форм или других данных.
- Использование отдельного файла с кодом сценария JavaScript. Для этого в теге <script> указывают параметр src, связывающий ссылку на сценарий с внешним файлом, содержащим программный код сценария. Значением параметра является полный или относительный URL файла с телом сценария, например <script src="http://.../.../telo.js"></script>. При этом файл, содержащий тело сценария, должен иметь расширение .js и не должен содержать HTML-тегов.
- Непосредственное размещение элементов языка JavaScript в параметрах тегов HTML. Иными словами, переменные и выражения языка используются в качестве параметров тегов на веб-страницах.

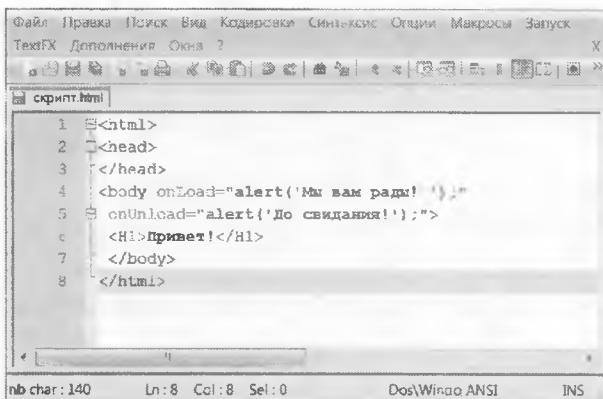
Итак, с лирикой покончено и пора переходить к практике. В силу того что книга имеет ограниченный объем и достаточно обзорный характер, полную спецификацию языка JavaScript при необходимости проще найти в Сети. Главная же задача главы — на примерах научиться читать и использовать уже существующие сценарии.

5.2. Простейшие сценарии

Включение сценария в тег <body>.

Всплывающее сообщение, метод alert()

Рассмотрим простейший пример использования элементов языка JavaScript в качестве параметров тега <body> (рис. 5.2).



```

Файл Правка Поиск Вид Кодировка Синтаксис Опции Макросы Запуск
TextFX Дополнения Окна ?
X
 скрипт.html
1 <html>
2 <head>
3 </head>
4 <body onLoad="alert('Мы вам рады!');"
5 onUnload="alert('До свидания!');"
6 <h1>Привет!</h1>
7 </body>
8 </html>

```

nb char: 140 Ln: 8 Col: 8 Sel: 0 Dos\Windows ANSI INS

Рис. 5.2. Код сценария тега <body>

Как видно из кода, параметрами тега <body> стали события onLoad и onUnload, соответственно обозначающие открытие и закрытие веб-страницы с надписью Привет. При открытии страницы обработчик события onLoad вызовет функцию alert, которая, в свою очередь, выдаст надпись Мы вам рады! (рис. 5.3).

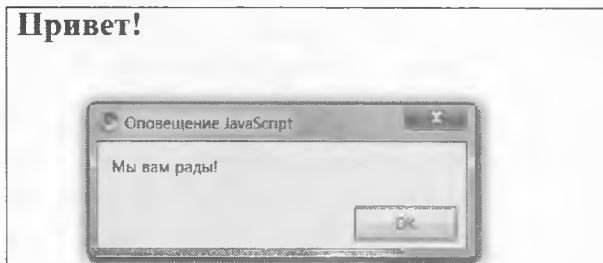


Рис. 5.3. Выполнение события onLoad при открытии страницы

В случае же закрытия страницы обработчик события onUnload вызовет ту же функцию alert, но уже с параметром 'До свидания!' (рис. 5.4).

Все достаточно просто, не правда ли? А можно ли поменять что-либо в тексте самой страницы? Конечно.

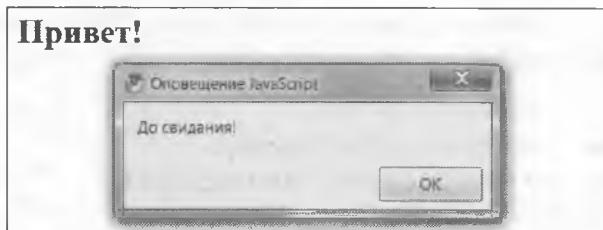


Рис. 5.4. Выполнение события onUnload при закрытии страницы

Вернемся к спецификации HTML и определим еще один не рассмотренный ранее тег `<div></div>`. Парный тег `<div>` предназначен для создания на странице разделов. Такие разделы можно делать вложенными, поэтому соблюдение парности тега «открыл — закрыл» обязательно для исполнения.

Изменение надписи в теле страницы при изменении фокуса

Рассмотрим возможность изменения содержимого надписи на странице в зависимости от того, является ли страница активной (рис. 5.5).

```

1 <html>
2 <head>
3 </head>
4 <body>
5 onClick="nadpis.align='center';"
6 onDoubleClick="nadpis.align='right';"
7 onBlur="nadpis.innerText='Зачем вы покидаете нашу страничку? Оставайтесь с нами';"
8 onFocus="nadpis.innerText='Мы рады, что вы вернулись к нам!';"
9 <div id=nadpis>
10 </div>
11 <h1>Привет!</h1>
12 </div>
13 </body>
14 </html>

```

Рис. 5.5. Обработчик событий активной и неактивной страницы

Рассмотрим данный листинг подробнее.

- `onClick="nadpis.align='center';"` — обработчик события, по щелчку кнопкой мыши центрирующий содержимое переменной `nadpis`.
- `onDoubleClick="nadpis.align='right';"` — обработчик события, по щелчку кнопкой мыши выравнивающий по правому краю содержимое переменной `nadpis`.
- `onBlur="nadpis.innerText='Зачем вы покидаете нашу страничку? Оставайтесь с нами';"` — обработчик события, выводящий надпись `Зачем вы покидаете нашу страничку? Оставайтесь с нами`, если щелчок кнопкой мыши произошел за пределами окна страницы.
- `onFocus="nadpis.innerText='Мы рады, что вы вернулись к нам!';"` — обработчик события, выводящий надпись `Мы рады, что вы вернулись к нам!`, если щелчок кнопкой мыши произошел в пределах окна страницы, и передающий параметры выравнивания надписи к вышеописанным обработчикам в зависимости от того, одинарный или двойной был щелчок.

- <div id=nadpis> – создание специального раздела для размещения на странице переменной nadpis. Параметр id тега <div> – обязательный и определяет переменную, которая будет выведена на страницу по вышеописанным правилам.

Таким образом, при одинарном щелчке в окне веб-страницы мы получим страницу, как на рис. 5.6

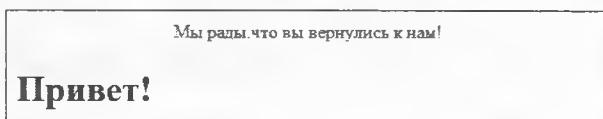


Рис. 5.6. Результат одинарного щелчка кнопкой мыши на странице

При двойном щелчке получим ту же надпись, только уже выровненную по правому краю (рис. 5.7).

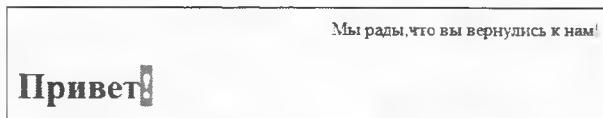


Рис. 5.7. Двойной щелчок на странице выравнивает надпись по правому краю

Если же щелкнуть кнопкой мыши за пределами страницы, получим надпись, вопиющую о возврате (рис. 5.8).

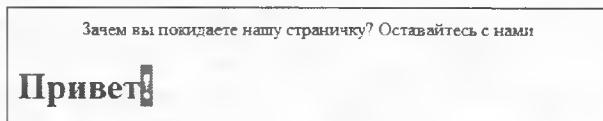


Рис. 5.8. Действие обработчика событий при уходе со страницы

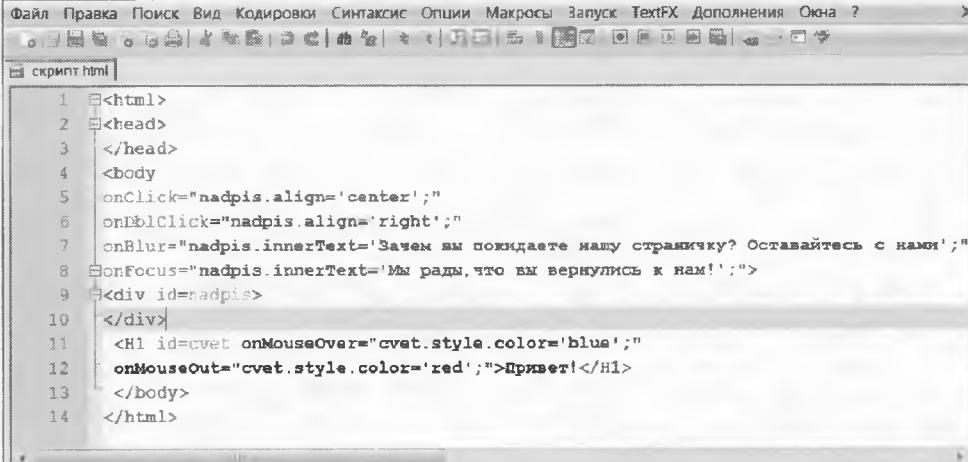
Изменение цвета надписи при наведении на нее указателя мыши

На примере рис. 5.9 разберем изменение цвета заголовка Привет! в зависимости от того, наведен ли на него указатель мыши.

Рассмотрим подробнее код сценария.

- <h1 id=cvet> – идентификатор, определяющий свойство объекта.
- onMouseOver="cvet.style.color='blue';" – событийная процедура, изменяющая цвет надписи Привет! на голубой при наведении указателя мыши.

- `onMouseOut="cvet.style.color='red';">` — событийная процедура, выделяющая надпись Привет! красным цветом при отсутствии над ней указателя мыши.
- `Привет!</H1>` — сама надпись и закрывающий тег.



```

Файл Правка Поиск Вид Кодировки Синтаксис Опции Макросы Запуск TextFX Дополнения Окна ?
 скрипт.html
 1 <html>
 2 <head>
 3 </head>
 4 <body>
 5 onClick="nadpis.align='center';"
 6 onDblClick="nadpis.align='right';"
 7 onBlur="nadpis.innerText='Зачем вы покидаете нашу страничку? Оставайтесь с нами!';"
 8 onFocus="nadpis.innerText='Мы рады, что вы вернулись к нам!';"
 9 <div id=nadpis>
10 </div>
11 <H1 id=cvet onMouseOver="cvet.style.color='blue';"
12 onMouseOut="cvet.style.color='red';">Привет!</H1>
13 </body>
14 </html>

```

Рис. 5.9. Листинг для изменения цвета надписи в зависимости от положения указателя мыши

Как видно, в языке JavaScript много возможностей для изменения объектов при совершении различных событий, вызываемых простым перемещением указателя мыши по поверхности страницы. Это позволяет украсить даже обычные пункты меню, задавать им различные цвета при наведении на них указателя мыши. Но это еще далеко не все возможности языка. Продолжим их рассмотрение на примерах взаимодействия с пользователем, то есть ответим на вопрос: каким образом получить исходные данные у пользователя и выдать ему результат.

5.3. Формы

Примеры работы с формами. Сложение двух чисел

По традиции рассматривать работу пользователя с формами в учебниках языков программирования начинают с создания калькулятора. Мы также не отступим от этого великолепного обычая и постараемся создать страницу, на которой будут складываться два числа. Заботу о взаимодействии с пользователем берет на себя объект «форма», определяемый парным тегом `<form></form>`. В нашем случае это не будет исключением (рис. 5.10).



```

1 <html>
2 <head>
3 </head>
4 <body>
5 <form>
6 <div align=center>
7 <input name=a>
8 <input name=b>
9 <input name=c>
10 </div>
11 <div align=center>
12 <input type=button value=+ onClick="c.value=a.value+ b.value">
13 </div>
14 </form>
15 </body>
16 </html>

```

Рис. 5.10. Листинг страницы, на которой выполняется сложение двух чисел

Рассмотрим листинг более подробно.

- <form> — начало объекта «форма».
- <div align=center> — определение центрирования раздела, содержащего вводимые данные и получаемый результат.
- <input name=a> — ввод в текстовое поле числа a.
- <input name=b> — ввод в текстовое поле числа b.
- <input name=c> — ввод в текстовое поле числа c.
- </div> — закрытие первого раздела.
- <div align=center> — определение центрирования раздела, содержащего кнопку со знаком +.
- <input type=button — определение типа объекта как стандартной кнопки.
- value=+ — определение надписи на кнопке как +.
- onClick= — определение событийной процедуры, выполняющейся при щелчке кнопкой мыши.
- "c.value=a.value+ b.value"> — присвоение текстовому полю с суммы текстовых полей a и b.
- </div> — закрытие второго раздела.
- </form> — закрытие объекта «форма».

Поскольку код листинга рассмотрен подробно, обратимся к результату его выполнения (рис. 5.11).



Рис. 5.11. Результат сложения двух чисел

Как видим из рис. 5.11, задание на сложение двух чисел выполнено неправильно, то есть оно выполнено буквально — сложены строковые значения переменных *a* и *b*, а это простая конкатенация строк. Это, конечно, неплохо, но нам необходима математическая операция сложения. Для этого мы должны преобразовать строковое значение переменной в числовое с помощью функции *parseInt*. Таким образом, код сценария примет вид согласно рис. 5.12.

```

Файл Правка Поиск Вид Кодировка Синтаксис Опции Макросы Запуск TextFX Дополнения Окна ?
 скрипт.html
 1 <html>
 2   <head>
 3   </head>
 4   <body>
 5     <form>
 6       <div align=center>
 7         <input name=a>
 8         <input name=b>
 9         <input name=c>
10       </div>
11     <div align=center>
12       <input type=button value=+ onClick="c.value=parseInt(a.value)+parseInt(b.value)">
13     </div>
14   </form>
15 </body>
16 </html>

```

Рис. 5.12. Изменение кода для выполнения арифметической операции сложения

Как видно из кода сценария, измененная строка имеет следующий вид:

```
<input type=button value=+ onClick="c.value=parseInt(a.value)+  
parseInt(b.value)">
```

Результат выполнения сценария можно наблюдать на рис. 5.13.

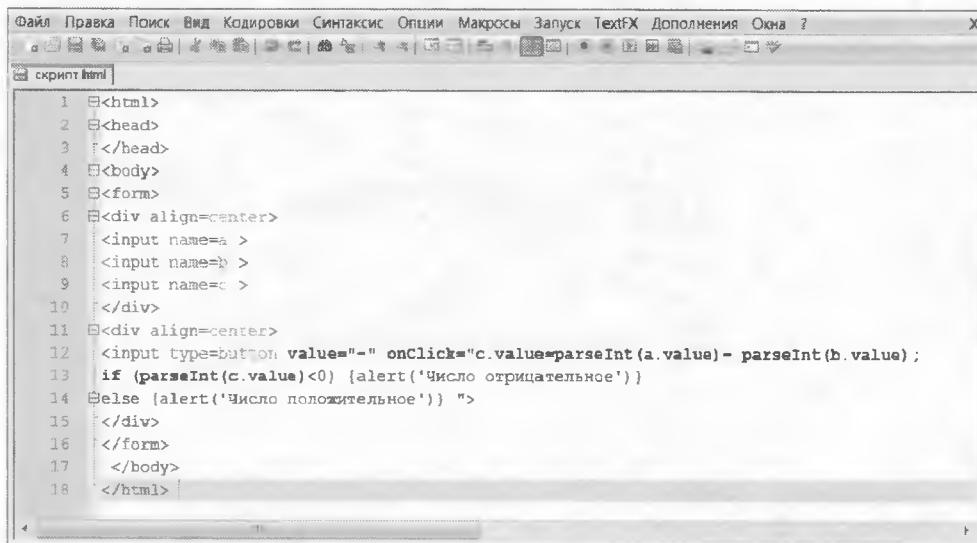


Рис. 5.13. Выполнение арифметического сложения двух чисел с помощью сценария

Выше описан пример линейного алгоритма, но линейных алгоритмов в практике программирования используется немного, большую часть все же занимают структуры ветвлений и циклы.

Условный оператор при выборе последовательности действий

Теперь имеет смысл рассмотреть применение ветвлений в коде сценария. Усложним предыдущую задачу и вместо суммы определим разность. При этом в зависимости от того, какое число будет получено — положительное или отрицательное, будем выводить сообщение о том, какое это число (рис. 5.14).



Файл Правка Поиск Вид Кодировка Синтаксис Опции Макросы Запуск TextFX Дополнения Окна ?

скрипт.html

```

1 <html>
2 <head>
3 </head>
4 <body>
5 <form>
6 <div align=center>
7 <input name=a>
8 <input name=b>
9 <input name=c>
10 </div>
11 <div align=center>
12 <input type=button value="-" onClick="c.value=parseInt(a.value) - parseInt(b.value);
13 if (parseInt(c.value)<0) {alert('Число отрицательное')}
14 else {alert('Число положительное')} ">
15 </div>
16 </form>
17 </body>
18 </html>

```

Рис. 5.14. Пример сценария, содержащего структуру ветвлений

Рассмотрим более подробно строки сценария, непосредственно решавшие поставленную задачу.

- <input type=button value="-" onClick="c.value=parseInt(a.value) - parseInt(b.value); — определяет разность двух чисел a и b.
- if (parseInt(c.value)<0) {alert('Число отрицательное')} — начало ветвлений, if можно понимать как слово «если», далее идет логическое условие, определяющее истинность или ложность выражения «число с меньше нуля?». Если ответ — истина, выводится сообщение Число отрицательное, в противном случае переходим к следующей строке.
- else {alert('Число положительное')} — продолжение ветвлений, else определяется как слово «иначе», таким образом определяя

действия на тот случай, если логическое условие будет ложным. В нашем примере будет выведено сообщение Число положительное.

- ">" — скобка, завершающая код сценария.

Посмотрим иллюстрацию работы сценария на рис. 5.15 и 5.16.

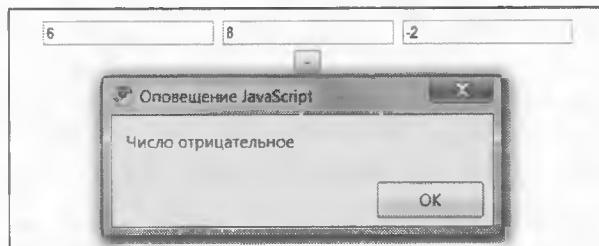


Рис. 5.15. Определение и вывод сообщения об отрицательном числе в результате решения задачи

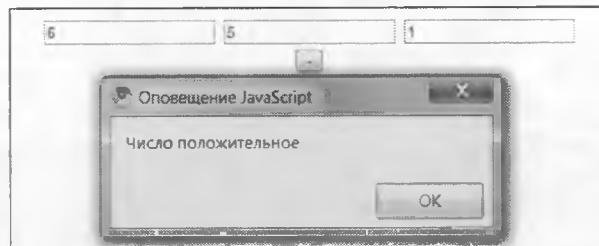


Рис. 5.16. Определение и вывод сообщения о положительном числе в результате решения задачи

Элементы формы для создания и обработки тестового опроса. Функции

Рассмотрим более сложную задачу на определение какого-либо результата, полученного по итогам тестового опроса. Например, создадим список вопросов с различными вариантами ответов (рис. 5.17). Отвечая на вопрос, посетитель получает соответствующий ответ.

Рассмотрим листинг более подробно.

- `<html>` — определение документа как веб-страницы.
- `<head>` — определение заголовка документа.
- `<script>` — начало сценария. Он вынесен в заголовок страницы, так как при обращении к функции сценария браузер уже должен знать, где она находится. На самом деле сценарий с функцией можно прописывать в любом месте документа, но если используются глобальные переменные, все же желательно прописывать в заголовке документа.
- `function proverka()` — начало функции.

- { — фигурная скобка обязательна, так как показывает начало функции. По окончании функции ставится закрывающая фигурная скобка.
- result = 0 — обнуление переменной, подсчитывающей количество правильных ответов.

```

1 <html>
2 <head>
3 <script>
4 function proverka() {
5   result = 0
6   if (document.test.aa[2].checked) result++
7   if (document.test.bb[0].checked) result++
8   if (document.test.cc.value==3) result++
9   alert("Правильных ответов " + result + " из трех возможных" )
10 </script></head>
11 <body>
12 <form name="test">
13 <div><b>Вопрос 1.</b> Какая кнопка в правом верхнем углу окна его закрывает?<br>
14 <input type=radio name=aa> Кнопка "мусс"<br>
15 <input type=radio name=aa> Кнопка "квадрат" <br>
16 <input type=radio name=aa> Кнопка "крестик"</div><br>
17 <div><b>Вопрос 2.</b> В случае нажатия кнопки "минус" произойдет:<br>
18 <input type=checkbox name=bb>Сворачивание окна с Рабочего стола<br>
19 <input type=checkbox name=bb>Изменение размера окна<br>
20 <input type=checkbox name=bb>Закрытие приложения</div> <br>
21 <div><b>Вопрос 3.</b> При решении следующей задачи:  $2+2*2$  ответ равен:<br>
22 <select name=cc>
23 <option value=1>
24 <option value=2>8
25 <option value=3>6
26 </select> </div><br>
27 <input type=button value="Проверить" onClick="proverka();">
28 </form> </body></html>

```

Рис. 5.17. Код для обработки запроса и вывода результата

- if (document.test.aa[2].checked) result++ — условие (если в текущем документе в форме с именем test активизирован второй элемент массива с именем aa, то увеличить значение переменной result на единицу). В свою очередь, чтобы было более понятно, надпись result++ равносильна надписи result=result+1. Если оператор не один, то каждый должен заключаться в фигурные скобки {}.
- if (document.test.bb[0].checked) result++ — условие (если в текущем документе в форме с именем test активизирован нулевой элемент массива с именем bb, то увеличить значение переменной result на единицу).
- if (document.test.cc.value==3) result++ — условие (если в текущем документе в форме с именем test значение выбранного элемента массива с именем cc равно 3, то увеличить значение result на единицу). Обратите внимание, что равенство обозначается двойным знаком равно ==.

- `alert("Правильных ответов " + result + " из трех возможных")` — вывести надпись с указанием количества правильных ответов.
- `}` — фигурная скобка, закрывающая тело функции.
- `</script></head>` — теги окончания тела сценария и заголовка документа соответственно.
- `<body>` — начало тела документа.
- `<form name="test">` — открытие формы элементов с именем `test`.
- `<div>Вопрос 1.` Какая кнопка в правом верхнем углу окна его закрывает?`
` — начало первого раздела с указанием текста вопроса.
- `<input type=radio name=aa>` Кнопка "минус"`
` — создание переключателя `radio` с именем `aa` и надписью рядом с ним.
- `<input type=radio name=aa>` Кнопка "квадрат" `
` — аналогично предыдущей строке.
- `<input type=radio name=aa>` Кнопка "крестик" — аналогично предыдущим двум строкам.
- `</div>
` — закрытие первого раздела, перевод строки.
- `<div>Вопрос 2.` В случае нажатия кнопки "минус" произойдет:`
` — начало второго раздела с указанием текста вопроса.
- `<input type=checkbox name=bb>` Сворачивание окна с Рабочего стола`
` — создание с независимой фиксацией (флажок) `checkbox` с именем `bb` и надписью рядом с ним.
- `<input type=checkbox name=bb>` Изменение размера окна`
` — аналогично предыдущей строке.
- `<input type=checkbox name=bb>` Закрытие приложения — аналогично предыдущим двум строкам.
- `</div>
>` — закрытие второго раздела, перевод строки.
- `<div>Вопрос 3.` При решении следующей задачи: $2+2*2$ ответ равен:`
` — начало третьего раздела с указанием текста вопроса.
- `<select name=cc>` — создание раскрывающегося списка с именем `cc`.
- `<option value=1>` — присвоение первому элементу списка значения «не определено», или «пусто».
- `<option value=2>8` — присвоение второму элементу списка значения 8.
- `<option value=3>6` — присвоение третьему элементу списка значения 6.
- `</select>` — завершение списка.

- `</div>
` – конец третьего раздела, перевод строки.
- `<input type="button" value="Проверить" onClick="proverka();">` – определение командной кнопки с надписью Проверить, при нажатии которой активизируется функция `proverka()`.
- `</form> </body> </html>` – завершение формы, завершение тела документа, конец веб-страницы.

Рассмотрим, на что нужно обратить внимание при использовании элементов данного листинга.

- В скобках рядом с именем функции можно ставить передаваемые ей параметры или оставлять их пустыми, но они должны быть.
- В случае вызова в функции значения элемента, используемого в форме, необходимо указывать полный адрес данного элемента: `документ.имя_формы.имя_элемента`.
- Тело функции начинается с открывающейся скобки `{` и заканчивается закрывающейся `}`, как и совокупность нескольких операторов — так называемый составной оператор, применяемый в структурах ветвления или цикла.
- При использовании однотипных элементов в форме (в нашем случае это фляжки и переключатели) с одинаковыми именами необходимо помнить, что они считаются элементами массива и нумеруются, начиная с нуля, а не с единицы.
- В структуре ветвления может быть использовано составное логическое условие. Главное при этом — не запутаться в скобках: количество открытых скобок всегда должно равняться количеству закрытых.
- Таким образом, при правильном ответе на все предложенные вопросы пользователь получит страницу, представленную на рис. 5.18.

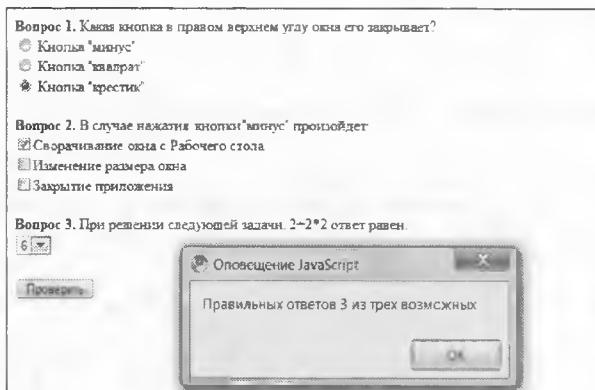


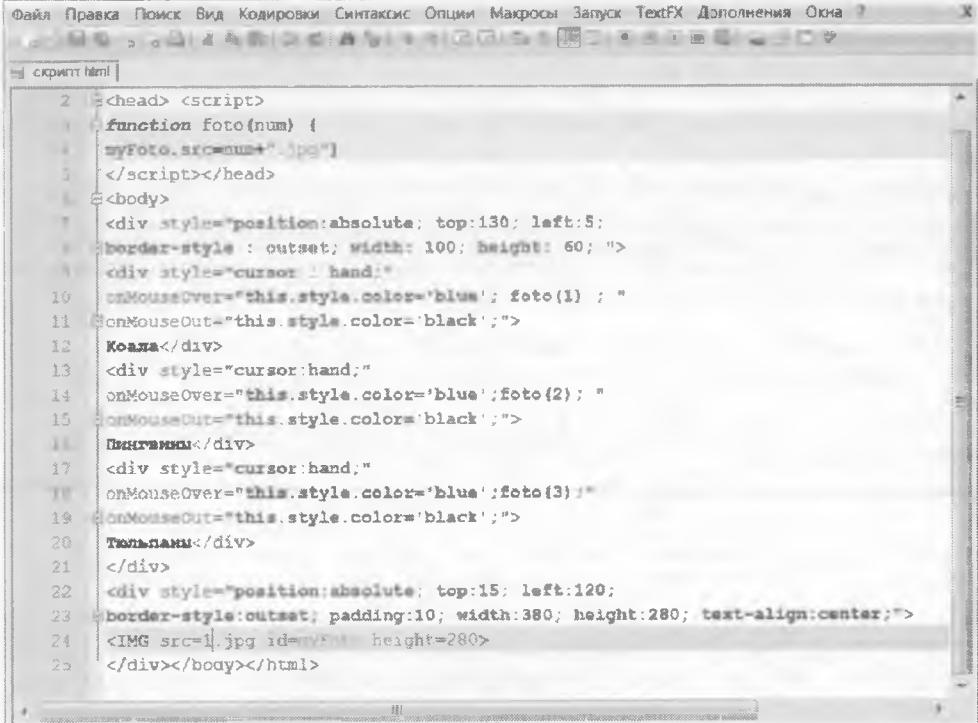
Рис. 5.18. Выполнение тестового задания с выводом результатов

5.4. Более сложные сценарии

Сценарий для вывода изображения при выборе пункта меню

Продолжим рассмотрение функций на примере сценария, позволяющего открывать фотографии. При наведении указателя мыши на надпись с именем фотографии в специально отведенном окне страницы будет отображаться сама фотография.

Для начала необходимо подготовить фотографии для просмотра, переименовав их согласно порядковым номерам. Это важно, так как для реализации задачи мы будем использовать функцию с передачей параметра, а параметром станет именно номер в имени фотографии. Далее нужно определиться с тем, где будут размещены фотографии. В приведенном листинге предполагается, что фотографии размещены в одном каталоге с файлом веб-страницы. Если у вас не так, то в теле функции придется указать путь к каталогу с фотографиями (рис. 5.19).



```

Файл Правка Поиск Вид Кодировка Синтаксис Опции Макросы Запуск TextFX Дополнения Окна ? X
| скрипт.html | 1 <head> <script>
2   function foto(num) {
3     myFoto.src='num'+".jpg"
4   </script></head>
5 <body>
6   <div style="position: absolute; top:130; left:5;
7 border-style : outset; width: 100; height: 60; ">
8     <div style="cursor : hand;">
9       onMouseOver="this.style.color='blue'; foto(1) ; "
10      onMouseOut="this.style.color='black';">
11        Коала</div>
12      <div style="cursor:hand;">
13        onMouseOver="this.style.color='blue';foto(2) ; "
14        onMouseOut="this.style.color='black';">
15          Пингвин</div>
16        <div style="cursor:hand;">
17          onMouseOver="this.style.color='blue';foto(3) ; "
18          onMouseOut="this.style.color='black';">
19            Тюльпаны</div>
20        </div>
21      <div style="position: absolute; top:15; left:120;
22 border-style:outset; padding:10; width:380; height:280; text-align:center;">
23        <IMG src=1.jpg id="myFoto" height=280>
24      </div></body></html>

```

Рис. 5.19. Листинг сценария для просмотра изображений

Рассмотрим листинг более подробно.

- `<html>` – начало веб-страницы.
- `<head> <script>` – начало заголовка страницы и сценария.
- `function foto(num) {` – определение и начало функции `foto`, получающей в качестве параметра значение переменной `num`.
- `myFoto.src=num+ ".jpg"}` – присвоение свойству `src` элемента `myFoto` строковой константы, состоящей из значения переменной `num` (номер в имени фотографии, выполняли при подготовке) и ее расширения `.jpg`. Если ваши фото и веб-страница находятся в разных каталогах, то перед переменной `num` укажите адрес, например `myFoto.src="foto/" + num + ".jpg"`.
- `</script></head>` – завершение сценария и заголовка.
- `<body>` – начало тела веб-страницы.
- `<div style="position: absolute; top:130; left:5; border-style : outset; width: 100; height: 60; ">` – начало раздела, определяющего стиль для вывода надписей: начинать с высоты 130, левый край 5 пунктов, рамка шириной 100 и высотой 60 пунктов.
- `<div style="cursor : hand; " onMouseOver="this.style.color='blue'; foto(1) ; " onMouseOut="this.style.color='black'; ">Коала</div>` – раздел, определяющий стиль и поведение надписи **Коала**. При наведении указателя мыши ей задается голубой цвет, в ином случае надпись остается черной.
- `<div style="cursor:hand; " onMouseOver="this.style.color='blue'; foto(2); " onMouseOut="this.style.color='black'; "> Пингвины</div>` – раздел, определяющий стиль и поведение надписи **Пингвины**. При наведении указателя мыши ей задается голубой цвет, в противном случае надпись остается черной.
- `<div style="cursor:hand; " onMouseOver="this.style.color='blue'; foto(3); " onMouseOut="this.style.color='black'; "> Тюльпаны</div>` – раздел, определяющий стиль и поведение надписи **Тюльпаны**. При наведении указателя мыши ей задается голубой цвет, в ином случае надпись остается черной.
- `</div>` – закрытие основного отдела надписей.
- `<div style="position: absolute; top:15; left:120; border-style:outset; padding:10; width:380; height:280; text-align:center; ">` – открытие раздела размещения фотографий и определение его стиля. При абсолютном позиционировании указывает располагать левый верхний угол рамки в 15 пунктах сверху и 120 пунктах от левого края, задает ширину рамки 380 и ее высоту 280 пунктов, располагает объект по центру.
- `` – указывает разместить фотографии с идентификатором `myFoto`, задавая явную ширину

изображений 280 пунктов. По умолчанию определяется изображение под именем 1.jpg.

- </div></body></html> — завершение раздела, тела документа и веб-страницы.

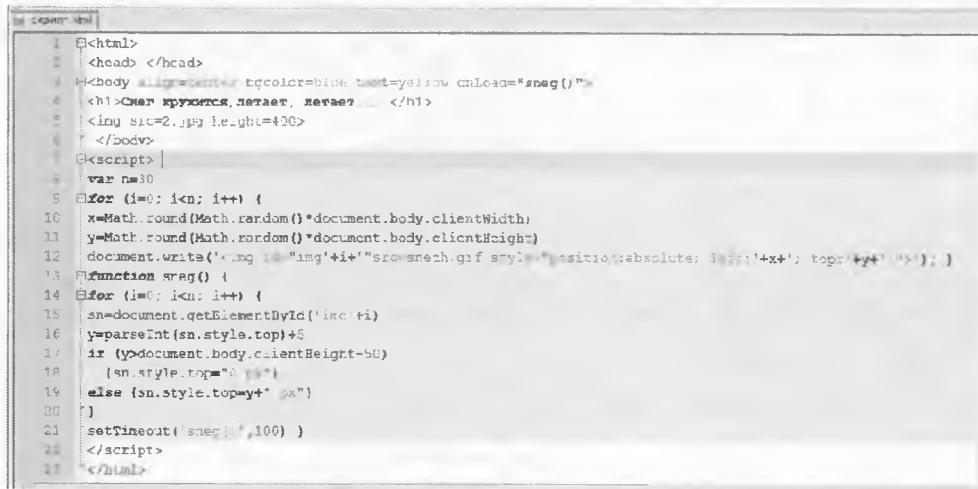
Посмотрим на результат выполнения нашей программы (рис. 5.20).



Рис. 5.20. Выполнение функции с параметром
при реализации сценария

Циклический алгоритм. Сценарий «Падающий снег»

Рассмотрим использование циклических структур на примере популярного в зимнее время года сценария «Падающий снег». Сначала подготовим снежинку. Желательно использовать изображение в формате GIF, поскольку он позволяет выполнить рисунок на прозрачном фоне, что в нашем случае критично. Пример кода приведен на рис. 5.21.



```

1 <html>
2   <head> </head>
3   <body align=center bgcolor=blue text=yellow onLoad="sneg()">
4     <h1>Снег кружится, летает, летает... </h1>
5     <img src=2.jpg height=400>
6   </body>
7 <script>
8   var n=30
9   for (i=0; i<n; i++) {
10     x=Math.round(Math.random()*document.body.clientWidth)
11     y=Math.round(Math.random()*document.body.clientHeight)
12     document.write("<img id="img"+i+" src=snezh.gif style='position: absolute; left:"+x+"px; top:"+y+"px;'" );
13   }
14   function sneg() {
15     for (i=0; i<n; i++) {
16       sn=document.getElementById('img'+i)
17       y=parseInt(sn.style.top)+5
18       if (y>document.body.clientHeight-50)
19         (sn.style.top="0px")
20       else (sn.style.top=y+"px")
21     }
22     setTimeout("sneg()",100)
23   }
24 </script>
25 </html>

```

Рис. 5.21. Сценарий «Падающий снег»

Рассмотрим код более подробно.

- <html> — начало веб-страницы.
- <head> </head> — заголовок.
- <body align=center bgcolor=blue text=yellow onLoad="sneg() "> — начало тела страницы с определением стиля содержимого: выравнивание объектов — по центру, цвет фона — голубой, цвет текста — желтый, при загрузке страницы вызвать процедуру sneg().
- <h1>Снег кружится, летает, летает...</h1> — текст страницы, оформленный наибольшим стилем заголовка.
- — установка базового изображения. В нашем случае взят файл 2.jpg из предыдущего задания с изображением пингвинов.
- </body> — завершение тела документа.
- <script> — начало сценария.
- var n=30 — определение количества повторений для генерации координат снежинок на экране.
- for (i=0; i<n; i++) — цикл. Читается следующим образом: выполнять тело цикла от начального значения счетчика i, равного 1, до тех пор, пока i меньше n, с каждым шагом увеличивая i на 1.
- { — начало тела цикла.
- x=Math.round(Math.random()*document.body.clientWidth) — присвоение координате X текущей снежинки случайного числа, лежащего

в диапазоне от 0 до ширины текущего окна. Иными словами, при изменении ширины окна снежинки будут генерироваться по всему его полю.

- `y=Math.round(Math.random() * document.body.clientHeight)` — присвоение координате Y текущей снежинки случайного числа, лежащего в диапазоне от 0 до высоты текущего окна.
- `document.write('')` — вывод на поверхность страницы изображения `snezh.jpg` с запоминанием персонального идентификатора каждой снежинки по заранее генерированным координатам.
- `}` — конец тела цикла.
- `function sneg() {` — начало функции.
- `for (i=0; i<n; i++)` — цикл: выполнять тело цикла от начального значения счетчика `i`, равного 1, до тех пор, пока `i` меньше `n`, с каждым шагом увеличивая `i` на 1.
- `{` — начало тела цикла.
- `sn=document.getElementById('img'+i)` — присвоить переменной `sn` (серийный номер) элемент с идентификатором `img1`, `img2`, ... `imgn` текущего документа.
- `y=parseInt(sn.style.top)+5` — для координаты Y текущего `id` координата увеличивается на 5 пунктов, и снежинка сдвигается вниз.
- `if (y>document.body.clientHeight-50)` — условие, определяющее, сдвинулась ли снежинка вниз за пределы экрана. Величины 50 пунктов вполне достаточно, чтобы весь небольшой рисунок ушел за нижнюю границу окна.
- `{sn.style.top="0 px"}` — действие при истинности условия определяет новую координату Y , равную 0.
- `else {sn.style.top=y+" px"}` — действие при ложности условия, сдвиг снежинки вниз.
- `}` — завершение тела цикла.
- `setTimeout('sneq()', 100)` — автоматический вызов функции `sneq()` через 100 мс.
- `}` — завершение тела функции.
- `</script>` — завершение тела сценария.
- `</html>` — конец веб-страницы.

Выполнение сценария можно наблюдать, к сожалению, не в динамике, а на рис. 5.22.



Рис. 5.22. Результат выполнения сценария «Падающий снег»

5.5. Функции

Функции, работающие с системными часами. Сценарий «Простые часы»

Продолжим рассматривать способы оживления веб-страницы. Один из распространенных и популярных способов — установка часов, которые якобы находятся на сервере и демонстрируют неусыпную заботу веб-мастера о пользователе. Для организации часов необходимо средствами языка JavaScript получить текущее время компьютера пользователя и вывести его в ту область веб-страницы, которая этого требует (рис. 5.23).

Разберем листинг более подробно.

- <head> </head> — заголовок.
- <script> — начало сценария.
- function clock() { — начало функции clock().

```

1  <html>
2  <head>
3  <script>
4  function clock() {
5      var time = new Date()
6      var hour = time.getHours()
7      var minute = time.getMinutes()
8      var second = time.getSeconds()
9      var temp = "" + hour
10     temp += (minute < 10) ? ":0" : ":" + minute
11     temp += (second < 10) ? ":0" : ":" + second
12     document.clock.digits.value = temp
13     id = setTimeout("clock()",1000)
14  } </script>
15 </head>
16 <body bgcolor=blue onLoad="clock()">
17 <form name="clock">
18 <input type="text" name="digits" size=6 value="">
19 </form> </body>
20 </html>

```

Рис. 5.23. Пример сценария «Простые часы»

- `var time = new Date()` — присвоение переменной `time` значения объекта хранения системного времени `Date()`.
- `var hour = time.getHours()` — присвоение переменной `hour` значения часов, взятых из переменной `time` методом `getHours()`.
- `var minute = time.getMinutes()` — присвоение переменной `minute` значения часов, взятых из переменной `time` методом `getMinutes()`.
- `var second = time.getSeconds()` — присвоение переменной `second` значения часов, взятых из переменной `time` методом `getSeconds()`.
- `var temp = "" + hour` `temp += (minute < 10) ? ":0" : ":" + minute` `temp += (second < 10) ? ":0" : ":" + second` — формирование в переменной `temp` строки, содержащей часы:минуты:секунды.
- `document.clock.digits.value = temp` — выводит на веб-страницу содержимое переменной `temp`.
- `id = setTimeout("clock()",1000)` — осуществляет циклический вызов функции `clock()` через одну секунду.
- `} </script>` — завершение функции и сценария.
- `</head>` — завершение заголовка.
- `<body bgcolor=blue onLoad="clock()">` — определение цвета страницы как голубого и вызов функции `clock()` при загрузке страницы.

- `<form name="clock">` — определение формы под именем `clock`, в которой и будут располагаться часы.
- `<input type="text" name="digits" size=8 value="">` — вывод содержимого часов в текстовое поле `digits` размером 8 пунктов.
- `</form> </body>` — завершение формы и документа.

Результат выполнения сценария проиллюстрирован на рис. 5.24.



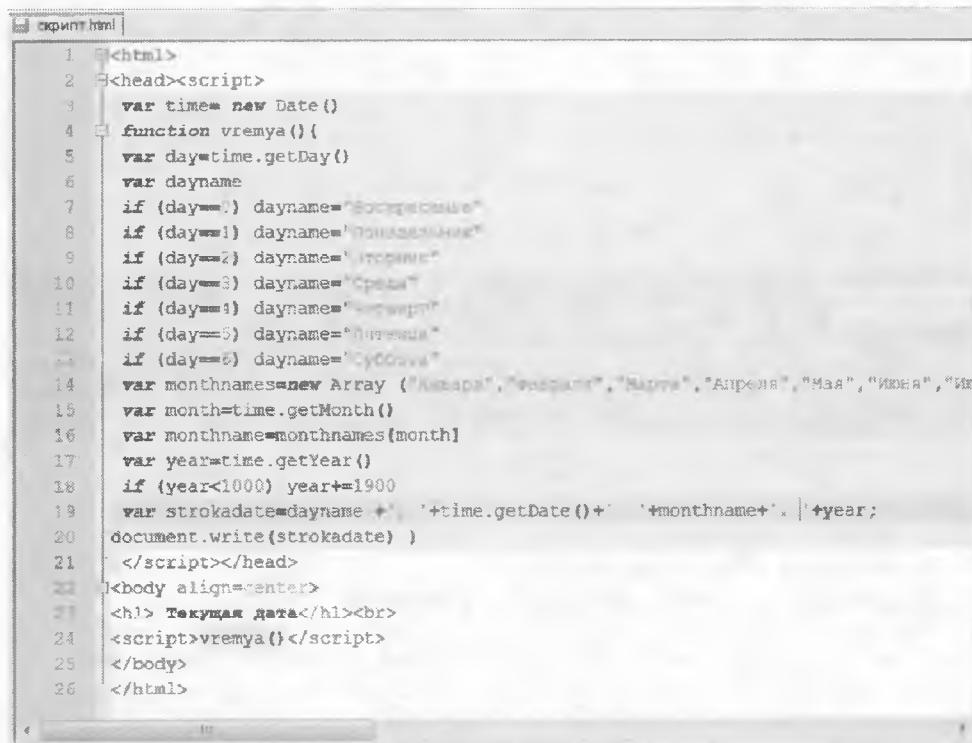
Рис. 5.24. Результат выполнения сценария «Простые часы»

Сценарий «Текущая дата». Знакомство с обработкой массивов

По аналогии с часами эффект присутствия создается и при выводе информации о текущей дате. Для иллюстрации работы с массивами в языке JavaScript рассмотрим код сценария, определяющего и выводящего текущую дату на поле веб-страницы (рис. 5.25).

Рассмотрим листинг более подробно.

- `<html>` — начало веб-страницы.
- `<head><script>` — начало заголовка и сценария.
- `var time= new Date()` — определение переменной `time` и присвоение ей значения текущей даты.
- `function vremya() {` — определение и начало функции.
- `var day=time.getDay()` — определение и присвоение переменной `day` начального числового значения текущего дня по системной дате.
- `var dayname` — определение переменной `dayname`.
- `if (day==0) dayname="Воскресенье"` — присваивание значения переменной `dayname` в зависимости от значения переменной `day`. Аналогично для последующих шести строк.
- `var monthnames=new Array ("Января", "Февраля", "Марта", "Апреля", "Мая", "Июня", "Июля", "Августа", "Сентября", "Октября", "Ноября", "Декабря");` — описание массива элементов `monthnames` и присвоение значений его элементам.



```

1 <!><html>
2 <head><script>
3 var time= new Date()
4 function vremya(){
5     var day=time.getDay()
6     var dayname
7     if (day==0) dayname="Воскресенье"
8     if (day==1) dayname='Понедельник'
9     if (day==2) dayname="Вторник"
10    if (day==3) dayname="Среда"
11    if (day==4) dayname="Четверг"
12    if (day==5) dayname="Пятница"
13    if (day==6) dayname="Суббота"
14    var monthnames=new Array ("Января","Февраля","Марта","Апреля","Мая","Июня","Июля",
15    var month=time.getMonth()
16    var monthname=monthnames[month]
17    var year=time.getFullYear()
18    if (year<1000) year+=1900
19    var strokadate=dayname + ', ' +time.getDate() + ' ' +monthname+ ', ' +year;
20    document.write(strokadate) }
21 </script></head>
22 <body align=center>
23 <h1> Текущая дата</h1><br>
24 <script>vremya()</script>
25 </body>
26 </html>

```

Рис. 5.25. Сценарий, выводящий текущую дату на веб-страницу и иллюстрирующий работу с массивом, содержащим имена месяцев

- `var month=time.getMonth()` — определение переменной `month` и присвоение ей числового значения текущего месяца по системной дате.
- `var monthname=monthnames [month]` — определение переменной `monthname` и присвоение ей названия месяца согласно расположению элемента в массиве.
- `var year=time.getFullYear()` — определение переменной `year` и присвоение ей значения текущего года.
- `if (year<1000) year+=1900` — определение условия текущего года. При истинности условия переменная `year` увеличивается на 1900.
- `var strokadate=dayname + ', ' +time.getDate() + ' ' +monthname+ ', ' +year;` — определение строковой переменной `strokadate` и формирование ее содержимого.
- `document.write(strokadate) }` — вывод на веб-страницу переменной `strokadate`, завершение функции.
- `</script></head>` — завершение сценария и заголовка.

- `<body align=center>` — начало тела документа и выравнивание по центру его содержимого.
- `<h1> Текущая дата</h1>
` — надпись, оформленная максимальным заголовком.
- `<script>vremya ()</script>` — сценарий, вызывающий функцию `vremya ()`.
- `</body>` — завершение тела документа.
- `</html>` — завершение веб-страницы.

Результат выполнения сценария иллюстрируется на рис. 5.26.



Рис. 5.26. Результат выполнения сценария, выводящего текущую дату, с использованием элементов массива, содержащих название месяцев года

5.6. Сценарии из библиотек сторонних разработчиков

Реализация сложного сценария. Сценарий «Слайд-шоу». Самостоятельная реализация сложных сценариев — достаточно трудное и кропотливое мероприятие, если только вы не изучали программирование как основную специальность. В то же время просторы Сети позволяют выбрать практически любой необходимый сценарий на страницах как любительских, так и специализированных сайтов. Вышеизложенной информации вполне достаточно, чтобы прочитать и понять небольшой сценарий и при этом определить, как его можно адаптировать под вашу страницу. Кроме того, данной информации хватит для прочтения большого сценария и его расшифровки и модификации в своих целях. Попробуем разобраться с одним популярным сценарием, который в свободном доступе можно найти в Интернете. Сценарий организует просмотр изображений в форме слайд-шоу, то есть позволяет перелистывать изображения, появляющиеся на странице (рис. 5.27).

Рассмотрим данный сценарий согласно полученным ранее знаниям. Поскольку он достаточно сложен — в нем подразумевается работа со слоями, то для детального понимания и модификации сценария вам понадобится учебник или спецификация языка JavaScript. Рассмотрим код в общих чертах, чтобы понять, как организовать свои изображения для использования в этом сценарии.

```

1 <html><head><script>
2 var imageIndex=0;
3 function goToFirst(){imageIndex=0; slaydshow()}
4 function goToPrevious(){if (imageIndex>0) imageIndex--; slaydshow()}
5 function goToNext(){if (imageIndex < images.length) imageIndex++; slaydshow()}
6 function goToLast(){imageIndex=images.length - 1; slaydshow()}
7 function slaydshow() {
8     imageHolderObject.src=images[imageIndex];
9     firstObject.style.visibility=(imageIndex==0) ? "hidden":"visible";
10    previousObject.style.visibility=firstObject.style.visibility;
11    nextObject.style.visibility=(imageIndex==images.length - 1) ? "hidden":"visible";
12    lastObject.style.visibility=nextObject.style.visibility;
13 }
14 </script></head>
15 <body>
16 <img id="imageHolder" src=""><p>
17 <a id="first" href="#" onClick="goToFirst();"><lt;<lt;</a>&ampnbsp
18 <a id="previous" href="#" onClick="goToPrevious();"><lt;</a>&ampnbsp
19 <a id="next" href="#" onClick="goToNext();"><gt;</a>&ampnbsp
20 <a id="last" href="#" onClick="goToLast();"><gt;<gt;</a></p>
21 </script>
22 var images=new Array()
23 images[0] = "pic1.jpg"; images[1] = "pic2.jpg"; images[2] = "pic3.jpg"; images[3] = "pic4.jpg"; images[4] = "pic5.jpg";
24 var imageHolderObject=document.all["imageHolder"];
25 var firstObject=document.all["first"];
26 var previousObject=document.all["previous"];
27 var nextObject=document.all["next"];
28 var lastObject=document.all["last"];
29 goToFirst();
30 </script></body></html>

```

Рис. 5.27. Сценарий, создающий слайд-шоу

- <html><head><script> — начало страницы и сценария.
- var imageIndex=0; — объявление переменной, в которой будет храниться номер фотографии, отображаемой на данный момент на экране.
- function goToFirst(){imageIndex=0; slaydshow()} — переход к первой фотографии.
- function goToPrevious(){if (imageIndex>0) imageIndex--; slaydshow()} — переход к предыдущей фотографии.
- function goToNext(){if (imageIndex < images.length) imageIndex++; slaydshow()} — переход к последующей фотографии.
- function goToLast(){imageIndex=images.length - 1; slaydshow()} — переход к последней фотографии.
- Далее идет функция, выводящая на экран текущее изображение и скрывающая неиспользуемые на данный момент ссылки.

```

function slaydshow() {
    imageHolderObject.src=images[imageIndex];
    firstObject.style.visibility=(imageIndex==0) ? "hidden":"visible";
    previousObject.style.visibility=firstObject.style.visibility;

```

```

nextObject.style.visibility=(imageIndex==images.length - 1) ?
"hidden":"visible";
lastObject.style.visibility=nextObject.style.visibility;

```

- </script></head> — окончание сценария.
- <body> — начало тела страницы.
- <p> — создание пустого изображения.
- Далее идет блок гиперссылок, осуществляющих переключение между изображениями, где последовательности символов, начинающиеся с &, соответствуют открывающимся и закрывающимся угловым скобкам, которые обозначают стрелки перехода между фотографиями.

```

<a id="first" href="#" onClick="goToFirst();"><&lt;&lt;></a>&nbsp;
<a id="previous" href="#" onClick="goToPrevious();"><&lt;</a>&nbsp;
<a id="next" href="#" onClick="goToNext();"><&gt;</a>&nbsp;
<a id="last" href="#" onClick="goToLast();"><&gt;<&gt;</a></p>
<script>

```

- var images=new Array() — объявление массива, содержащего фотографии.
- images[0] = "pic1.jpg"; images[1] = "pic2.jpg"; images[2] = "pic3.jpg"; images[3] = "pic4.jpg"; images[4] = "pic5.jpg"; images[5] = "pic6.jpg"; — присвоение элементам массива соответствующих имен фотографий. Соответственно, количество элементов массива зависит от количества фотографий. Только не забывайте присваивать значения элементам массива.
- Блок объявления набора экземпляров объектов, которые соответствуют гиперссылкам и так называемому пустому изображению.

```

var imageHolderObject=document.all["imageHolder"];
var firstObject=document.all["first"];
var previousObject=document.all["previous"];
var nextObject=document.all["next"];
var lastObject=document.all["last"];

```

- goToFirst(); — вызов первой фотографии; функция активизируется при загрузке страницы.
- </script></body></html> — окончание сценария и страницы.

Результат выполнения сценария — слайд-шоу. При загрузке страницы будет выведена первая фотография, описанная в массиве, а под ней — стрелки перехода между снимками (рис. 5.28).

Как видно из представленного в данной главе материала, оживить сайт не так трудно, как казалось первоначально. Сценарии, написанные не только на JavaScript, но и на других языках программирования, пре-восходно справляются с большинством поставленных задач. Они могут с успехом использоваться на веб-страницах при размещении сайта на бесплатных веб-серверах, имеющих некоторые ограничения, например на сервисе narod.ru.

Если же язык программирования и написанные на нем сценарии не могут удовлетворить ваши требования, то, скорее всего, необходимо задуматься о платном хостинге и переходе на более сложную систему управления сайтом. Как вариант можно рассмотреть систему управления содержимым Joomla!, но об этом мы поговорим подробнее в следующих главах.

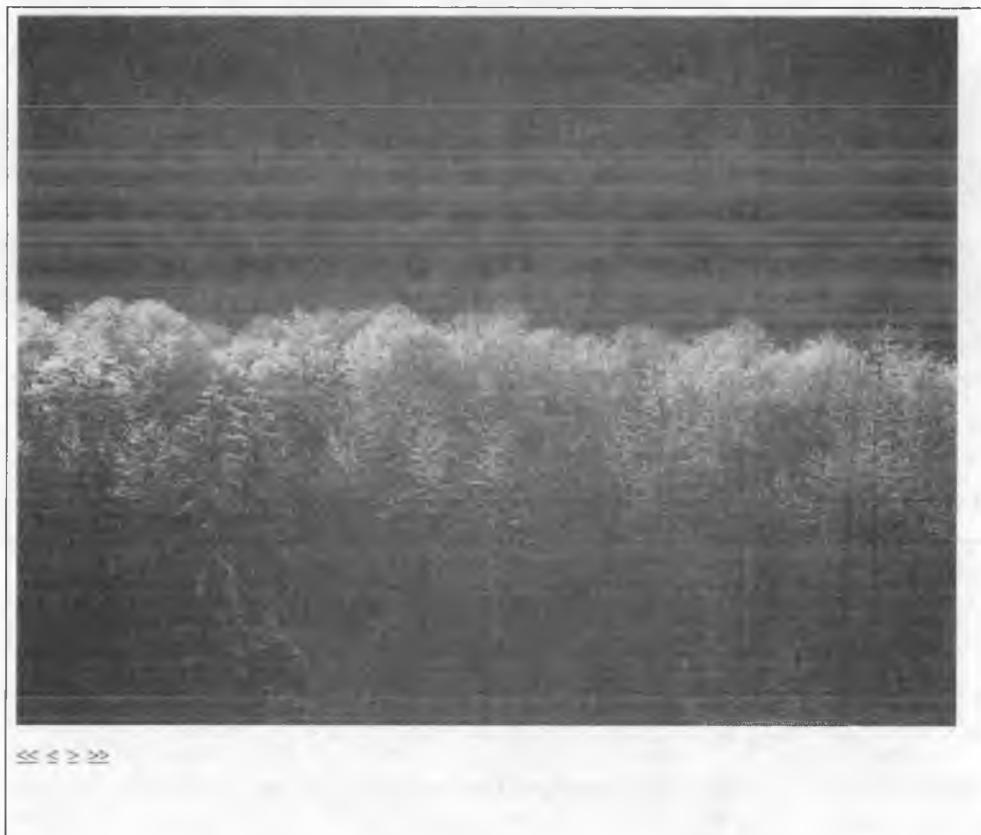


Рис. 5.28. Результат выполнения сценария «Слайд-шоу»

5.7. Обзор библиотеки jQuery

Что такое библиотека

Рассматривая тенденции развития веб-программирования за последние несколько лет, можно сделать следующий достаточно противоречивый вывод: в принципе, почти все программы, которые используются на сайтах, уже написаны. Однако для того, чтобы найти ту единственную необходимую программу, следует потратить столько времени, трафика и сил, что проще ее написать самому. Впрочем, такой парадокс характерен не только для программирования, но и, пожалуй, для любой современной человеческой деятельности, связанной с обработкой информации. Тем не менее тенденция развития программ для веб-приложений указывает на то, что расширение их функциональности идет в направлении, которое было еще несколько лет назад доступно только прикладным программам. Иначе говоря, современное веб-приложение должно быть анимированным, уметь обращаться к серверу без перезагрузки страниц, использовать те элементы программы, которые всегда считались не веб-элементами, например деревья и сплиттеры, и т. д. и т. п.

Основные облачные технологии обработки информации завязаны на использовании браузеров, которые, как вы помните, являются исполнителями HTML-кода как инструкции по отображению содержимого веб-страницы. В свою очередь, ключевые технологии по созданию веб-страниц являются строго статическими, как сам язык HTML или CSS. Это приводит к тому, что всю необходимую подвижную функциональность веб-приложения приходится реализовывать с помощью языков программирования, в частности JavaScript.

Что же такое библиотека? У каждого из нас есть определенный запас знаний, которые мы получали и усваивали различными путями. Если имеющихся знаний не хватает для решения поставленной задачи, то мы, соответственно, обращаемся в то место, где подобные знания хранятся, — в библиотеку. Вне зависимости от того, где она находится — в соседнем доме за углом, куда нас отвели в первом классе, или в Интернете, где обсуждаются интересующие нас вопросы, — библиотека выполняет свою главную функцию — хранит различную информацию и предоставляет ее пользователям по их запросам.

Библиотеки компьютерных программ также не исключение в этом общем правиле. В досетевую эпоху любой программист, нарабатывающий опыт по специальности, рано или поздно обзаводился определенным набором программ, написанных самостоятельно либо полученных от знакомых или через третьи руки. Подобные личные библиотеки программ очень облегчали повседневный труд по написанию кода. Если библиотека была достаточно обширной, а задача тривиальной, то порой написание

программы сводилось к объединению библиотек и созданию интерфейса. В настоящее время, когда понятия «компьютер» и «сеть» начинают сливаться в единое целое, обмен библиотеками не столь актуален. Ведь гораздо проще вступить в сообщество, развивающее определенное направление в программировании, и помещать собственноручно написанные программы в общие файлы библиотек. Затем их можно самостоятельно использовать в полном объеме и предоставлять для работы всем желающим сделать мир красивее. Таким образом, можно сказать, что библиотеки в мире программирования — это файлы с набором подпрограмм, которые где-то хранятся и доступны для использования.

В настоящее время существует множество библиотек, позволяющих реализовывать такие вещи, как красочные графические эффекты, анимация, перетягивание или сортировка предметов, общение с сервером без перезагрузки страницы и многое другое. К их числу относится библиотека `jQuery`, об использовании которой мы и поговорим.

Обратимся к самой библиотеке `jQuery`, которую можно совершенно бесплатно скачать с сайта проекта по адресу http://docs.jquery.com/Downloading_jQuery. На момент написания книги доступна версия в виде файла с именем `jquery-1.6.js`. На момент прочтения книги вами последняя версия, скорее всего, будет отличаться номером в имени файла и, как следствие, более широким списком функций. Как видите, библиотека представлена в виде обычного текстового файла, имеющего расширение `.js`. Расширение указывает на то, что файл хранит в себе код, написанный на языке программирования `JavaScript`, и может быть подключен к любому файлу с `HTML`-кодом путем указания его имени в теле тега `<script>`. Если просмотреть содержимое файла в текстовом редакторе (рис. 5.29), то можно увидеть программный код, написанный на языке `JavaScript`.

```

87
88 // [[Class]] -> type pairs
89 class2type = {};
90
91 jQuery.fn = jQuery.prototype = {
92     constructor: jQuery,
93     init: function( selector, context, rootjQuery ) {
94         var match, elem, ret, doc;
95
96         // Handle $("") , $(null) , or $(undefined)
97         if ( !selector ) {
98             return this;
99         }
100
101         // Handle $(<DOMElement>)
102         if ( selector.nodeType ) {
103             this.context = this[0] = selector;
104             this.length = 1;
105             return this;
106     }

```

Рис. 5.29. Фрагмент программного кода в файле библиотеки `jquery-1.6.js`

Теперь вспомним про объектную модель документа – DOM, о которой было рассказано в главе 3. Так вот, использование библиотек, подобных jQuery, как раз и позволяет упростить создание динамического HTML-документа. Это те самые веб-страницы, которые хоть и являются по своей основе статическими, но могут изменять свойства своих объектов как обычные программные приложения, просто реагируя на события. Вот здесь-то и необходимо иметь углубленные знания в использовании не только языка программирования JavaScript, но и языка каскадных таблиц стилей CSS. Действие jQuery основано на управлении изменением свойств объектов HTML-страницы путем изменения значения правила CSS для конкретного элемента веб-страницы. Краеугольным камнем здесь, как и в каскадных таблицах стилей, является понятие селектора.

Подключение библиотеки jQuery. Пробуем получить из нее то, что нужно

Сама библиотека представляет собой файл с расширением JS. Напомню, что мы работаем с файлом `jquery-1.6.js`. Для использования полученный файл с библиотекой jQuery следует разместить в корневом каталоге с создаваемым сайтом или в подкаталоге сайта, специально отведенном под размещение сценариев. Это необходимо сделать для того, чтобы можно было подключить библиотеку с любой страницы, использующей сценарии. Кроме того, для подключения сценариев к веб-странице есть три различных способа: указание названия непосредственно в HTML-теге веб-страницы, прописывание тела сценария в головной части веб-страницы либо подключение сценария, находящегося в отдельном файле, путем указания его расположения. Соответственно, для подключения библиотеки jQuery мы должны инициализировать ее в коде веб-страницы с помощью следующего тега:

```
<script type="text/javascript" src="jquery-1.6.js"></script>
```

Здесь значением параметра `src` будет имя файла библиотеки, если используется ее версия, отличная от описанной в книге.

Теперь я расскажу, как получать из библиотеки те функции, которые необходимы для сценария.

Рассмотрим сценарий, при выполнении которого объект `panel` опускается по окну браузера при однократном нажатии (`click`) кнопки (`button`) (рис. 5.30).

```

6
7 <script type="text/javascript" >
8   $(document). ready (function () {
9     $(" .button" ) .click( function () {
10       $("panel" ) .slideDown("slow");
11     });
12   });
13 </script>

```

Рис. 5.30. Пример использования библиотечных функций jQuery в теле сценария

- `<script type="text/javascript">` — объявление начала тела самого сценария.
- `$(document).ready(function (){` — определение готовности объекта DOM и задание события `ready` для функции.
- `$(".button") .click(function (){` — привязка выполнения функции по событию `click`, произведенному над элементом класса `button`.
- `$("#panel") .slideDown("slow");` — над элементом, имеющим ID `panel`, выполнять действие, связанное с его перемещением вниз по окну браузера.
- `});` — завершение вложенной функции.
- `});` — завершение функции.
- `</script>` — завершение тела сценария.

Как видно из описания листинга, подключение и вызов на выполнение функций из библиотеки не составляют большого труда. Иными словами, проблема не в том, как взять книгу с библиотечной полки, а в том, какую книгу необходимо брать для выполнения нужных действий. Это, в свою очередь, материал для дополнительного изучения как самого языка программирования JavaScript, так и CSS вместе с HTML.

Почему? Чтобы получить элемент из библиотеки jQuery, необходимо иметь представление о том, что такое селектор, и понимать, как он работает. Вот несколько примеров работы селектора jQuery:

- `$("#header")` — получить элемент с `id="header"`;
- `$("h3")` — получить все элементы `<h3>`;
- `$("div#content .photo")` — получить все элементы с классом `= "photo"`, которые находятся в элементе `div` с `id="content"`;
- `$("ul li")` — получить все элементы `` из списка ``;
- `$("ul li:first")` — получить только первый элемент `` из списка ``.

Пример опускающейся и поднимающейся слайд-панели

Попробуем реализовать действие на основе предыдущего листинга. При выборе ссылки будет раскрываться вниз и закрываться вверх слайд-панель (рис. 5.31 и 5.32). Данный пример в динамике можно посмотреть по адресу <http://anton.shevchuk.name/wp-demo/jquery-tutorials/simple-slide-panel.html>.



Рис. 5.31. Закрытая слайд-панель



Рис. 5.32. Открытая слайд-панель

Полный листинг веб-страницы выглядит следующим образом:

```

<head>
<title>Simple Slide Panel</title>
<script type='text/javascript' src='http://ajax.googleapis.com/ajax/libs/
jquery/1.4/jquery.min.js'></script>
<script type="text/javascript">
$(document).ready(function(){
    $(".btn-slide").click(function() {
        $("#panel").slideToggle("slow");
        $(this).toggleClass("active"); return false;
    });
});
</script>
<style type="text/css">
body {
    margin: 0 auto;
    padding: 0;
    width: 570px;
    font: 75%/120% Arial, Helvetica, sans-serif;
}
a:focus {
    outline: none;
}
#panel {
    background: #754c24;
    height: 200px;
<head>
<title>Simple Slide Panel</title>

```

- Инициализация библиотеки jQuery, находящейся по следующему адресу <http://ajax.googleapis.com/ajax/libs/jquery/1.4/jquery.min.js>:

```
<script type='text/javascript' src='http://ajax.googleapis.com/ajax/libs/
jquery/1.4/jquery.min.js'></script>
```

- Начало тела сценария. Подготовка и формирование модели DOM:

```
<script type="text/javascript">
```

```
$(document).ready(function() {
```

- Изменение значения класса для элемента, имеющего ID "panel", со значения ".btn-slide" на "active" и обратно. Таким образом, сам элемент panel будет опускаться и подниматься в зависимости от событий, а смена значения класса изменит размеры и позиции фоновых изображений прямоугольной панели, которая будет и сужаться, и раздвигаться:

```
$(".btn-slide").click(function() {  
    $("#panel").slideToggle("slow");  
    $(this).toggleClass("active"); return false;  
});  
});  
</script>
```

- Открытие таблицы стилей для данного файла:

```
<style type="text/css">
```

- Определение правил стиля для тела веб-страницы:

```
body {  
    margin: 0 auto;  
    padding: 0;  
    width: 570px;  
    font: 75%/120% Arial, Helvetica, sans-serif;  
}
```

- Отключение фокусировки на объекте:

```
a:focus {  
    outline: none;  
}
```

- Объявление правил стиля элемента panel:

```
#panel {
```

```

background: #754c24;
height: 200px;
display: none;
}

```

- Объявление правил стиля для селектора `slide`. URL в скобках указывает на размещение рисунка, содержащего изображение стрелки:

```

.slide {
margin: 0;
padding: 0;
border-top: solid 4px #422410;
background: url(images/btn-slide.gif) no-repeat center top;
}

```

- Объявление правил стиля для селектора `btn-slide`. URL в скобках указывает на размещение рисунка, содержащего изображение стрелки:

```

.btn-slide {
background: url(images/white-arrow.gif) no-repeat right -50px;
text-align: center;
width: 144px;
height: 31px;
padding: 10px 10px 0 0;
margin: 0 auto;
display: block;
font: bold 120%/100% Arial, Helvetica, sans-serif;
color: #fff;
text-decoration: none;
}

```

- Объявление правил стиля для селектора `active`:

```

.active {
background-position: right 12px;
}
</style>
</head>

```

- Начало тела документа:

```
<body>  
  
    <!-- you can put content here -->  
  
</div>
```

- Вызов ссылки для селектора `btn-slide` с надписью на ней `Slide Panel`:

```
<p class="slide"><a href="#" class="btn-slide">Slide Panel</a></p>
```

- Окончание тела документа и самой веб-страницы:

```
</body>  
  
</html>
```

Из примера видно, что вносимые сценарием изменения в виде смены класса ссылки просто переключают отображение одного стиля объекта на другой, заранее описанный в каскадной таблице стилей CSS. Эта таблица расположена в головной части веб-страницы. В нашем случае происходит простое увеличение/уменьшение высоты объекта `panel`.

Калькулятор «Сколько будет стоить свадьба»

Продолжим рассматривать возможность применения библиотеки jQuery на примере создания приложения «Сколько будет стоить свадьба» в виде калькулятора, написанного на JavaScript. Он используется для расчета услуг по проведению свадьбы и опубликован по адресу <http://ruseller.com/lessons.php?rub=32&id=1084>. Полный код страницы калькулятора «Сколько будет стоить свадьба» будет выглядеть следующим образом:

```
<html>  
  
<head>  
  
<script type="text/javascript" src="jquery-1.6.js"></script>  
  
,<script type="text/javascript">  
  
$(document).ready(function() {  
  
    //Калькулятор  
  
    $("#tableSelect").change(function() {  
  
        var totalSum = 0;  
  
        $("input[id^='pos'], select[id^='pos']").each(function() {  
  
            var idElement = "#" + $(this).attr("id"), //Получаем id элемента
```

```

        priceField = parseInt($("#idElement"+"_price").text()), //  

Стоимость позиции

        countField = parseInt($("#this").val()); //Значение элемента

        $("#idElement"+"_count").text(priceField * countField); //Выво-  

дим сумму по позиции

        totalSum += priceField * countField;

    });

    $("#count_price span").text(totalSum);

});


```

</script>

<body>

<table id="tableSelect" style="width: 100%; border="0" cellspacing="0" cellpadding="0">

<tbody><tr>

<td>Ведущий</td>

<td class="vertical">Часы</td>

<td>

<select id="pos_1" name="veduschiy">

<option value="0">-</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

<option value="8">8</option>

<option value="9">9</option>

<option value="10">10</option>

</select>

</td>

```
<td id="pos_1_price" class="vertical">3000</td>
<td id="pos_1_count">0</td>
</tr>
<tr>
<td>Банкетный зал</td>
<td class="vertical">Человек</td>
<td>
<input id="pos_2" name="people" type="text" />
</td>
<td id="pos_2_price" class="vertical">2000</td>
<td id="pos_2_count">0</td>
</tr>
<tr>
<td>Теплоход</td>
<td class="vertical">Человек</td>
<td>
<input id="pos_3" name="people_t" type="text" />
</td>
<td id="pos_3_price" class="vertical">3000</td>
<td id="pos_3_count">0</td>
</tr>
<tr>
<td>Видеосъемка</td>
<td class="vertical">Часов</td>
<td>
<select id="pos_4" name="video">
<option value="0">-</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
```

```
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
    </select>
</td>
<td id="pos_4_price" class="vertical">2500</td>
<td id="pos_4_count">0</td>
</tr>
<tr>
    <td>Укражение шарами</td>
    <td class="vertical">-</td>
    <td>
        <select id="pos_6" name="bubbles">
            <option value="0">Нет</option>
            <option value="1">Да</option>
        </select>
    </td>
    <td id="pos_6_price" class="vertical">5000</td>
    <td id="pos_6_count">0</td>
</tr>
</tbody></table>
</body>
</html>
```

Изучая представленный код, мы видим много ранее рассмотренных элементов, в частности сценарии, таблицы и формы. Как видно по структуре страницы, сначала инициализируется сама библиотека jQuery, затем с помощью символа \$ в сценарии вызываются функции из подключенной библиотеки и потом в теле веб-страницы описывается сама форма.

Если подробнее рассматривать элементы формы, то видно, что каждый элемент вне зависимости от своего типа имеет идентификатор, который представлен в виде `pos_номер` и предназначен для простоты написания кода с использованием библиотеки `jQuery`. Есть следующие идентификаторы: `pos_номер_price` – стоимость позиции за единицу и `pos_номер_count` – итоговая сумма по позиции.

Если с формой все достаточно понятно, то постараемся подробнее рассмотреть код самого сценария, использующего библиотеку `jQuery`.

```

<script type="text/javascript">

$(document).ready(function() {

    //Калькулятор

    $("#tableSelect").change(function() {

        var totalSum = 0;

        $("input[id^='pos'], select[id^='pos']").each(function() {

            var idElement = "#" + $(this).attr("id"), //Получаем id элемента

                priceField = parseInt($(idElement + "_price").text()), //Стоимость позиции

                countField = parseInt($(this).val()); //значение элемента

            $(idElement + "_count").text(priceField * countField); //Выводим сумму по позиции

            totalSum += priceField * countField;

        });

        $("#count_price span").text(totalSum);

    });

});

```

- Первая строка после комментария показывает, что все обрабатываемые изменения относятся к таблице `tableSelect`.
- Вторая строка присваивает итоговой сумме нулевое значение: `var totalSum = 0`.
- Стока, содержащая запись `$("input[id^='pos'], select[id^='pos']").each`, заставляет выполнять соответствующие действия для каждого `select` и `input`, имеющих `id`, которые начинаются на `pos`. Внутри функции `each` мы указываем те действия, которые желаем выполнять. Переменные задаются следующим образом.

- `idElement = $("#" + $(this).attr("id"))` — создание переменной `idElement`, которая определяет смежные поля с родственными идентификаторами.
- `priceField = parseInt($("#" + idElement + "_price").text())` — будет хранить стоимость по позиции из ячейки с родственными идентификаторами. Как помните, `parseInt` переводит текст, находящийся в данной переменной, в число.
- `countField = parseInt($("#" + this).val())` — хранит получаемое значение, введенное пользователем, и также переводит его в число.
- Страна `$(idElement + "_count").text(priceField * countField)`; — выводит сумму по позиции в нужную ячейку таблицы.
- Страна `totalSum += priceField * countField`; подсчитывает общую сумму, которую строка `$("#count_price span").text(totalSum)`; записывает в отведенное ей место.

На рис. 5.33 представлен результат выполнения описанного сценария.

Ведущий	Часы	<input type="text" value="5"/>	3000	15000
Банкетный зал	Человек	<input type="text" value="200"/>	2000	400000
Теплоход	Человек	<input type="text" value="200"/>	3000	600000
Видеосъемка	Часы	<input type="text" value="3"/>	2500	7500
Украшение шарами	-	<input type="text" value="Да"/>	5000	5000

Рис. 5.33. Иллюстрация работы калькулятора «Сколько будет стоить свадьба»

На примере использования сценария в виде калькулятора «Сколько будет стоить свадьба» видно, каким образом можно применять рассматриваемую библиотеку jQuery. Она позволяет выполнять на веб-странице расчеты с использованием как числовых констант, так и переменных, получаемых сценарием для обработки через соответствующую форму, находящуюся на веб-странице. Кроме того, можно применять раскрывающиеся списки, значения выбранных элементов которых также могут передаваться сценарию для обработки. Если задаться целью, то можно объединить сценарий расчета с формой передачи заявки по электронной почте, сформировав готовый счет-заказ на услуги или товары, производимые вашей компанией, но это требует более серьезных знаний в данной области.

Создание фотогалереи на базе библиотеки jQuery

На приведенных ранее примерах рассмотрены основные принципы подключения библиотеки jQuery и основные приемы работы с ее элементами. Чтобы использовать все преимущества библиотечных функций языков

программирования, необходимо обладать достаточно фундаментальными знаниями как в области JavaScript, так и в области CSS. Но несмотря на отсутствие фундаментальных знаний в нужных областях, при наличии базовых знаний, почерпнутых из ранее изложенного материала, вы можете сравнить два сценария, выполняющих одну и ту же функцию, но исполненных как с применением подключенной библиотеки jQuery, так и без нее, в классическом ключе кода JavaScript.

Рассмотрим работу веб-страницы, на которой демонстрируются изображения, — фотогалереи. В классическом ключе кода языка JavaScript ее создание мы рассмотрели в разделе 5.6 (см. рис. 5.27 и 5.28). Для построения веб-страницы с подключением jQuery вновь обратимся к популярному сайту, пропагандирующему использование этой библиотеки. На веб-странице по адресу <http://anton.shevchuk.name/wp-demo/jquery-tutorials/img-replacement.html> во всем великолепии можно рассмотреть данный пример, иллюстрирующий возможность создания фотогалерей (рис. 5.34).

Illustrations (Image 5)



Рис. 5.34. Пример реализации фотогалереи

Как видно из рис. 5.34, для полномасштабного отображения выбранного изображения необходимо щелкнуть на его эскизе, расположеннном в нижней части страницы. Обратимся к коду страницы и рассмотрим его сразу с пояснениями.

- Начало веб-страницы и головной части документа:

```
<html>
  <head>
```

- Задание названия страницы в теге `<title>`:

```
<title>Image Replacement</title>
```

- Подключение внешней библиотеки `jquery.min.js`, находящейся на удаленном сервере:

```
<script type='text/javascript' src='http://ajax.googleapis.com/ajax/libs/
  jquery/1.4/jquery.min.js'></script>
```

- Начало сценария, реализующего фотогалерею:

```
<script type="text/javascript">
  $(document).ready(function() {
```

- Добавление дополнительного тега `` в заголовке второго уровня `<h2>`:

```
    $("h2").append('<em></em>')
```

- Определение для селектора `thumbs` события `click` для вызова функции:

```
    $(".thumbs a").click(function() {
```

- Определение сохранения атрибута `"href"` в переменной с именем `largePath`:

```
      var largePath = $(this).attr("href");
```

- Определение сохранения атрибута `"title"` в переменной с именем `largeAlt`:

```
      var largeAlt = $(this).attr("title");
```

- Стока, использующая функции библиотеки jQuery, где выполняется замена в элементе `` значений атрибутов `src` и `alt` соответствующими значениями переменных `largePath` и `largeAlt`:

```
      $("#largeImg").attr({ src: largePath, alt: largeAlt });
```

- Присвоение значения элементу `"h2 em"` из переменной `largeAlt`:

```
      $("h2 em").html(" (" + largeAlt + ")"); return false;
    });
  
```

- Закрытие функций, окончание сценария:

```
});  
  
</script>
```

- Начало описания таблицы стилей веб-страницы:

```
<style type="text/css">
```

- Определение правила для тела страницы:

```
body {  
  
    margin: 20px auto;  
  
    padding: 0;  
  
    width: 580px;  
  
    font: 75%/120% Arial, Helvetica, sans-serif;  
  
}
```

- Определение правила для заголовка второго уровня:

```
h2 {  
  
    font: bold 190%/100% Arial, Helvetica, sans-serif;  
  
    margin: 0 0 .2em;  
  
}
```

- Определение правила для заголовка второго уровня с включенным тегом ``:

```
h2 em {  
  
    font: normal 80%/100% Arial, Helvetica, sans-serif;  
  
    color: #999999;  
  
}
```

- Определение правила для селектора с `id="largeImg"`:

```
#largeImg {  
  
    border: solid 1px #ccc;  
  
    width: 550px;  
  
    height: 400px;  
  
    padding: 5px;  
  
}
```

- Определение правила для селектора `thumbs img`:

```
.thumbs img {  
    border: solid 1px #ccc;  
    width: 100px;  
    height: 100px;  
    padding: 4px;  
}
```

- Определение правила для селектора `thumbs img` при значении атрибута `= hover`:

```
.thumbs img:hover {  
    border-color: #FF9900;  
}
```

- Завершение таблицы стилей, окончание головной части веб-страницы:

```
</style>  
</head>
```

- Начало тела документа:

```
<body>
```

- Определение заголовка фотогалереи:

```
<h2>Illustrations</h2>
```

- Определение `id` для первого изображения фотогалереи, указание на его местоположение, присвоение значения атрибуту `alt`:

```
<p></p>
```

- Определение значения селектора `class`:

```
<p class="thumbs">
```

- Определение значений атрибутов `href`, `title` и `src` для каждого изображения, входящего в фотогалерею:

```
<a href="images/img2-lg.jpg" title="Image 2"></a>
```

```
<a href="images/img3-lg.jpg" title="Image 3"></a>
```

```
<a href="images/img4-lg.jpg" title="Image 4"></a>
```

```
<a href="images/img5-lg.jpg" title="Image 5"></a>  
  
<a href="images/img6-lg.jpg" title="Image 6"></a>  
  
</p>
```

- Завершение тела документа, окончание веб-страницы:

```
</body>  
  
</html>
```

Как видим, этот код очень короткий по сравнению с кодом сценария, не использующим подключенную библиотеку jQuery и показанным на рис. 5.27. И в этом случае, как всегда для динамического HTML, все описанные действия сводятся к изменению значений свойств объектов с помощью каскадных таблиц стилей CSS.

Использование неограниченного количества фотографий возможно как в первом классическом случае, так и во втором, только что рассмотренном. Поэтому справедлив вопрос: «Если визуально нет разницы, то в чем смысл?» Все дело в том, что при использовании обычного сценария фотогалереи все изображения будут загружаться только вместе с перезагрузкой всей веб-страницы, на что требуется время и трафик. При этом сначала будет заново загружен весь контент страницы, а затем только изображение, и это будет происходить постоянно при желании пользователя посмотреть новую фотографию.

При использовании динамического HTML в рассмотренном нами примере загрузка фотографии в выделенную для нее область экрана будет выполняться индивидуально, то есть без повторной перезагрузки всей веб-страницы. Согласитесь, что это немаловажно, ведь в результате мы не только бережем нервы посетителей сайта, но и экономим машинное время веб-сервера, отводимое на обслуживание сессии нашего гостя, тем самым многократно повышая производительность обслуживаемого сайта.

Создание раскрывающегося меню средствами библиотеки jQuery

В заключение данной главы, посвященной созданию сценариев средствами JavaScript, хотелось бы рассмотреть еще один показательный элемент программирования в динамическом HTML. Познакомимся с реализацией выпадающих (drop-down) пунктов подменю, которые раскрываются при наведении указателя мыши на один из главных пунктов меню. Этот пример подразумевался в разделе 3.7 как первоначальный пример при знакомстве с объектной моделью документа DOM и динамическим HTML. Если помните, раздел посвящен технологиям каскадных таблиц

стилей CSS и возможностям DHTML. Пример в статическом виде можно посмотреть на рис. 5.35, а его описание, как и замечательную коллекцию сценариев с использованием подключаемых возможностей библиотеки jQuery, можно найти по адресу <http://anton.shevchuk.name/javascript/jquery-for-beginners-2>.

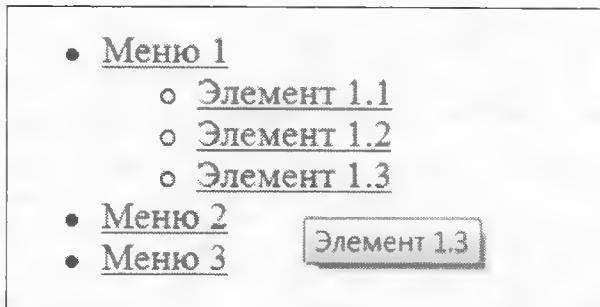


Рис. 5.35. Демонстрация возможности выпадающего меню

Как понятно из представленного рисунка, при наведении на пункт **Меню 1** указателя мыши данный пункт раскрывается вниз, демонстрируя подпункты **Элемент 1.1**, **Элемент 1.2** и **Элемент 1.3**. Количество как пунктов, так и подпунктов меню не ограничено, а их описание и привязка к соответствующим веб-страницам осуществляются непосредственно в HTML-коде документа.

Рассмотрим полный код веб-страницы, реализующий такое меню.

```

<html>
<head>
<title>Image Replacement</title>
<script type='text/javascript' src="jquery-1.6.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    $('.topmenu ul li').hover(
        function() {
            $(this).addClass("active");
            $(this).find('ul').stop(true, true); // останавливаем всю
текущую анимацию
            $(this).find('ul').slideDown();
        },
        function() {

```

```
        $(this).removeClass("active");
        $(this).find('ul').slideUp('fast');
    }
);
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div class="topmenu">
```

```
    <ul>
```

```
        <li><a href="#" title="Меню 1">Меню 1</a>
```

```
            <ul>
```

```
                <li><a href="#" title="Элемент 1.1">Элемент 1.1</a></li>
```

```
                <li><a href="#" title="Элемент 1.2">Элемент 1.2</a></li>
```

```
                <li><a href="#" title="Элемент 1.3">Элемент 1.3</a></li>
```

```
            </ul>
```

```
        </li>
```

```
        <li><a href="#" title="Меню 2">Меню 2</a>
```

```
            <ul>
```

```
                <li><a href="#" title="Элемент 2.1">Элемент 2.1</a></li>
```

```
                <li><a href="#" title="Элемент 2.2">Элемент 2.2</a></li>
```

```
                <li><a href="#" title="Элемент 2.3">Элемент 2.3</a></li>
```

```
            </ul>
```

```
        </li>
```

```
        <li><a href="#" title="Меню 3">Меню 3</a>
```

```
            <ul>
```

```
                <li><a href="#" title="Элемент 3.1">Элемент 3.1</a></li>
```

```
                <li><a href="#" title="Элемент 3.2">Элемент 3.2</a></li>
```

```
                <li><a href="#" title="Элемент 3.3">Элемент 3.3</a></li>
```

```
            </ul>
```

```
        </li>
```

```
</ul>
```

```
</div>
```

```
</body>
```

```
</html>
```

Теперь постараемся в нем разобраться.

С кодом тела страницы, что лежит в пределах тега `<body>`, все достаточно понятно. Класс `topmenu` организуется в теге `<div>`, в котором в виде гиперссылок записываются пункты меню и их подпункты. Подпункты меню спрятаны и будут появляться/раскрываться только при наведении на них указателя мыши.

Далее сценарий выполняет выборку согласно написанному коду, который мы рассмотрим более детально. Поскольку весь код основан на использовании объектной модели документа DOM, наша задача на текущий момент времени — постараться разобраться в том, какая часть сценария за что отвечает.

- Подключение к библиотеке сценариев, расположенной на локальном компьютере в каталоге размещения веб-страницы:

```
<script type='text/javascript' src="jquery-1.6.js"></script>
```

- Начало тела сценария. Определение готовности объектной модели документа (DOM) к работе, начало работы функций сценария:

```
<script type="text/javascript">
$(document).ready(function() {
```

- В классе `topmenu` добавляется обработчик событий `hover`, применяемый для обработки элементов списка `li`:

```
$('.topmenu ul li').hover(
  function() {
```

- Путем добавления класса `active` изменяется фон выбранного элемента при наведении на него указателя мыши:

```
$(this).addClass("active");
```

- Останавливается текущая анимация перемещения пунктов меню (открытие/закрытие), то есть определяется движущийся элемент `ul`, который останавливается.

```
$(this).find('ul').stop(true, true); // останавливаем всю
текущую анимацию
```

- Определяется элемент `ul`, и для него вызывается анимация с помощью библиотечной функции `slideDown`:

```
$(this).find('ul').slideDown();
},
```

- Строки функции, описывающие отображение выпадающих пунктов подменю. Действие также происходит через добавление класса `active` и вызов библиотечной функции `slideUp`, поднимающей подпункт меню:

```
function() {
    $(this).removeClass("active");
    $(this).find('ul').slideUp('fast');
}
);
});
```

- Завершение работы сценария:

```
</script>
```

Несмотря на всю кажущуюся простоту библиотеки jQuery, использование ее элементов, вызываемых функцией `$()`, связано с глубоким знанием и пониманием всех процессов, происходящих при программировании на языке JavaScript. Кроме того, веб-программист в полном объеме должен представлять себе объектную модель документа — DOM. Ведь только используя ее иерархию, можно получить доступ к конкретному объекту веб-страницы и осуществлять последующие манипуляции со свойствами с помощью каскадных таблиц стилей CSS.

И все же функционал возможностей jQuery огромен, а отсутствие сложностей при подключении библиотеки и большое количество готовых решений дают даже неопытному веб-программисту широкие возможности в работе.

Глава 6

ГРАФИКА САЙТА

6.1. Основные понятия

В наше время, насыщенное множеством различных информационных потоков, достаточно трудно привлечь к своему продукту внимание, даже если он содержит жизненно важную информацию. Встречают по одежке — эту особенность человеческой психики давно используют не только профессиональные маркетологи, но и обычные рыночные заявки. Привлечь и удержать первое внимание, заставить посетителя вчитаться, всмотреться и, если совсем повезет, вдуматься в содержание. Как результат — можно получить постоянного посетителя сайта и сформировать из него агента влияния. Еще в девяностые годы прошлого столетия исследователи сделали ошеломляющий вывод: более 70 % напечатанных журнальных статей не читаются, а сразу отправляются в макулатуру... Я думаю, что в настоящее время данный процент можно с уверенностью увеличить.

Важно привлечь первое внимание посетителя, то поверхностное внимание, когда еще не включился разум на получение информации, когда посетитель, не осмысливая, только оперирует образами: нравится — не нравится. Для этих целей и используют на страницах сайта графические объекты. Это могут быть рисунки, фотографии, фоны, анимации и Flash-ролики, которые редко несут большой и важный объем информации, но подсознательно определяют концентрацию внимания на отдельных объектах, формируют настроение при просмотре, вызывают симпатию или отторжение. Таким образом, нельзя недооценивать возможности графики на веб-страницах, но и переборщить с нею тоже нельзя, если только ваш сайт не посвящен иллюстрированию комиксов.

Лирика о типах и форматах. Основные понятия графики для веб-страниц. Графические объекты, используемые на страницах сайтов, и технологии их подготовки принято относить к разделу компьютерных технологий. Этот раздел называют веб-графикой. Соответственно, как и любая другая технология, ориентированная на использование в сетях, веб-графика также имеет свои особенности, требования и правила, несоблюдение которых может не только оттолкнуть посетителей от вашего сайта, но и вообще парализовать работу сервера. Например, вы на своей странице

использовали великолепные рисунки, фоны и фотографии, но не уменьшили их размер. Посетители валом хлынули к вам в гости, и... веб-сервер просто не успевает реагировать на новые запросы пользователей и обрабатывать их, так как он занят передачей огромного массива данных с вашей графикой старым пользователям. На новые запросы пользователей, недоумевающих, что же так долго грузится сайт, система безопасности сервера реагирует, как на вирусную DOS-атаку, блокируя работу раздражителя, то есть вашего сайта. В результате вы лишаетесь благодарной аудитории до последующей модификации сайта. Кроме того, в стране в большинстве случаев Интернет еще не высокоскоростной и выделенные каналы — роскошь, особенно для небольших городов. По этим причинам перенасыщенные графикой страницы у обладателей низкоскоростного трафика ничего, кроме раздражения, не вызывают. Если на сайте не представлена жизненно необходимая информация, то пользователь просто закрывает его, не дожидаясь полной загрузки.

Как и в любой компьютерной области, программы работают с данными в виде файлов. Веб-графика не является исключением и может содержать свои данные в файлах векторных или растровых форматов.

Векторными называют изображения, созданные с помощью графических примитивов: прямых, окружностей, прямоугольников, дуг и т. д. Чем хорош векторный формат? Тем, что векторное изображение вычисляемо, то есть, если нужно нарисовать линию, программа должна знать ее начальную точку, направление и цвет, что достаточно просто вне зависимости от длины линии. При изменении масштаба самого изображения качество рисунка не меняется. Например, все шрифты хранятся в векторном виде, и, когда мы их увеличиваем или уменьшаем, мы просто заставляем машину пересчитывать координаты точек. Однако, если вы захотите нарисовать жезл инспектора дорожно-постовой службы, придется задать несколько векторов с белым и черным цветом заливки попеременно.

В отличие от векторных, в растровых рисунках программа явно задает координаты и цвет для каждой точки (пикселя). Таким образом, при рисовании инспекторского жезла черным и белым карандашами по очереди мы ставим черные или белые точки, записывая их координаты и цвет. Если мы будем сжимать жезл, то лишние пиксели будут безвозвратно удаляться, а если его растягивать, то образуются пустоты, которые нечем заполнить. Иначе говоря, при изменении масштабов самого рисунка существенно страдает его качество. Как видим, растровый способ записи изображений более объемный, но он чаще используется именно в веб-графике, поэтому поговорим о нем подробнее.

Любое растровое изображение состоит из фиксированного количества точек (пикселов), размер которых определяется разрешением экрана, то есть индивидуален для каждой его настройки. Следовательно, при настройке с малым или большим разрешением экрана один и тот же рисунок либо не по-

местится на экране, либо будет слишком куцым, но его размер останется постоянным. Это следует учитывать в первую очередь для планирования размещения графических объектов на страницах.

Традиционно принято считать, что веб-страница должна размещаться на экране с разрешением 800×600 пикселов, что оправдано и до сих пор, так как горизонтальную полосу прокрутки и предоставляемые ею возможности можно использовать только при просмотре больших панорамных объектов или фотографий. Однако это часто приводит к нарушению читабельности страницы и, как следствие, к отторжению аудитории.

Второе свойство растровых файлов — это глубина кодирования цвета, или так называемое битовое разрешение. Оно определяет количество битов, выделяемых для записи цвета одного пикселя. Иными словами, если вы используете для кодирования пикселя один бит, то пиксель может иметь только два цвета; если два бита, то четыре цвета; если три бита, то восемь цветов и т. д. Отсюда видно, что чем многоцветнее палитра вашего рисунка, тем больше битов для кодирования цветов пикселя используется. Это увеличивает объем рисунка и самой страницы, что напрямую сказывается на скорости получения страницы пользователем.

Исходя из расчетов и возможностей пропускных сетевых каналов принято, что вес веб-страницы, отягощенной графическими объектами, не должен превышать 60–75 Кбайт. Это гарантирует как нормальную работу веб-сервера, так и комфортную деятельность пользователя. Таким образом, не считая фона, не рекомендуется размещать на странице более трех глобальных графических объектов, имеющих большой объем. Это актуально, если, конечно, у вас не фотогалерея. Входящие в моду графические элементы интерфейса, такие как кнопки, обозначающие пункты меню, также не рекомендуется использовать, так как они отягощают веб-страницы графикой и уменьшают релевантность веб-страниц при их индексации в поисковых машинах.

Учитывая все положительные и отрицательные требования к веб-графике, можно определить типы (форматы) графических файлов, ставшие де-факто стандартами для веб-страниц, — JPEG (.jpeg), GIF (.gif), PNG (.png). Рассмотрим их более подробно.

Файлы, сохраненные в формате **JPEG**, получили наибольшее распространение и являются самыми популярными в веб-графике. Данный формат предназначен для хранения фотографических растровых изображений с использованием встроенного механизма сжатия, способного уменьшить исходный размер изображения до ста раз без видимой потери качества за счет удаления избыточной информации, не воспринимаемой глазом человека. Гибкие настройки при сохранении изображения в данном формате позволяют пользователю найти компромисс между качеством и объемом графического файла. Файлы уже являются архивными, и, как правило,

программами-архиваторами почти не сжимаются. При просмотре браузер или другая программа самостоятельно распаковывает файл во время открытия. При неудачно выбранном соотношении «объем — качество» либо чрезмерном увеличении рисунок может иметь заметную для глаз мозаичную структуру. Рекомендуется выбирать JPEG для фотографий или рисунков, на которых много однотонных мест и мало резких контрастных переходов, что является заметным при неудачном выборе пропорций сжатия.

Специально созданный в 1987 году для пересылки графической информации по сетям, формат **GIF** использует полное сжатие без потерь информации. Он позволяет выводить файл слету — сразу из Сети, используя чересстрочный вывод содержимого файла на экран. В этом случае сначала каждая восьмая, затем четвертая и т. д. строки появляются на экране, не заставляя пользователя долго ожидать полной закачки, а затем и загрузки файла. На место еще не полученных строк выводятся копии уже полученных, создавая иллюзию полностью загруженного изображения. Кроме того, очень привлекательна возможность сохранения в одном файле нескольких изображений с указанием времени задержки при их смене. Проще говоря, создается небольшой мультипликационный ролик, так называемая **GIF**-анимация, что делает страницу не статичной, а активной, оживляя ее регулярной сменой изображения. Файлы в **GIF**-формате позволяют установить прозрачный цвет или включить в содержимое файла текстовые строки.

Широко распространен еще один относительно новый графический формат **PNG**, представляющий переносимую сетевую графику. Важнейшие его достоинства — возможность сжатия без потерь качества как по горизонтали, так и по вертикали. В результате многократно уменьшается размер файла и поддерживается цвет пикселя с глубиной кодирования до 48 бит, что, в свою очередь, позволяет придавать пиксели любой цвет, начиная с прозрачного. Возможность сохранять анимированные файлы наподобие **GIF**-анимации доступна в другом варианте **PNG**-формата — формате **MNG**.

Выбор типа файла, в котором вы собираетесь сохранять свои графические объекты, зависит от вашего желания и возможностей. При этом можно пользоваться некоторыми рекомендациями. Формат **JPEG** применяется для фотографий или сканированных изображений, многоцветных сложно сочетаемых цветов, где человеческий глаз будет обманут заменой оттенков пикселов. Формат **GIF** удобен для четких рисунков с обозначенными границами и ограниченным количеством цветов, для содержащих текст изображений, рисунков, требующих прозрачного фона, анимированных рисунков. Для этих же целей можно использовать формат **PNG**.

6.2. Кисти, краски и холсты — обзор инструментов для подготовки веб-графики

Для обработки графических объектов используются различные программные пакеты, называемые графическими редакторами. Их достаточно много, и, в принципе, подготовить графический объект для размещения можно даже с помощью редактора Paint, встроенного в операционную систему Windows. Кроме того, ускоренными темпами начинает распространяться бесплатный графический редактор с открытым кодом — GIMP. Но на просторах нашей родины по-прежнему властвует профессиональный пакет от компании Adobe. Сюда включен и растровый графический редактор Adobe Photoshop, с помощью которого мы и будем рассматривать технологию создания и обработки графических объектов для веб-страниц.

Adobe Photoshop, как и любой другой программный продукт от фирмы Adobe, является коммерческим и распространяется на основе коммерческой лицензии. Эта лицензия весьма дорогая, но для привлечения покупателей — в маркетинговых целях — фирма предоставляет триальную, или пробную, версию своего продукта с ограниченным сроком работы в 30 суток с момента установки. Скачать программу можно с сайта компании www.Adobe.com. Предоставляемого времени должно хватить, чтобы можно было доказать начальству необходимость оплаты лицензионной программы и последующего ее использования на благо вашей фирмы, а также создания графического наполнения веб-страниц сайта на неопределенный промежуток времени.

Помимо Adobe Photoshop, имеет смысл скачать программы, входящие в пакет от Adobe. Это Adobe Flash для создания и обработки Flash-роликов и Adobe Dreamweaver — специальный редактор веб-страниц, позволяющий создавать и редактировать веб-страницы сайта с нуля. При этом можно абсолютно не иметь представления о языке гипертекстовой разметки страниц HTML, то есть это редактор, работающий по принципу «что вижу на экране конструктора, то и получаю на своей странице».

Для всех программ пакета от Adobe характерен одинаковый интерфейс, а также доступна миграция объектов из одного приложения в другое без потери качества. Это ориентирует покупателя на приобретение полного пакета, а не использование разных программ от различных производителей для совершения каких-либо действий с обрабатываемыми данными. Подобная унификация интерфейсов программ также позволяет пользователю, уже освоившему любую из программ пакета, начать работать с другими приложениями без дополнительного изучения меню и интерфейсных настроек. Интерфейс всех программ интуитивно понятный, и для каждого инструмента предусмотрена всплывающая подсказка с его названием. Кроме

того, доступна справочная система, но только при использовании легальной версии и подключении к Интернету.

После установки программы прежде всего ознакомьтесь с пунктами меню, пощелкайте, пооткрывайте и поймите логику приложения.

- **Файл** — все операции с файлами, соответственно, при работе с документом обращаемся к этому пункту меню.
- **Редактирование** — редактирование содержимого файла: манипулирование выделенными частями, масштабирование, трансформирование, настройка и т. д. — привычные операции, как и в любом офисном приложении.
- **Изображение** — работа с размером, кадрированием, яркостью и контрастностью, преобразование цветовых режимов и т. д., то есть операции непосредственно над выделенным изображением.
- **Слои** — замечательнейшее меню, позволяющее создать «слоеный пирог» из различных изображений; понадобится в процессе создания GIF-анимации.
- **Выделение** — говорит само за себя и позволяет выделить любую часть изображения по различным признакам.
- **Фильтр** — мечта художника, у которого украли кисти и краски, и он вынужден был рисовать шариковой ручкой на листе в клеточку, а затем пришел домой и отсканировал листок, чтобы с помощью меню **Фильтр**, знаний и фантазии воссоздать картинку в цвете и фотографическом качестве. Либо наоборот — превратил утро в вечер, а зиму — в лето. При использовании легальной программы есть возможность добавлять различные фильтры в коллекцию меню.
- **Просмотр** — демонстрация изображения в различных режимах.
- **Окно** — вывод окна с панелями, содержащими различные инструменты для работы с объектами. Сняв соответствующий флажок, вы убираете определенную панель с инструментами.

В различных версиях программы существуют дополнительные меню, и поэтому для более углубленного изучения возможностей программного пакета от фирмы Adobe имеет смысл обращаться к специализированной документации по установленному у вас пакету.

Обычно слева в окне программы находится панель инструментов. Захватив за верхнюю область, ее можно переместить в любое удобное место, как и любое другое окно. Подведя указатель мыши к инструменту и чуть удерживая его в неподвижности, можно получить всплывающую подсказку, а если нажать и чуть удержать инструмент левой кнопкой мыши, то можно получить палитру инструментов, из которых выбирается необходимый инструмент для работы (рис. 6.1).

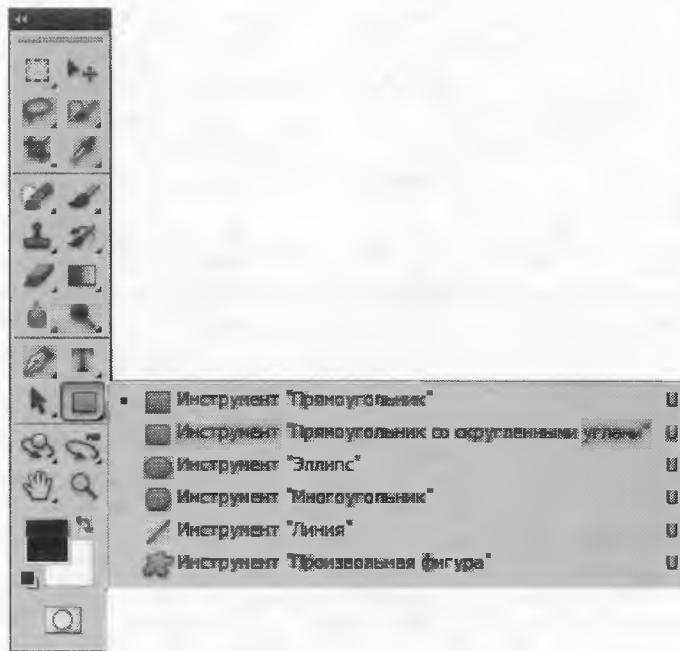


Рис. 6.1. Щелчок на значке инструмента и удержание над ним указателя мыши открывают палитру инструментов

Ознакомившись с интерфейсом программы Adobe Photoshop, можно перейти к рассмотрению некоторых приемов обработки графических объектов для размещения на веб-страницах.

Дополнительная информация об инструментах программы и ответы на популярные вопросы вынесены в раздел 6.6 в конце данной главы, чтобы сейчас не занимать время тех пользователей, которые торопятся.

6.3. Начинаем рисовать. Создание графических объектов для веб-страниц

Поскольку цель книги — изложение некоторых методов для создания сайтов, то и рассмотрение возможности создания графических объектов будем проводить в условиях, приближенных к боевым. Иначе говоря, рассмотрим непосредственно простые приемы создания только тех объектов, которые и будут размещены на веб-странице. Если вы думаете, что этого мало, уверяю вас — на самом деле вы ограничены только полетом своей фантазии, а я — объемом книги.

Создание кнопки-баннера. Для привлечения посетителей на сайт обычно принято обмениваться баннерными кнопками с сайтами соответствующей тематики. Если же необходимо провести коммерческую рекламу,

то администрация сайта распространяет свой рекламный баннер на других сайтах, используя баннерообменные сети или непосредственно покупая рекламное место у владельцев более популярных сайтов. В любом случае нужно создать графический объект — так называемую кнопку (или баннер), которая и будет лицом вашего сайта на страницах чужого. Изготовление баннера, конечно, можно заказать и при этом платить, но можно изготовить его самостоятельно.

При изготовлении нужно учитывать сложившиеся размеры баннеров, от которых и зависит стоимость занимаемого рекламного места. Следует также учитывать размеры баннеров, если вы изготавливаете элементы дизайна сайта. Дело в том, что современные браузеры имеют программные блоки, распознающие рекламу и блокирующие ее. Настройка этих блоков — полное право владельца компьютера, но если ваши графические элементы дизайна сайта будут по размерам совпадать с размерами рекламных баннеров, то очень велика вероятность того, что они будут опознаны браузером как реклама и не будут отображаться на экране пользователя. Типовые размеры рекламных баннеров приведены в табл. 6.1.

Таблица 6.1. Типовые размеры рекламных баннеров

Размер	Вид	Размер	Вид
300×250	Прямоугольник средней величины	120×90	Кнопка 1
250×250	Всплывающий квадрат	120×60	Кнопка 2
240×400	Вертикальный прямоугольник	120×240	Вертикальный баннер
336×280	Большой прямоугольник	125×125	Квадратная кнопка
180×150	Прямоугольник	728×90	Ведущий стенд
468×60	Длинный баннер	160×600	Широкий небоскреб
234×60	Половина длинного баннера	120×600	Небоскреб
88×31	Микрополоса	300×600	Объявление на полстраницы

Начнем изготовление баннера «Кнопка 1» по соответствующей классификации табл. 6.1. Размеры баннера соответственно 120 пикселов в ширину и 90 пикселов в высоту.

Запустите программу Adobe Photoshop и выполните команду **Файл > Создать**. В результате откроется окно создания файла, в котором и введем необходимые параметры (рис. 6.2).

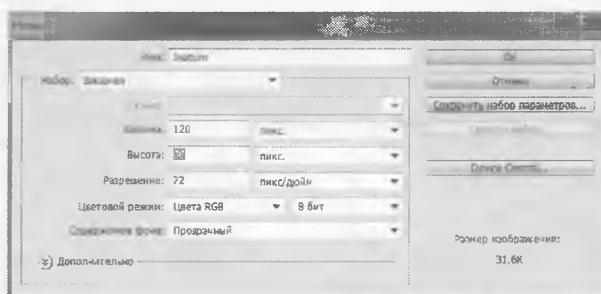


Рис. 6.2. Окно создания нового файла изображения

Файл будет иметь имя `bsaturn`, размеры 120×90 пикселов, разрешение оставим стандартное — 72 пикселя на дюйм, цветовой режим и глубина цвета — по умолчанию (чтобы не утяжелять маленькую картинку). Содержимое фона выставим прозрачным, чтобы баннер можно было разместить на любом фоне веб-страницы. Подтвердим выбор нажатием кнопки **OK** и получим в центре экрана маленький прямоугольник. Сразу возникает вопрос: «Как же там рисовать, он же очень маленький?» Спешу успокоить — для отображения обрабатываемого изображения или его части в любом удобном масштабе существует окно **Навигатор** (рис. 6.3). Передвигая ползунок на его нижней линейке, вы измените масштаб картинки. Перемещаться по рисунку можно, сдвигая выделенную красным цветом область квадрата на любую часть изображения.



Рис. 6.3. Панель Навигатор, отображающая изменение масштаба просмотра изображения

Увеличив размер рабочей области до приемлемого для работы уровня, создадим надпись с названием фирмы. Для этого выберем инструмент **Горизонтальный текст**, обозначаемый на панели инструментов буквой **T**, и растянем область текста по заготовке. Установив курсор в выделенную область так, чтобы он начал мигать, введем надпись: **Сатурн! Открытый космос ждет Вас!** Выравнивание, определение размера и начертания текста можно выполнять с помощью появившейся панели инструментов **Текст** (рис. 6.4).

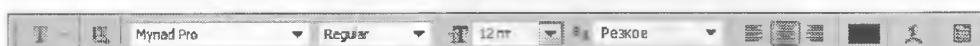


Рис. 6.4. Панель инструментов Текст

При работе с текстовыми шрифтами главное не увлекаться и помнить, что во главу угла ставится читабельность текста, а не его «завитушки». Цвет текста выбирается на упомянутой панели щелчком кнопкой мыши на инструменте под названием **Задайте цвет текста**, при этом открывается палитра доступных для текста цветов (рис. 6.5).



Рис. 6.5. Палитра выбора цвета текста

Для других объектов цвет задается на стандартной панели инструментов в области выбора цветов — основного и фонового (рис. 6.6).

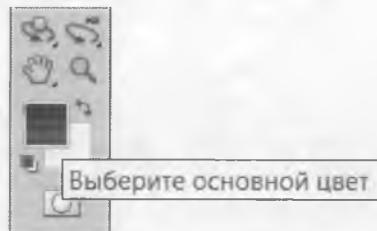


Рис. 6.6. Меню выбора цветов на стандартной панели инструментов

Щелкнув кнопкой мыши на квадрате, обозначающем основной или фоновый цвет, получим палитру выбора цветов (рис. 6.7).



Рис. 6.7. Палитра выбора цветов

Наигравшись с текстом, получим приблизительно следующую надпись (рис. 6.8).

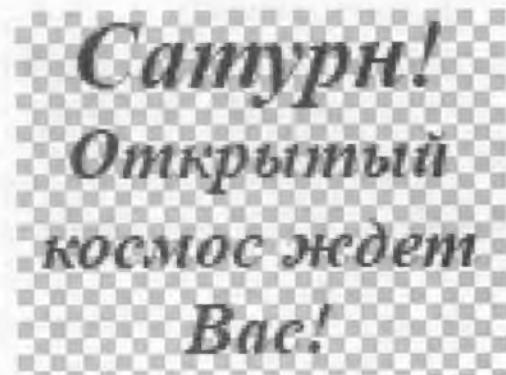


Рис. 6.8. Надпись на баннерной кнопке

Конечно, можно оставить и так — простейший баннер как графический объект готов, осталось только сохранить его, и можно будет добавлять на веб-страницы. Но мы не ищем легких путей. На баннере недостаточно одной надписи, пусть даже очень красивой. Нужно сделать ее более привлекательной, поэтому придется познакомиться со слоями.

Для начала придадим нашему баннеру вид кнопки. Сначала выбираем инструмент Прямоугольник со скругленными краями и рисуем им фигуру, вписывающуюся в область баннера. При этом появится новое окно Слой (рис. 6.9).

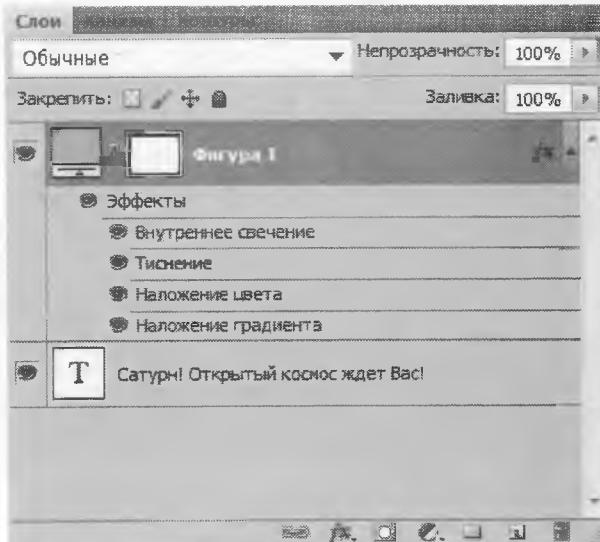


Рис. 6.9. Окно Слой

Как видно из рисунка, у нас появился второй слой **Фигура 1**, содержащий прямоугольник со скругленными краями. Для придания прямоугольнику стиля кнопки обратимся к окну **Стили** (рис. 6.10).



Рис. 6.10. Окно Стили

Выберем стиль, наиболее соответствующий нашему представлению о кнопке, из предложенной библиотеки стилей (рис. 6.11).



Рис. 6.11. Выбранный стиль бanniера

Осталось поменять местами стили таким образом, чтобы слой с надписью оказался поверх слоя с оформленным прямоугольником. Для этого сделаем активным слой **Сатурн! Открытый космос ждет Вас!** щелчком мыши (рис. 6.12).

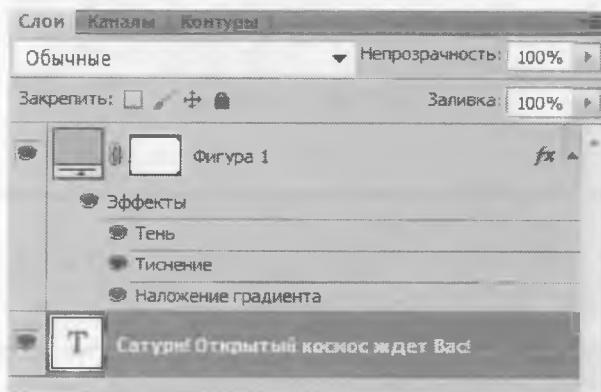


Рис. 6.12. Активизация слоя, содержащего надпись

Далее выполним команду **Слой** ▶ **Упорядочить** ▶ **На передний план**, и слой **Сатурн! Открытый космос ждет Вас!** разместится поверх слоя **Фигура 1** (рис. 6.13).

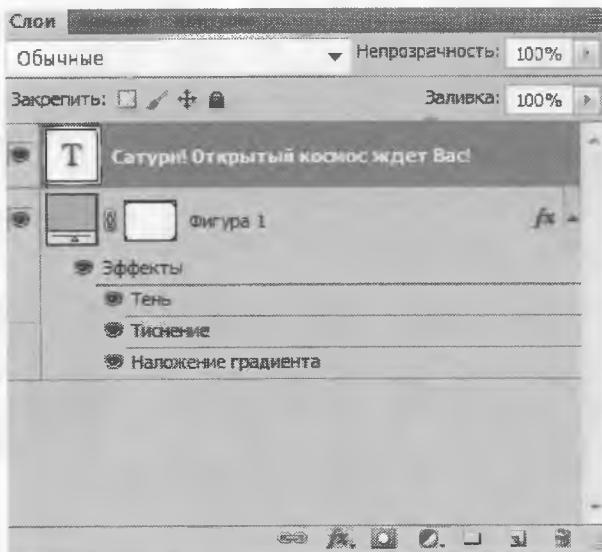


Рис. 6.13. Взаимное изменение расположения слоев

В результате наша кнопка примет следующий вид (рис. 6.14).



Рис. 6.14. Итоговый вид кнопки-баннера

Остается только сохранить файл с кнопкой, оптимизировав его для размещения на веб-странице. Выполним команду меню **Файл > Сохранить для Web**, и в результате появится окно, позволяющее выбрать параметры оптимизации файла (рис. 6.15).

При нажатии кнопки **Сохранить** появится стандартное окно сохранения документов, где и следует указать место и имя хранения файла (рис. 6.16).

Далее готовая кнопка используется по мере необходимости для установки в определенных местах веб-страниц. По аналогии можно создавать любые веб-объекты: управляющие кнопки, кнопки пунктов меню и эмблемы сайтов.

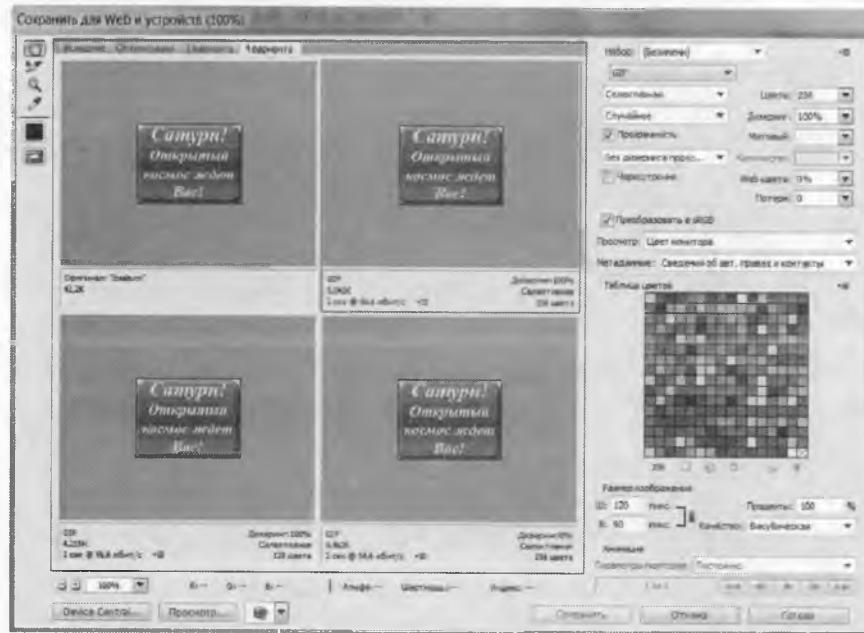


Рис. 6.15. Окно оптимизации файла при сохранении для Сети

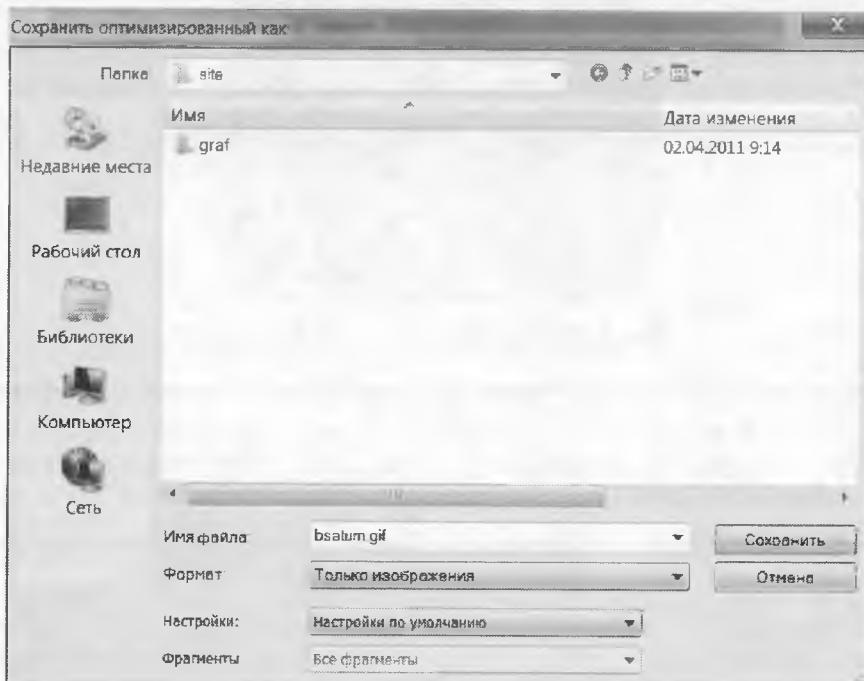


Рис. 6.16. Окно сохранения документа

6.4. GIF-анимация. Создание главного изображения сайта

Главное изображение сайта — заголовок, который первым встречает посетителя. Создадим его не статичным в своем поведении, как созданная выше кнопка баннера, а динамичным, то есть он будет изменять свое содержимое через некоторый промежуток времени. Таким образом мы создадим так называемую GIF-анимацию. Хотя это и противоречит требованиям к заголовкам, на этом примере мы рассмотрим применение фильтров и научимся создавать анимированные объекты.

Обращаю ваше внимание на следующие требования к заголовкам сайта, выполненным в виде графического объекта.

- Заголовок должен нести смысловую нагрузку.
- Заголовок при соответствии стилистическому содержанию страницы должен обращать на себя внимание.
- Все надписи заголовка должны быть четко читаемыми, воздержитесь от сложно воспринимаемых декоративных шрифтов.
- Эффекты анимации в заголовках следует применять с величайшей осторожностью, так как моргающий анимированный текст зачастую вызывает раздражение, а не интерес.

В начале работы необходимо создать файл с рисунком определенного размера. Здесь следует задуматься и посчитать. Классическая веб-страница рассчитывается на ширину экрана размером 800 пикселов — это принятое по умолчанию минимальное расширение экрана пользователя, равное 800×600 пикселов. Не учитывать это требование можно, но при этом сразу стоит смириться с потерей части аудитории, у которой не такие широкоэкранные мониторы, как у веб-мастера, поскольку горизонтальная прокрутка страницы вызывает только раздражение. В то же время жестко фиксированное в размерах изображение при высоком разрешении экрана будет выглядеть мелким, появятся пустые места, которые также вызывают только недоумение. Как же быть? Все изначально решается при планировании сайта, и в идеальном случае он вставляется в резиновую таблицу, меняющую свою ширину при изменении размеров окна браузера. Но изображения имеют жестко заданный размер в пикселях и растягиваться не могут. Что делать? Дам один простой совет: сделайте главное изображение слайда не сплошным, а фрагментарным. Например, в левой стороне будет встречающая надпись, а в правой — логотип, или наоборот. Поместите эти изображения в трехстолбцовую таблицу, правая и левая ячейки которой будут четко заданной ширины, имеющей размер изображений, а средняя ячейка пусть остается резиновой по ширине. Самое сложное в этой композиции — подобрать фон для всех ячеек и изображений таким образом, чтобы они образовывали один цельный объект.

Так мы и поступим — в левой части у нас будет встречающая надпись, а в правой — логотип. При общей ширине экрана 800 пикселов на достаточно солидную и привлекательную надпись можно выделить до 600 пикселов, оставив 200 на размещение картинки логотипа. Высоту можно выбирать на свой вкус, главное, чтобы изображение не съедало полезную просмотренную область экрана. Создадим файл с размером изображения 600×120 пикселов, выполнив те же действия, что и в предыдущем примере, только для фона выберем белый цвет (рис. 6.17).

Для создания фонового изображения первоначально необходимо выполнить градиентную заливку. Выберем для нашего изображения линейный градиент, чтобы впоследствии проще было организовать переход к другому изображению. Основным выберем голубой цвет, а фоновым — белый. Затем проведем указателем мыши с нажатой левой кнопкой слева направо по обрабатываемой области и определим для нашего объекта градиентную заливку как нисходящую к белому цвету (рис. 6.18).

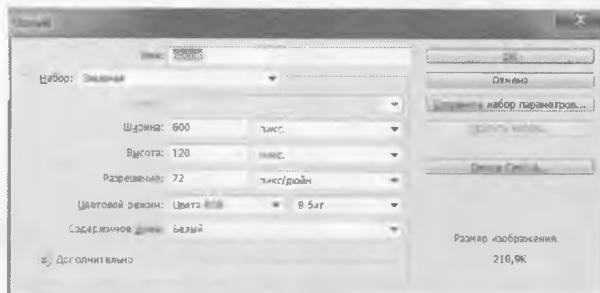


Рис. 6.17. Создание изображения надписи на белом фоне



Рис. 6.18. Придание градиентной заливки фону изображения надписи

Чтобы придать фону эффект движения, последовательно воспользуемся двумя фильтрами. Первый придаст эффект взрыва, разбив наш фон на разномастные блоки. Выполним команду **Фильтр** \blacktriangleright **Стилизация** \blacktriangleright **Экструзия** и увидим следующее окно (рис. 6.19).

Выбрав предлагаемые параметры, получим результат, как на рис. 6.20, где явно видны небольшие пирамидки, придающие фону эффект взрывного движения.

Теперь необходимо сгладить этот слишком жесткий рисунок. Для этого применим фильтр **Размытие** \blacktriangleright **Размытие по поверхности** и в полученном окне настройки выберем требуемые параметры (рис. 6.21).

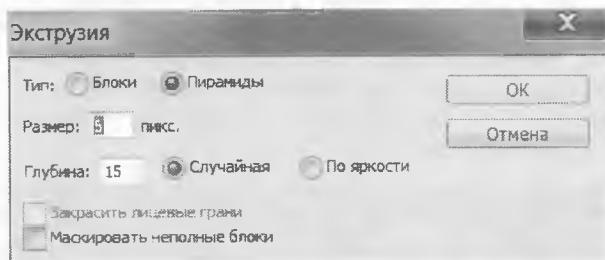


Рис. 6.19. Окно настройки фильтра Экструзия



Рис. 6.20. Применение фильтра Экструзия

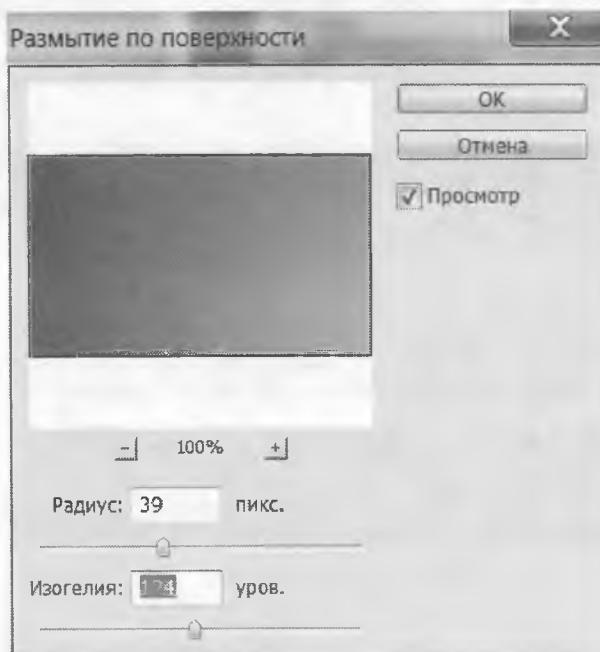


Рис. 6.21. Окно настройки фильтра Размытие по поверхности

В результате выполненных операций с фильтрами фон нашей надписи получился таким, как и ожидалось, — размыто-рябоватым, создающим иллюзию нестабильного движения (рис. 6.22).

Для придания большей космичности надписи создадим несколько звезд, которые впоследствии будут мигать на надписи. Выберем в стандартной панели инструментов кнопку **Многоугольник** и в появившемся окне настройки зададим дополнительные параметры в раскрывающемся меню **Параметры инструмента "Многоугольник"** (рис. 6.23).



Рис. 6.22. Подготовленный фон

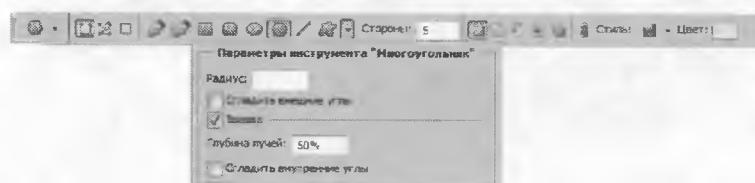


Рис. 6.23. Выпадающее меню Параметры инструмента "Многоугольник"

Расположим несколько звезд на поверхности фона, получив при этом следующее изображение (рис. 6.24).



Рис. 6.24. Фон с расположенными на нем звездами

Для завершающего штриха создадим две надписи, которые будут сменять друг друга: Компания "Сатурн" и Открытый космос у Ваших ног (рис. 6.25 и 6.26).

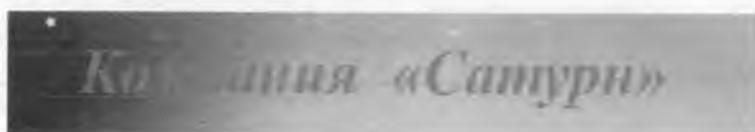


Рис. 6.25. Надпись Компания "Сатурн"

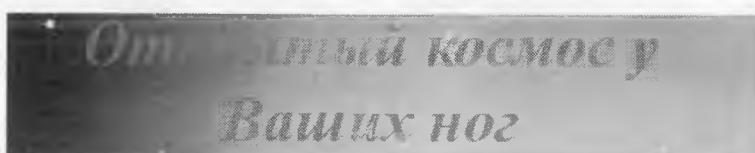


Рис. 6.26. Надпись Открытый космос у Ваших ног

Обратите внимание, что при создании надписи должны располагаться на разных слоях (рис. 6.27).

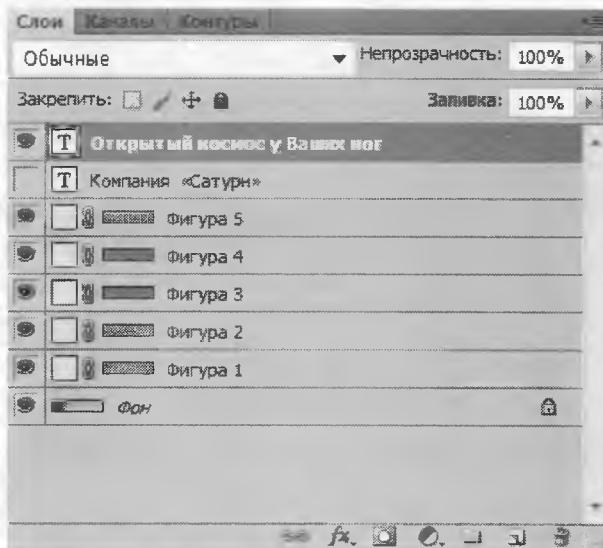


Рис. 6.27. Список слоев для открытого документа

Как видно из рис. 6.27, текущим (выделенным синим цветом) является слой *Открытый космос у Ваших ног*, а слой *Компания "Сатурн"* является неотображаемым. Для активизации необходимого слоя следует щелкнуть на нем кнопкой мыши. Чтобы сделать слой отображаемым или нет, нужно установить или снять глаз, расположенный слева от имени слоя.

Документ готов к созданию GIF-анимации. Технология ее создания заключается в посменном включении и отключении видимости различных слоев в файле. Таким образом, в течение одного промежутка времени на экран выводится одна группа изображений, а на протяжении другого — другая. Количество слоев и возможностей анимированных объектов ограничено только здравым смыслом: с каждым слоем размер файла увеличивается, что приводит к его долгой загрузке, зато после загрузки из Сети файл к ней больше не обращается.

Для создания GIF-анимации выполним команду **Окно ▶ Анимация**, в результате откроется окно покадровой анимации, которое обычно появляется в нижней части рабочей области (рис. 6.28).

Поскольку у нас всего две записи, то можно воспользоваться только двумя кадрами для смены слоев, но мы сделаем четыре — для организации дополнительных пауз между кадрами. Добавьте дополнительно три кадра на линейку раскадровки, щелкнув на значке с изображением папки с отогнутым левым нижним углом (рис. 6.29).

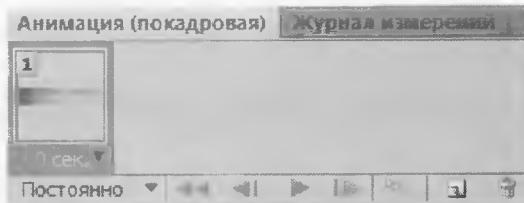


Рис. 6.28. Окно меню Анимация



Рис.6.29. Созданные четыре кадра для GIF-анимации

Теперь, выделяя каждый кадр по отдельности, включим для него только те слои, которые будут отображаться. Например, для первого кадра это слои, указанные на рис. 6.30.

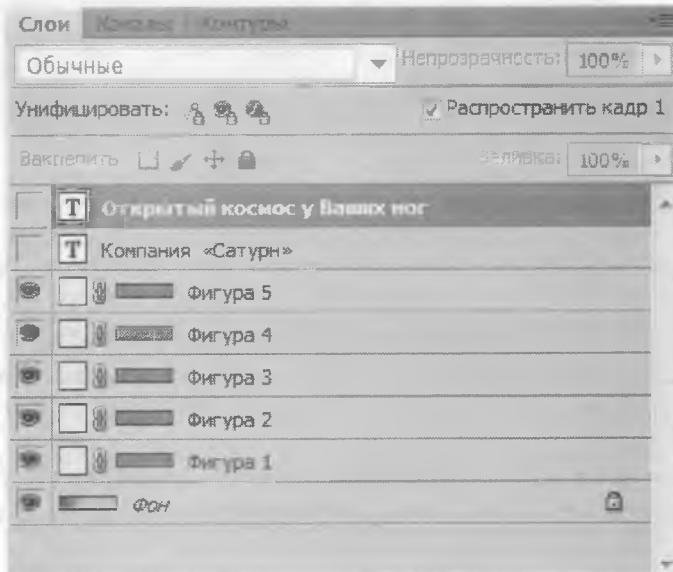


Рис. 6.30. Слои, отображаемые в первом кадре

Для второго кадра будут отображаться слои, указанные на рис. 6.31.

Для третьего кадра будут отображаться слои, указанные на рис. 6.32.

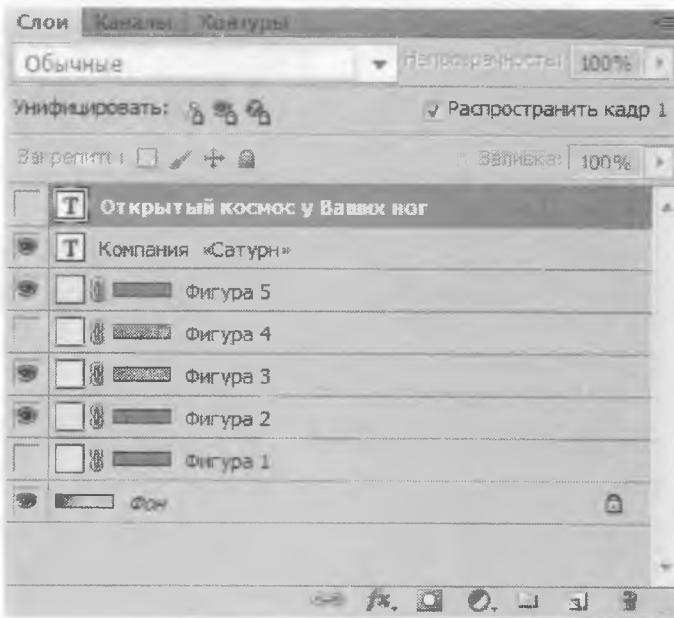


Рис. 6.31. Слои, отображаемые во втором кадре

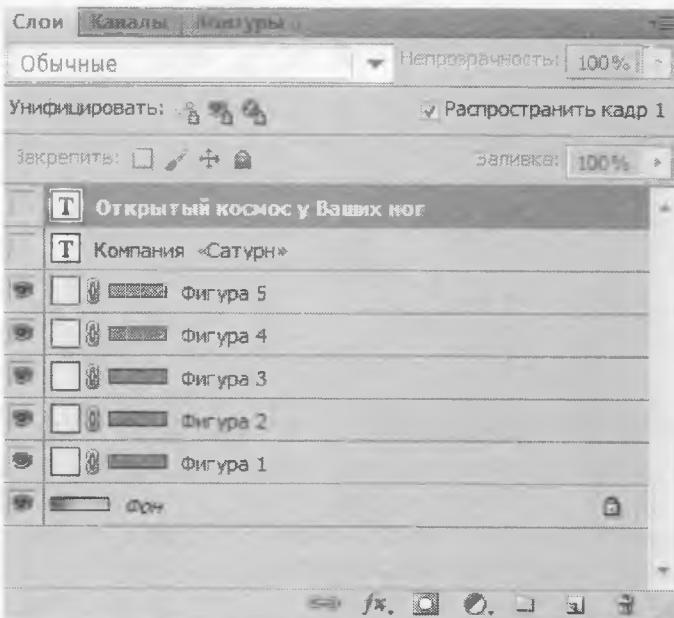


Рис. 6.32. Слои, отображаемые в третьем кадре

И для четвертого кадра это будут слои, показанные на рис. 6.33.

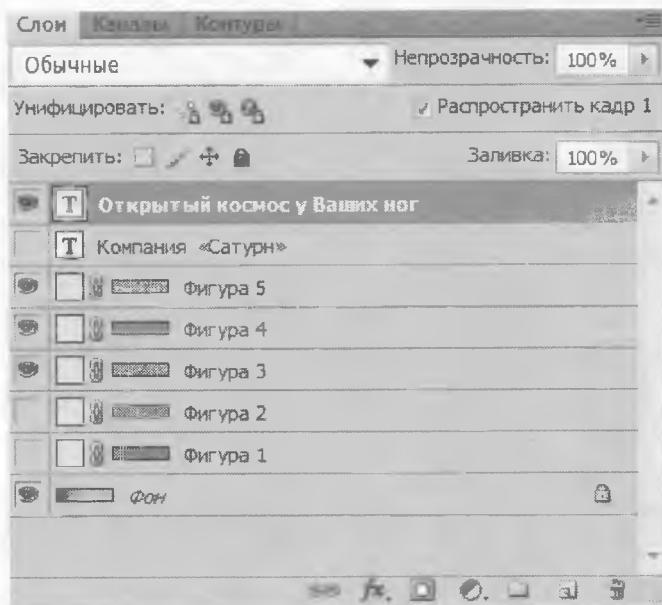


Рис. 6.33. Слои, отображаемые в четвертом кадре

Теперь необходимо установить для каждого кадра время смены, щелкнув на треугольнике внизу кадра. При этом появится меню с вариантами времени задержки, из которых и выберем устраивающий нас (рис. 6.34).

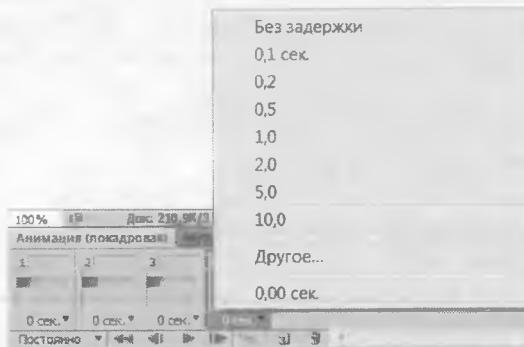


Рис. 6.34. Меню для выбора времени задержки для смены кадров в анимации

После выставления времени задержки на шкале повторений цикла кадров обязательно задайте режим Постоянно (рис. 6.35).

Рядом с режимом смены кадров, справа, показана панель проигрывателя (см. рис. 6.35). Нажав здесь кнопку воспроизведения, можно увидеть выполнение анимированного файла еще в процессе его создания и оценить эффективность временных задержек смены кадров для их корректировки. Выставленные мной временные интервалы представлены на рис. 6.36.

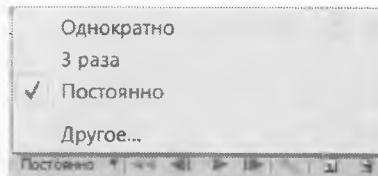


Рис. 6.35. Установка режима постоянного отображения смены кадров



Рис. 6.36. Временные интервалы смены кадров

Далее выполняем уже знакомую и описанную выше процедуру: сохраняем файл в формате GIF. Для этого нужно выполнить пункт меню **Файл > Сохранить для Web** (рис. 6.37).

Рассмотренный способ создания GIF-анимированных изображений при достаточной степени усидчивости позволит вам создавать шедевры, ограниченные только вашей фантазией.

На этом можно было бы и закончить основные приемы работы с графикой для веб-страниц, но существует еще один способ применения и обработки веб-графики, который мы и рассмотрим далее.

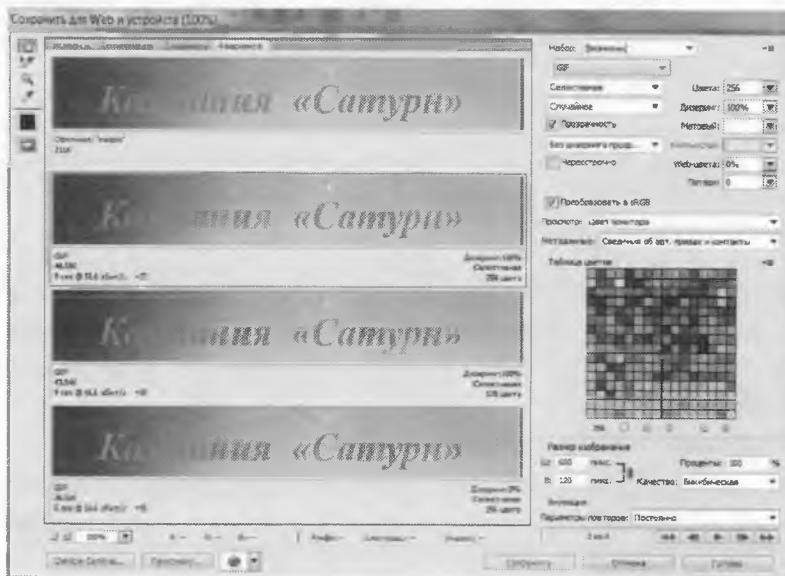


Рис. 6.37. Сохранение файла для Сети

6.5. Основы фотомонтажа

Монтаж изображения из различных частей. Достаточно часто случается, что необходимо скомпоновать изображение для веб-страницы из частей различных изображений или фотографий — создать так называемый фотомонтаж, как, например, на рис. 6.38.



Рис. 6.38. Пример создания фона на основе совмещения различных графических объектов

Как видно из рисунка, есть несколько наложенных друг на друга изображений. Кроме того, рисунок является анимированным — девушка, сидящая слева, покачивает ногой.

Технология создания подобных изображений заключается в использовании слоев и выделении отдельных графических изображений, взятых из разных файлов и размещенных на слоях одного файла. Иными словами, нужно из одного места вырезать и в другое вставить. Рассмотрим эти технологические приемы на примере изображений с пейзажем (рис. 6.39) и башней (рис. 6.40).

Условия задачи просты: поместить башню на побережье.

Сначала нужно открыть оба файла в программе Adobe Photoshop и перейти к рисунку с башней. Инструментом Прямоугольное выделение (рис. 6.41) следует выделить интересующую нас башню (рис. 6.42).



Рис. 6.39. Пейзаж



Рис. 6.40. Башня

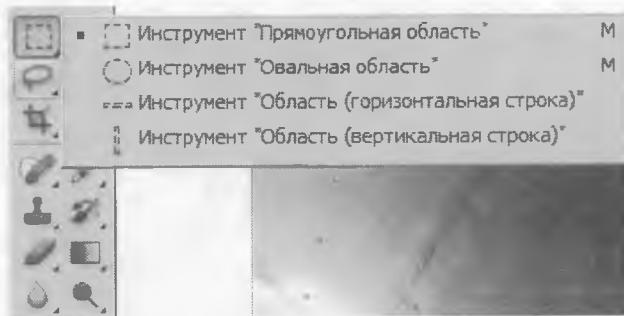


Рис. 6.41. Инструмент Выделение



Рис. 6.42. Выделенная башня

Затем вызовем контекстное меню выделенной области, щелкнув правой кнопкой мыши, и выберем пункт **Скопировать на новый слой** (рис. 6.43).

Будет создан новый слой, содержащий только выделенную башню. Далее ее необходимо повернуть и избавить от ненужного фона. Поэтому сначала отключим видимость фонового слоя (рис. 6.44).

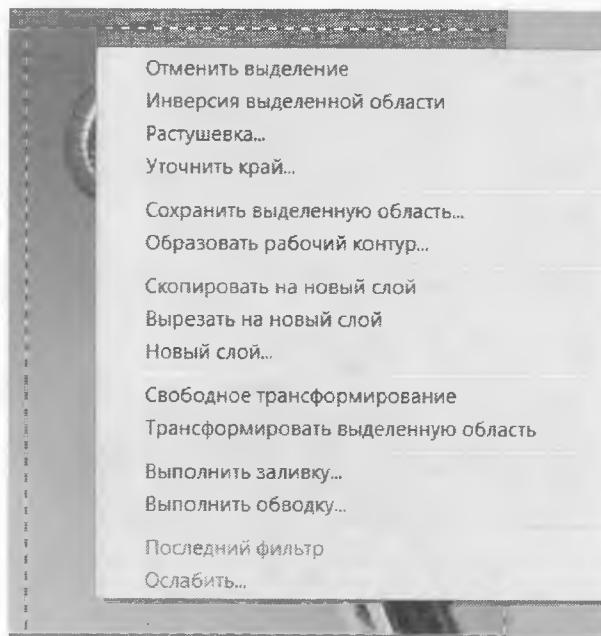


Рис. 6.43. Контекстное меню выделенной области

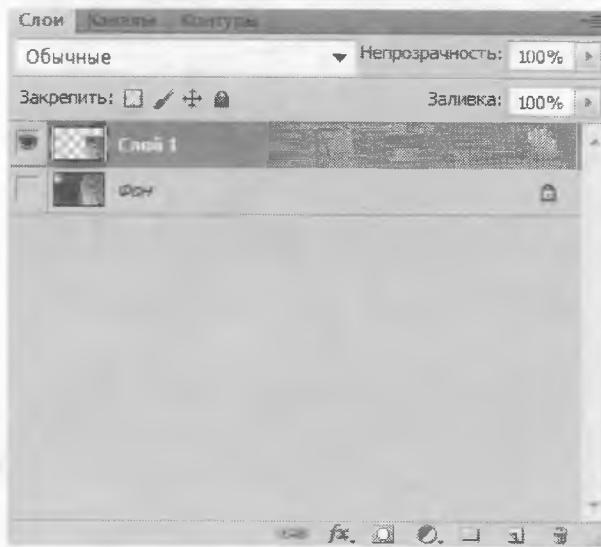


Рис. 6.44. Отключение видимости фонового слоя

Развернем башню вертикально. Для этого опять выделим ее инструментом **Выделение** и выберем пункты главного меню программы **Редактирование** ▶ **Свободное трансформирование**. В результате по области выделения рамка примет

движущийся вид, а при наведении указателя мыши на любой угол выделенной области он будет принимать различную форму. Нас на данный момент интересует двусторонняя стрелка, выгнутая дугой. Поймав с ее помощью угол картинки, нажмите левую кнопку мыши и, удерживая, поворачивайте выделенную область до тех пор, пока башня не примет вертикальное положение (рис. 6.45). После этого подтвердите изменение нажатием клавиши Enter и снимите выделение.



Рис. 6.45. Разворот башни в вертикальное положение

Приступим к удалению фонового изображения неба. В этом случае можно взять инструмент **Ластик** и начать по-детски все стирать, что достаточно долго и требует повышенного внимания. Мы же будем использовать инструмент **Волшебная палочка**, который предназначен для выделения пикселов с одинаковым цветом (рис. 6.46).

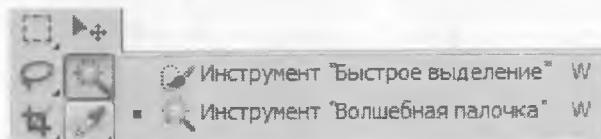


Рис. 6.46. Инструмент Волшебная палочка

Касаемся волшебной палочкой фоновой области и полученную выделенную область удаляем нажатием клавиши **Delete**. Для более удобной работы можно увеличивать масштаб отображения в окне Навигатор (рис. 6.47).

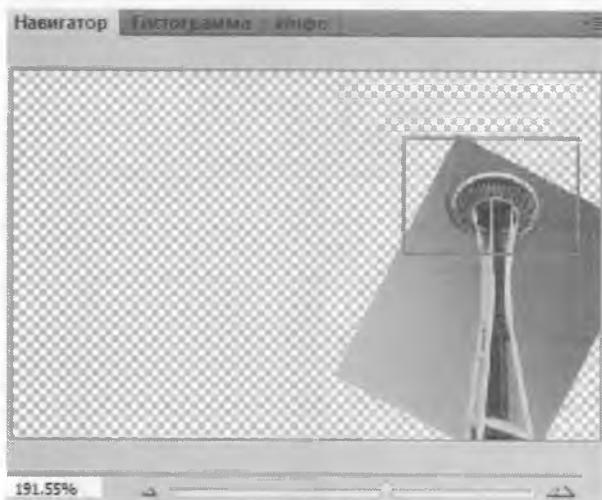


Рис. 6.47. Использование окна Навигатор для изменения масштаба отображения

После грубой очистки волшебной палочкой мелкие ограхи можно стереть и ластиком, предварительно выбрав для него минимальный размер кисти в окне Кисть (рис. 6.48).

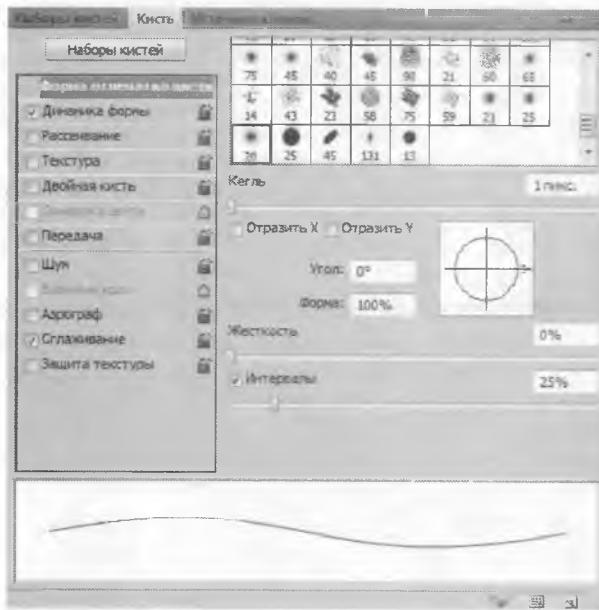


Рис. 6.48. Окно выбора кисти

После выполнения вышеописанных действий мы получим чистую башню, готовую к перестановке (рис. 6.49).



Рис. 6.49. Готовый рисунок с объектом перестановки

Далее выделим башню и выберем пункт меню Редактирование ▶ Скопировать, перейдем на вкладку файла с изображением пейзажа и выполним команду Редактирование ▶ Вставить. В результате получим изображение, как на рис. 6.50.

Для размещения башни на скальном берегу в очередной раз воспользуемся инструментом Выделение, а затем выберем пункт меню Редактирование ▶ Свободное трансформирование. В этом случае возьмем мышью изображение и переставим его на берег. Затем нужно поймать в углу двустороннюю диагональную стрелку и пропорционально изменить размер выделенного изображения (рис. 6.51).

Как видно на рис. 6.51, башня стоит не натурально. Для исправления этого дефекта разместим ее основание за камнями, создав впечатление, будто башня выглядывает из-за мыса. Увеличим масштаб просмотра в окне Навигатор, вооружимся ластиком и с максимально возможной аккуратностью сотрем основание башни, чтобы оно описывало очертание камней. Итоговое изображение приведено на рис. 6.52.



Рис. 6.50. Добавление башни в область пейзажа



Рис. 6.51. Башня, установленная на берегу



Рис. 6.52. Готовый пейзаж со вставленным изображением

Для публикации осталось сохранить итоговый файл, но перед этим необходимо превратить двухслойный рисунок в однослоиный. Для этого выберите пункты меню **Слои** ▶ **Выполнить сведение**. Теперь сохраните файл обычным способом: меню **Файл** ▶ **Сохранить для Web**.

Как видно из приведенных выше примеров, подготовка изображений для размещения на страницах сайта не представляет особой сложности и сводится к простым действиям: нарисовать — стереть — вырезать — вставить. Все ограничено только фантазией и художественным вкусом веб-дизайнера.

Дополнительные инструменты и приемы работы с ними, рассмотренные в данной главе, позволяют сделать изображения более интересными и даже технологичными. Как показывает практика, этих приемов вполне достаточно для графического оформления простых веб-страниц. Если же изложенного материала оказалось мало, рекомендую обращаться к специализированной литературе по программе Adobe Photoshop.

6.6. Часто возникающие вопросы и ответы на них

Какое разрешение должно быть у изображений для публикации на веб-страницах?

Типовое разрешение для публикации на веб-страницах определяется как 75 пикселов/дюйм, но можно оставлять предлагаемое по умолчанию значение 72 пикселя/дюйм.

Что такое оптимизация графики для веб-страниц?

Поскольку основной критерий оценки графики — это объем файла, то вся оптимизация графики для веб-страниц сводится к подбору параметров, определяющих соотношение «качество изображения — размер файла». Оптимизация выполняется после выполнения команды **Файл** ▶ **Сохранить для Web**, при этом в появившемся окне можно выбрать параметры оптимизации.

Можно ли изменить размер изображения, если да, то как?

Размер изображения, как и размер холста, на котором оно находится, можно изменить, выполнив команду **Изображение** ▶ **Размер изображения (Размер холста)**. В появившемся окне можно задать нужные параметры (рис. 6.53).

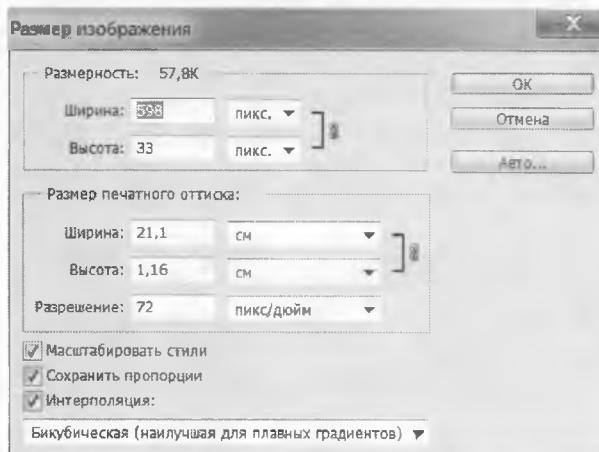


Рис. 6.53. Окно изменения размеров изображения

Как кадрировать изображение?

Для кадрирования изображения (отсечения ненужной графической информации, расположенной вне зоны выбранного кадра) используют инструмент **Рамка** (рис. 6.54). Размер кадра настраивается перемещением боковых и угловых маркеров, после чего нужно щелкнуть кнопкой мыши внутри выделенной области и подтвердить процедуру нажатием клавиши **Enter**.

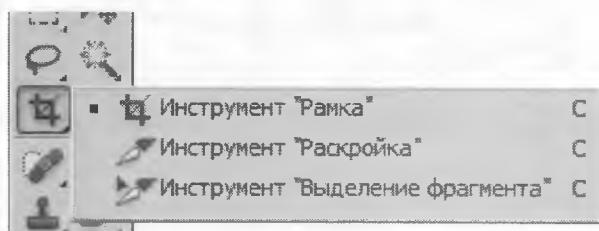


Рис. 6.54. Инструмент Рамка для выполнения операции кадрирования

Каким образом можно работать со светом изображения (осветлять, затемнять рисунок)?

Изображение можно осветлять с помощью инструментов Осветлитель/Затемнитель, показанных на рис. 6.55. При осветлении/затемнении выделаемой области на панели свойств объекта можно задать значение изменения параметра.



Рис. 6.55. Инструменты для осветления/затемнения локальной области

Для осветления/затемнения всего изображения в целом следует использовать инструменты из меню: Изображение ▶ Коррекция ▶ Яркость/Контрастность (рис. 6.56), Изображение ▶ Коррекция ▶ Уровни (рис. 6.57), Изображение ▶ Коррекция ▶ Кривые (рис. 6.58). Перемещая треугольные бегунки на шкалах, можно высветлить/затемнить изображение целиком.

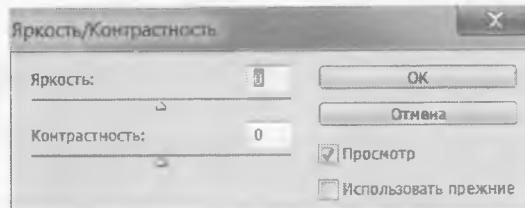


Рис. 6.56. Окно Яркость/Контрастность

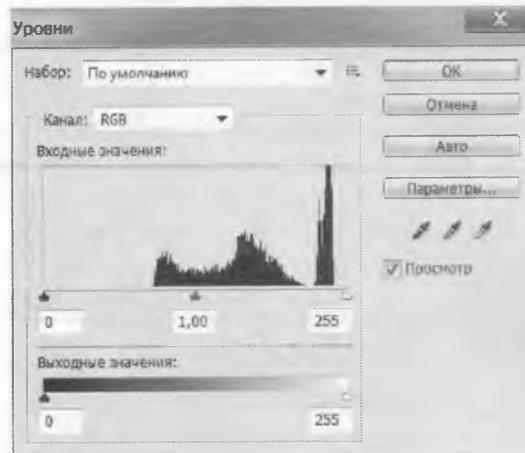


Рис. 6.57. Окно Уровни

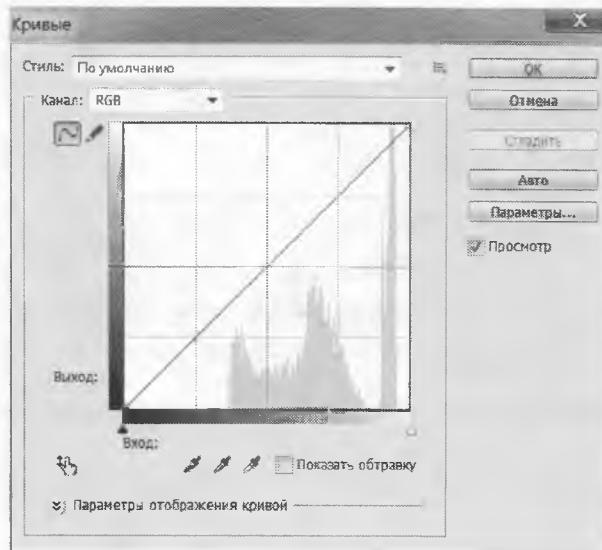


Рис. 6.58. Окно Кривые

Как убрать эффект красных глаз?

Для удаления эффекта красных глаз, часто возникающего на фотографиях, используется инструмент с таким же названием (рис. 6.59). Предварительно следует увеличить обрабатываемую область и проштамповать ее щелчками мыши.

Можно ли усилить резкость изображения?

Резкость изображения можно усилить как для локального участка, так и для всего изображения в целом. Для усиления локальной резкости следует воспользоваться инструментом Резкость (рис. 6.60). Для усиления резкости всего изображения следует выполнять команды из меню Фильтр ▶ Усиление резкости.

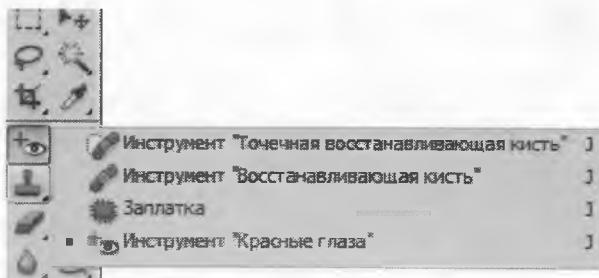


Рис. 6.59. Инструмент Красные глаза



Рис. 6.60. Инструмент Резкость для усиления локальной резкости изображения

Как устраниить царапины и другие технические дефекты на изображении?

Для устранения технических дефектов изображения следует использовать инструмент Штамп (рис. 6.61). Первоначально следует выбрать размер инструмента в окне Кисть, чтобы он соответствовал размерам дефекта. Нужно определить область исходной информации как можно ближе к дефекту, ведь именно оттуда штамп будет брать информацию для закрашивания, при этом область должна находиться не ближе диаметра кисти инструмента. Удерживая нажатой клавишу Alt, нужно щелкнуть кнопкой мыши на определенной ранее области и, отпустив клавишу Alt, начать заштамповывать дефект. При работе с инструментом Штамп следует учитывать то, что область исходной информации будет перемещаться, как перемещается и сам инструмент.



Рис. 6.61. Инструмент Штамп

Как убрать паразитный узор?

Паразитный узор, или так называемый муар, возникающий обычно при сканировании изображений, можно убрать с помощью фильтров из группы Фильтр ▶ Шум.

Глава 7

ТЕХНОЛОГИЯ FLASH

7.1. Введение во Flash-анимацию

Все большую популярность в мире компьютеров получает технология Flash. Сначала были робкие попытки создания анимированных заставок, а теперь на Flash написаны не только рекламные баннеры, но и сами сайты. Единственное и непременное условие для успешного применения Flash — это упорный труд в освоении данного программного продукта и неограниченная фантазия создателя.

Но освоение любого нового программного продукта, как и любой новой профессии, требует времени и повторений, то есть получения опыта. Мы же обычно жестко ограничены во времени при глубоком изучении всех технологических аспектов программ. Нам всегда нужно, чтобы было быстро и красиво, ну и живенько так. В этой главе мы вкратце рассмотрим способы создания Flash-ролика, на котором кольца Сатурна все-таки будут вращаться.

Интерфейс программы Adobe Flash

Начать знакомство с программой Adobe Flash, как всегда, придется с освоения интерфейса. Первоначально программный продукт Flash был реализован фирмой Macromedia, которую в дальнейшем купила компания Adobe. Несмотря на это, основные концепции самой программы и ее интерфейс остались почти не измененными с первых версий, соответственно, приемы работы, описанные в этой главе, подойдут к другим версиям.

При первоначальном запуске Adobe Flash нас встречает окно приветствия (рис. 7.1), предоставляющее выбор определенных действий при создании или открытии файла проекта. В зависимости от версии программы окно может отличаться от приведенного.

Как видно из содержимого окна, предоставляется возможность открыть ранее созданный документ, создать документ на основе предлагаемых шаблонов либо создать файл приложения. Пока мы не собираемся штурмовать глубины Flash, поэтому создадим файл ActionScript 3.0. Отличие данного типа файла от ActionScript 2.0 заключается только в расширенной

поддержке встроенного языка программирования. Чтобы окно приветствия больше не появлялось при запуске, достаточно установить флажок **Больше не показывать** в левом нижнем углу окна.



Рис. 7.1. Окно приветствия программы

Сразу следует сохранить созданный файл, выполнив стандартную команду **Файл > Сохранить**. В появившемся окне задайте место и имя сохранения документа. Обратите внимание на определенное отличие Adobe Flash от Adobe Photoshop. В первой программе файл с внесенными изменениями сохраняется в двух форматах: файл, сохраненный в формате FLA, содержит код документа и поэтому может быть открыт только в программе Adobe Flash, в то же самое время одноименный файл, сохраненный в формате SWF, содержит готовый клип-ролик, который можно проигрывать в любом Flash-проигрывателе. Например, возьмите уже знакомый нам по программе Photoshop инструмент **Текст** и добавьте с его помощью надпись на поле сцены (рис. 7.2).

Начинаем понимать то, про что читаем :)

Рис. 7.2. Первая надпись с помощью инструмента Текст

Теперь выполним команду **Файл > Сохранить** и запустим созданный клип на просмотр, нажав сочетание клавиш **Ctrl+Enter**. Можно выполнить команду **Управление > Тестировать ролик** и посмотреть, куда был сохранен изначальный файл (рис. 7.3).

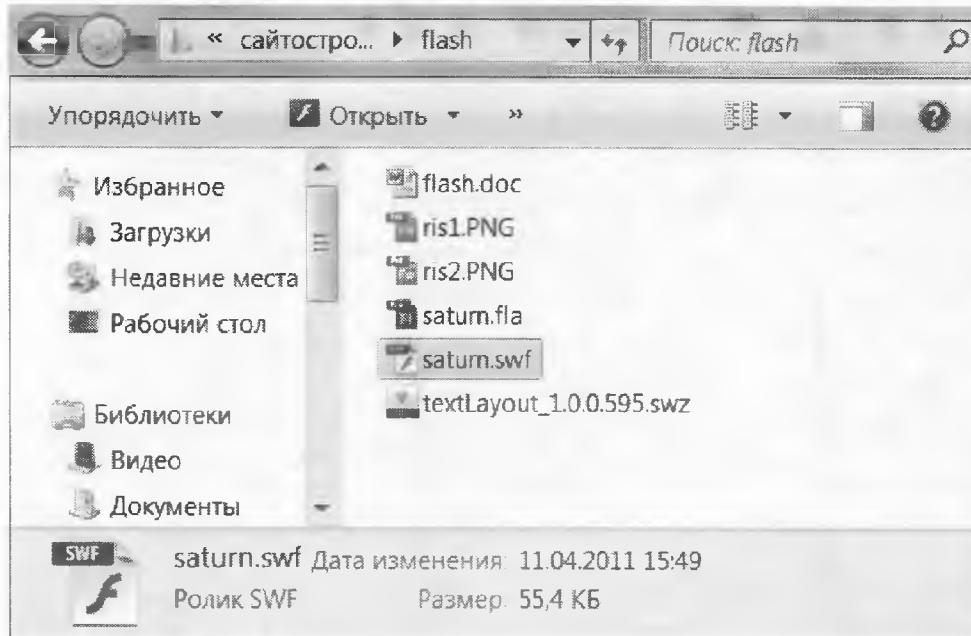


Рис. 7.3. Автоматическое сохранение готового клипа в формате SWF при тестировании

Появившийся готовый файл с расширением SWF и есть искомый продукт, ссылку на который можно вставлять на свою веб-страницу. Поэтому после работы и изменения материала не забывайте тестировать ролик для обновления конечного файла.

Поскольку редактор Adobe Flash — векторный, то есть изображения в нем рисуются с помощью графических примитивов (линий, окружностей, дуг и пр.), то меню и особенности работы в нем отличаются от работы в растровых графических редакторах. Отсюда и вытекают некоторые особенности в терминологии. Например, рабочей областью называется все серое пространство вокруг белой основной сцены, на которой и происходят все задуманные режиссером действия. Таким образом, рабочая область играет роль кулис, или театральных подмостков.

Пункты меню в основном стандартные и отвечают за выполнение следующих операций.

- **Файл** — все операции с файлами.
- **Редактирование** — манипуляция выделенными объектами: вырезать, вставить, скопировать и пр.
- **Просмотр** — просмотр клипа в различных режимах, настройка разметочных сеток и мерных линеек, изменение масштаба просмотра сцены.
- **Вставка** — вставка различных объектов: символов, слоев и пр.
- **Модификация** — модификация, внесение изменений в создаваемый клип: группировка объектов, трансформация объектов, их выравнивание, сглаживание и т. д.
- **Текст** — обработка текста.
- **Команды** — управление командами, написанными на языке ActionScript.
- **Управление** — управление воспроизведением ролика.
- **Отладка** — операции, вызывающие диалоговые окна для отладки ролика.
- **Окно** — отображение/скрытие окна с инструментами.

Благодаря возможности перетаскивать, включать и выключать окна с инструментами в любом порядке можно организовать рабочую среду так, как удобно лично вам. Если же тот или иной инструмент не находится в рабочей области, то включите его через пункт меню **Окно**.

Общий вид рабочего пространства помогает представить рис. 7.4.

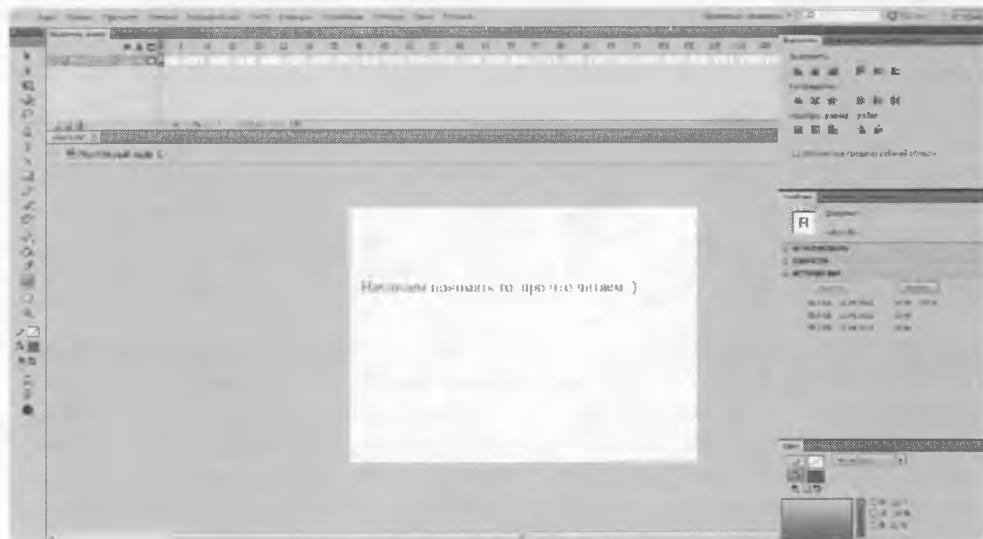


Рис. 7.4. Примерная организация рабочего пространства

В верхней части окна обычно располагают временную шкалу, по слойно отображающую содержимое сцены для каждого кадра ролика.

Анимация. Еще немного теории

В предыдущей главе при рассмотрении возможностей программы Adobe Photoshop мы познакомились с понятием GIF-анимации, что позволило создать анимированный заголовок сайта. Чем же GIF-анимация отличается от анимации, созданной во Flash?

Во Flash предусмотрены три разных механизма создания анимации изображения.

- **Классическая** — покадровая анимация, при которой изображения прорисовываются отдельно на каждом кадре и затем просматриваются как кинопленка.
- **Автоматическая анимация** (tweened — анимация) — пользователь прорисовывает только первый и последний кадры сцены движения, а программа уже достраивает промежуточные кадры. В свою очередь, автоматическая анимация разделяется:

- на анимацию движения (motion) — основана на перемещении объекта;
- анимацию формы (shape) — основана на трансформации (изменении) формы объекта.

- **Анимация, базирующаяся на выполнении сценариев на языке ActionScript**, — в этом случае изменение свойств и поведение объектов задаются программным кодом.

Прежде чем создавать анимацию, необходимо определиться, каким из способов будет сделан ролик, поскольку у каждого из методов есть свои достоинства и недостатки. Следует также отметить, что программа позволяет экспортить анимированные ролики в формат GIF, поэтому создать простейшую GIF-анимацию во Flash гораздо проще, чем в Photoshop.

Общую схему создания ролика можно описать следующим образом. Поскольку во Flash используется объектно-ориентированный подход, то любое действие связано с изменением определенных свойств обрабатываемых объектов. Применительно ко Flash вместо термина «объект» используют термин «символ». Символ представляет собой шаблон объекта с определенным набором свойств и хранится в библиотеке символов, поэтому может быть использован в различных кадрах или роликах. Каждая новая копия символа, располагаемая в кадре, называется экземпляром символа и несет весь набор характерных для него свойств. Таким образом, при наличии в кадре нескольких экземпляров символа их свойства не дублируются и позволяют сократить размер итогового файла.

Соответственно, динамика ролика обеспечивается за счет изменения в течение определенного промежутка времени какого-либо свойства символа: координаты на сцене, цвета, размера и пр. С каждым состоянием экземпляра связывают отдельный кадр ролика, а кадр, соответствующий изменению состояния экземпляра, называют ключевым. Он, в свою очередь, сам является экземпляром и может быть изменен.

Смена кадров в ролике описывается с помощью элемента интерфейса Временная шкала, где задаются параметры: частота смены кадров, моменты начала или завершения движения (трансформации) объектов и т. д.

В ролике может быть использовано различное количество объектов, поэтому, чтобы при изменении своих свойств они не взаимодействовали с другими объектами, их размещают на отдельные слои.

Таким образом, можно определить основную последовательность действий, выполняемую при создании ролика. После создания файла-заготовки в окне Свойства задаются размеры сцены и значение цвета фона. При нажатии кнопки Изменить появляется окно параметров документов (рис. 7.5). Обращаю ваше внимание на то, что в различных версиях программы элементы интерфейса могут различаться.

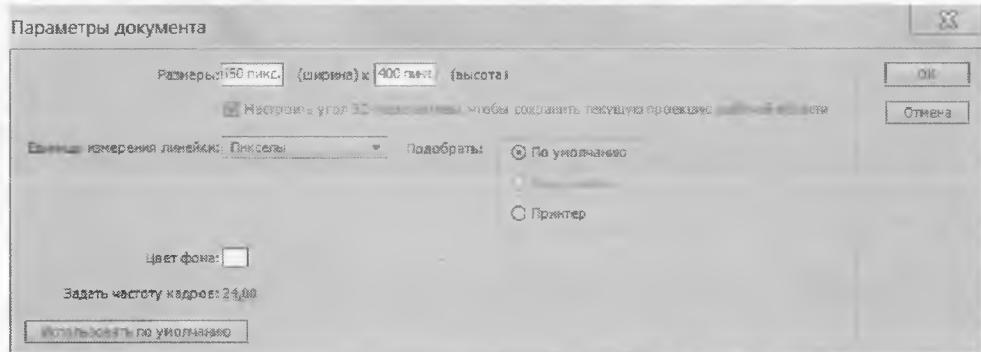


Рис. 7.5. Окно изменения параметров документа, вызванное через окно Свойства при установке свойств фона

Из содержимого окна видны все установки, доступные для свойства страницы. Единственное, на что хочется обратить ваше внимание,— значение Задать частоту кадров. Здесь указывается количество кадров, проигрываемое в течение 1 секунды показа ролика. Соответственно, чем больше число, тем больше кадров в ролике и тем он объемнее. Кроме того, необходимо учитывать и физиологию человека — особенности нашего зрения таковы, что мы воспринимаем процесс непрерывным в том случае, если картинка сменяется 24 раза за секунду. Таким образом, частота 24 кадра в секунду является максимально возможной для глаза человека, то есть более высокая частота смены кадров не оправдана физиологически. Сложившиеся же

требования с учетом скорости загрузки страниц рекомендуют устанавливать для простых анимационных роликов значение 12 кадров в секунду. Если же ролик сложный и многослойный, а также отягощен большим объемом, то рекомендуют уменьшить значение до 8 кадров в секунду.

Наконец, последнее теоретическое отступление перед началом работы — рассмотрим, что такое кадры.

Кадры во Flash бывают двух типов: ключевые и обычные. Ключевой — кадр, в котором задаются изменения в анимации, то есть это кадры, где определяются первоначальные и конечные свойства объекта. Ключевых кадров всегда как минимум два — первый и последний кадры ролика, но и на протяжении всего ролика свойства объекта могут неоднократно изменяться, в результате чего увеличивается количество ключевых кадров. При этом стоит помнить следующее: определение ключевых кадров ролика возможно только для расчетной анимации, когда промежуточные кадры додрисовывает сам компьютер. Для покадровой анимации, когда кадры рисует автор, ключевыми являются все кадры ролика.

На временной шкале ключевой кадр обозначается маленьким черным кружочком в нижней части прямоугольника, обозначающего кадр (рис. 7.6).

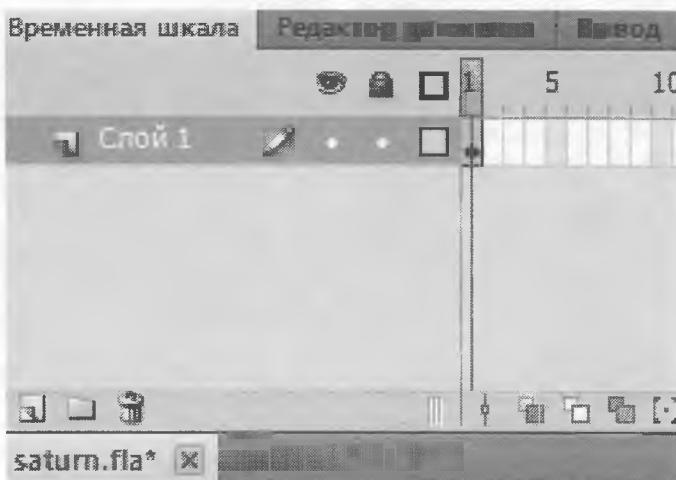


Рис. 7.6. Обозначение ключевого кадра

При создании изображения на первом кадре он становится ключевым автоматически. Ключевыми также автоматически становятся все кадры, на которых изменяются свойства отображаемых объектов. **Обычный** кадр обозначается пустым прямоугольником и включает в себя содержимое предшествующего ключевого кадра. Пустые кадры обозначаются прямоугольниками белого цвета, а кадры, содержащие объекты, — прямоугольниками серого цвета.

7.2. Построение изображений в слоеном пироге

Слой первый. Фон

Создайте файл, в котором будете сохранять свой рисунок — в нашем случае это `saturn.fla` — и задайте размеры сцены. Как помните из прошлой главы, мы оставили в графическом заголовке страницы место размером 200×120 пикселов, чтобы поместить логотип сайта в графическом заголовке. Теперь воспользуемся этими заранее запланированными размерами.

Обозначим первый слой как фон, вызвав из его контекстного меню пункт Свойства (рис. 7.7).

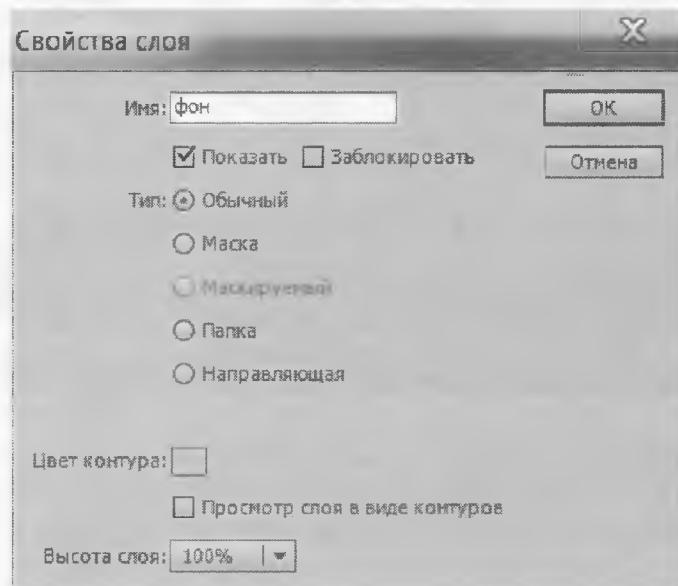


Рис. 7.7. Окно Свойства слоя

Выберем инструмент **Прямоугольник** и зададим свойства его цвета: отсутствие обводки и градиентная заливка в цвет, аналогичный цвету ранее подготовленного графического заголовка (рис. 7.8).

Нарисовав прямоугольник по всему объему сцены, получим следующий результат (рис. 7.9).

Теперь разместим на фоне несколько звездочек, чтобы было некоторое соответствие с графическим заголовком из предыдущей главы. Для этого выберем инструмент **Многоугольник** и в параметрах инструмента зададим форму звезды (рис. 7.10).



Рис. 7.8. Выбор градиентной заливки фонового прямоугольника



Рис. 7.9. Градиентная заливка слоя Фон, выполненная с помощью инструмента Прямоугольник

Нарисуем одну звездочку и, поскольку их у нас будет несколько, сразу преобразуем ее в символ для последующей записи в библиотеку символов. Для этого в контекстном меню звезды выберем пункт Преобразовать в символ и в появившемся окне укажем имя нового символа, в нашем случае звезда (рис. 7.11).

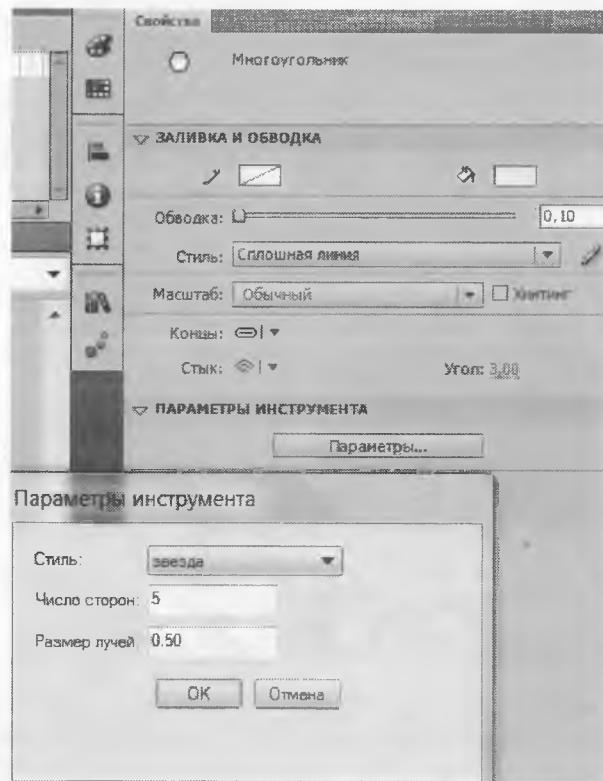


Рис. 7.10. Окно установки стиля «звезда» для инструмента Многоугольник

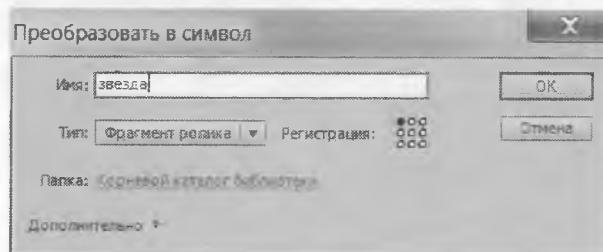


Рис. 7.11. Окно добавления символа «звезда» в библиотеку символов

Теперь, скопировав звездочку, можно вставить ее несколько раз и разбросать экземпляры символа по полю экрана, немного трансформировав их (рис. 7.12).



Рис. 7.12. Итоговое изображение фона со звездами

Затем придадим звездам эффект мерцания по аналогии с GIF-анимацией, когда мы отключали изображение той или иной звезды на определенном слое. Только теперь необходимо отключить изображение звезды в кадре. Для этого выделим на временной шкале группу кадров либо просто пройдем по кадрам указателем мыши с нажатой левой кнопкой. Можно также щелкнуть кнопкой мыши на прямоугольнике кадра, удерживая нажатой клавишу **Ctrl**. Когда кадры будут выделены, откроем их контекстное меню, щелкнув правой кнопкой мыши на выделенной области, и выберем пункт **Вставить ключевой кадр**. Когда в выделенных кадрах появятся кружочки, обозначающие, что кадр стал ключевым, переходим из кадра в кадр и просто удаляем те или иные звезды, лучше всего подряд на нескольких кадрах. После успешного завершения операции можно нажимать сочетание клавиш **Ctrl+Enter** и лицезреть мерцающее космическое небо.

На этом с фоном закончим. Теперь перейдем к следующим слоям, на которых изобразим сам Сатурн и его кольца. Перед этим заблокируем слой фона, чтобы случайно не внести изменений, и сделаем его невидимым, чтобы он нам не мешал. Для этого на временной шкале отключим соответствующие пиктограммы для слоя **фон** (рис. 7.13).

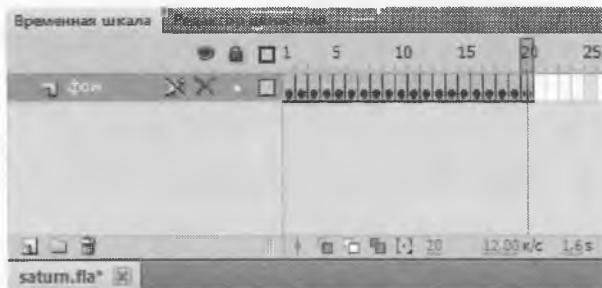


Рис. 7.13. Отключение видимости слоя и установка запрета на его редактирование

Слой второй. Сатурн

Для создания изображения Сатурна прежде всего необходимо создать новый слой, который так и назовем — **сатурн**. На временной шкале щелкните на значке создания слоя (рис. 7.14), затем откройте окно со свойствами и переименуйте слой.



Рис. 7.14. Значок Создание слоя

Таким образом, временная шкала уже должна выглядеть, как на рис. 7.15.



Рис. 7.15. Состояние временной шкалы после создания слоя под названием сатурн

Возьмите инструмент **Овал** и с помощью соответствующих значков укажите для него отсутствие цвета контура и наличие радиальной градиентной заливки (рис. 7.16).

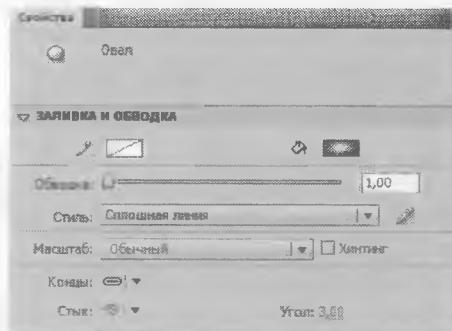


Рис. 7.16. Параметры инструмента Овал

Затем, чтобы планета была круглой, нарисуйте круг, удерживая нажатой клавишу **Shift**. Для выравнивания полученного градиентного шара по центру кадра необходимо воспользоваться значками на панели инструментов **Выровнять** (рис. 7.17).

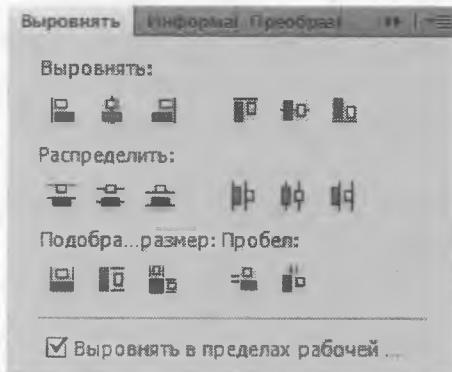


Рис. 7.17. Панель инструментов Выровнять

Для изменения цвета градиента нашего рисунка воспользуемся окном инструментов Цвет (рис. 7.18).

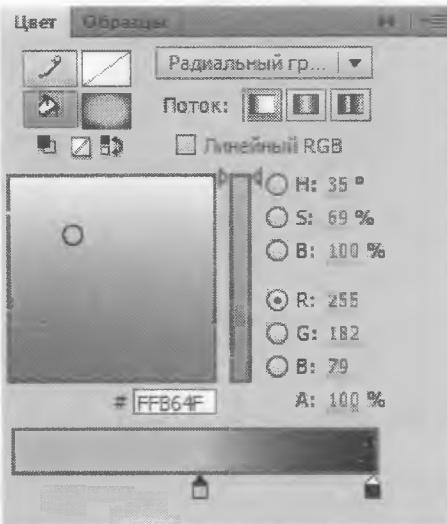


Рис. 7.18. Окно выбора параметров инструмента Цвет

Обратите внимание на нижнюю шкалу окна. Она отображает начальный и конечный цвет градиентной заливки. Передвигая ползунки, обозначающие пороги градиента, можно задать стиль изменения цветового перехода. Щелкнув на ползунке кнопкой мыши, можно придать ему свой цвет. Кроме того, можно создать новый порог изменения цвета, щелкнув на шкале, или убрать лишний, просто утащив его вниз, под шкалу. Опираясь цветовыми значениями инструмента Цвет, придадим цвет нашему Сатурну. На рис. 7.19 показано, как теперь выглядит сцена кадра.

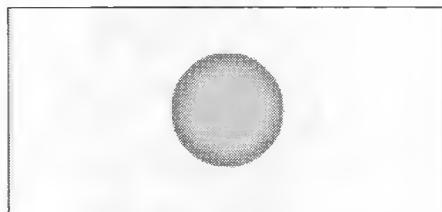


Рис. 7.19. Сцена кадра с объемным изображением Сатурна

Придадим эффект вращения планете. Для этого воспользуемся анимацией формы и заставим перемещаться по поверхности планеты выраженное пятно градиента, тем самым имитируя процесс вращения объекта. Редактирование градиентной заливки объекта выполняется с помощью инструмента Преобразование градиента. Выбрав его и щелкнув кнопкой мыши на объекте, получим достаточно своеобразное выделение объекта (рис. 7.20).

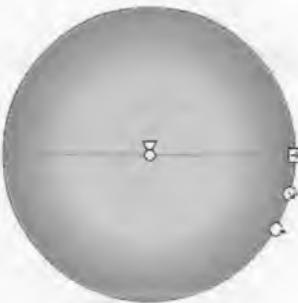


Рис. 7.20. Выделение объекта инструментом Преобразование градиента

Появившийся контур фигуры обозначает границу градиента, квадратный маркер справа на оси позволяет изменять ширину градиента, придавая ему форму овала, верхний круглый маркер, расположенный ближе к квадратному, дает возможность менять радиус градиентной заливки, а нижний круглый маркер обеспечивает угол наклона градиента. Центральный маркер, в свою очередь, позволяет изменять расположение центра градиентной заливки. Изменяя градиент с помощью описанных маркеров, можно придать обрабатываемой фигуре любой эффект объема.

Перейдем на первый ключевой кадр и, взяв центральный маркер мышью, переместим его за пределы окружности в левую сторону и чуть вверх, придав эффект рассвета над планетой (рис. 7.21).

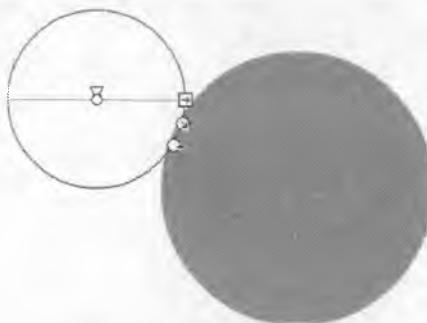


Рис. 7.21. Смещение области градиента для создания эффекта рассвета

Затем перейдем на последний кадр, назначим его ключевым, вызвав контекстное меню кадра, и переместим область градиента симметрично центру шара, создав эффект заката над планетой (рис. 7.22).

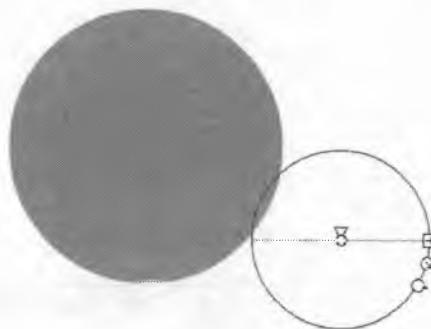


Рис. 7.22. Смещение области градиента для создания эффекта заката

Для создания анимации формы необходимо щелкнуть на поле с именем слоя — в нашем случае **сатурн**, чтобы выделить все входящие в него кадры. Затем нужно открыть контекстное меню выделенных слоев и выбрать пункт **Создать анимацию формы**. После этого первый и последний ключевые кадры на временной шкале соединяются стрелкой, указывающей на то, что будет произведена расчетная анимация (рис. 7.23).

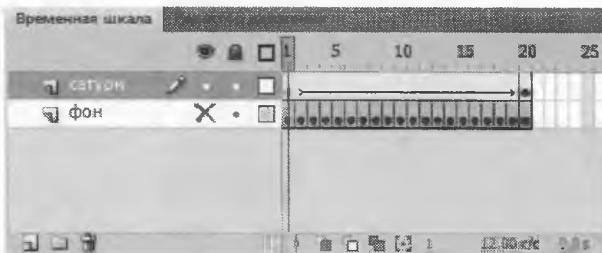


Рис. 7.23. Определение на временной шкале расчетной анимации формы для слоя **сатурн**

Теперь нажмите сочетание клавиш **Ctrl+Enter** и наблюдайте рассвет и закат над планетой (рис. 7.24).



Рис. 7.24. Просмотр ролика с демонстрацией рассвета и заката на фоне мерцающих звезд

Слой третий. Кольца

По аналогии создадим новый слой и назовем его кольца. При этом заблокируем слой сатурн, но оставим его видимым, используя значки, показанные на рис. 7.25.



Рис. 7.25. Блокирование для внесения изменений в слой фон и сатурн, при котором слой сатурн доступен для просмотра

Для анимации движения колец будем также использовать анимацию формы, а для этого необходимо создать кольцо. Воспользуемся инструментом Эллипс, нарисовав один эллипс внутри другого и затем удалив середину. Последовательность действий приведена на рис. 7.26–7.28.



Рис. 7.26. Рисование внешнего контура кольца

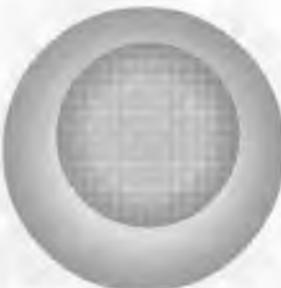


Рис. 7.27. Рисование внутреннего контура кольца



Рис. 7.28. Удаление внутреннего контура кольца, в получившемся проеме виден Сатурн

Затем следует развернуть получившееся кольцо, придав ему вид эллипса, как будто мы смотрим на него под углом. Для этого выберем инструмент **Свободное преобразование** и, используя пограничные маркеры, придадим кольцу следующий наклонный вид (рис. 7.29).

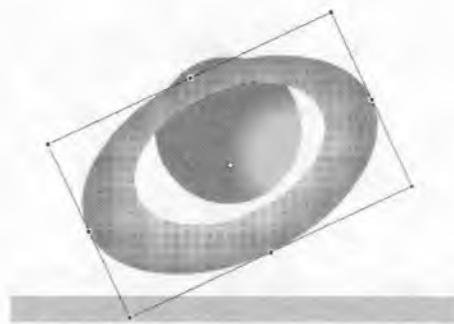


Рис. 7.29. Придание кольцу эллиптической формы

Выбрав инструмент **Специвыделение** и щелкнув кнопкой мыши на внутренней границе кольца, увидим область действия маркеров, которыми, в свою очередь, можно изменить внутреннюю часть кольца (рис. 7.30).



Рис. 7.30. Изменение внутренней части кольца путем перемещения области, выделенной инструментом Специвыделение

Промежуточный результат выполненных операций показан на рис. 7.31.



Рис. 7.31. Готовое кольцо

Для придания кольцу эффекта вращения в очередной раз воспользуемся инструментом **Преобразование градиента**. Сдвинем центр градиента к ближайшему левому краю кольца (рис. 7.32).

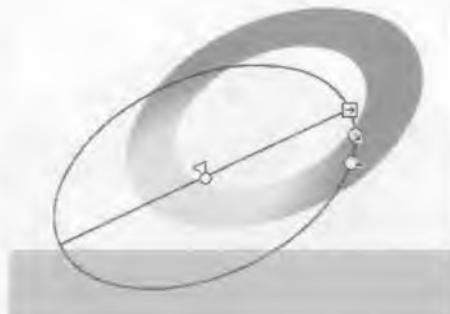


Рис. 7.32. Смещение центра градиента кольца

Далее, поскольку движение кольца будет определяться перемещением центра градиента по его поверхности, разбиваем кольцо на четыре части, в которые и будут добавлены ключевые кадры. В каждом созданном ключевом кадре устанавливаем центр градиента в свою позицию. Последовательность действий показана на рис. 7.33–7.36.



Рис. 7.33. Смещение центра градиента в нижнюю позицию

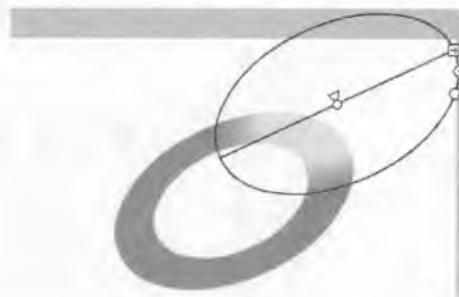


Рис. 7.34. Смещение градиента в крайнюю правую позицию

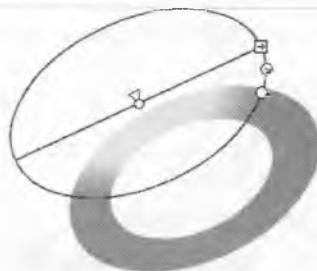


Рис. 7.35. Смещение градиента в верхнюю позицию

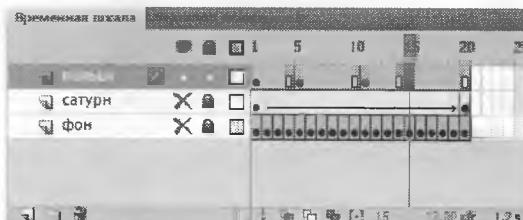


Рис. 7.36. Состояние временной шкалы после создания и изменения ключевых кадров

Осталось выделить все кадры и выбрать в их контекстном меню пункт Создать анимацию формы. Нажмем сочетание клавиш **Ctrl+Enter**: кольцо вертится и планета крутится (рис. 7.37).



Рис. 7.37. Демонстрация вращения кольца Сатурна

Существует только одна маленькая проблема: нужно поместить заднюю часть кольца за Сатурном. Решить эту задачу просто: перетащите мышью слой сатурн выше слоя кольца (рис. 7.38).



Рис. 7.38. Взаимное изменение расположения слоев сатурн и кольца

Результат не заставит себя долго ждать (рис. 7.39).

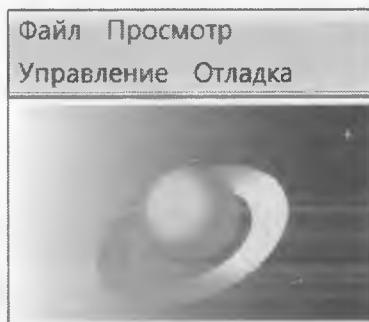


Рис. 7.39. Итоговый ролик с правильным расположением слоев

Вот такая красота: звезды мерцают, планета вертится, кольца движутся — полный космос. Осталось сюда как-то нашу компанию подключить — запустим космический корабль.

7.3. Как научить летать. Переход на веб-страницы

Слой четвертый. Покорители космоса — анимация движения

Заблокируем и отключим для показа все созданные ранее слои и создадим новый слой, назвав его корабль. Приложив максимум фантазии, нарисуем космический корабль, бороздящий просторы вселенной, например, как на рис. 7.40.

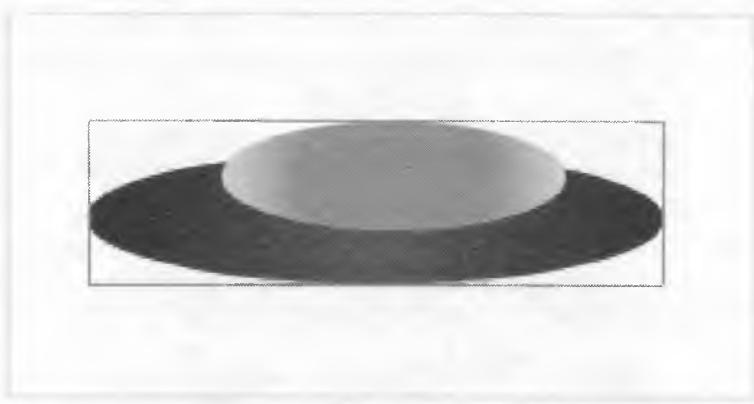


Рис. 7.40. Космический корабль

Теперь нужно выделить оба эллипса, образующих корпус нашего корабля, и выбрать пункт меню **Модификация** ▶ **Группировать**, тем самым превратив два объекта в один. После этого откроем контекстное меню объекта и выберем **Преобразовать в символ** — это необходимо для последующей покадровой анимации. Используя инструмент **Свободное трансформирование**, уменьшим корабль и перенесем его в точку старта (рис. 7.41).



Рис. 7.41. Точка старта космического корабля

Далее на последнем кадре, который нужно сделать ключевым, установим точку финиша корабля. Это также лучше сделать за пределами слайда для создания иллюзии прохождения мимо. Выделим кадры слоя и выберем в их контекстном меню пункт **Создать классическую анимацию движения**. В результате мы увидим движение корабля по бескрайним просторам космоса на фоне Сатурна (рис. 7.42).

Казалось бы, поставленная задача выполнена, можно демонстрировать впечатляющие результаты: вращающаяся планета, движущиеся кольца, еще и летающая тарелка...

Но здесь следует учитывать психологию человека: ведь обязательно захочется щелкнуть на движущемся объекте, а там ничего не происходит. Непорядок, надо, чтобы была какая-то реакция на щелчок.

Следовательно, предвосхитим задачу, которую в обязательном порядке поручит нам решать начальство, — сделаем ссылку с данного объекта на главную страницу сайта `index.html`.



Рис. 7.42. Демонстрация анимации движения.
Корабль движется на фоне Сатурна

Слой пятый, завершающий. Переход на страницу `index.html`

Для этого создадим пятый слой и назовем его переход. Далее возьмем инструмент **Прямоугольник** и нарисуем прямоугольник прямо поверх нашей рабочей сцены, чтобы он полностью ее перекрывал. Затем выделим кадры слоя, вызовем их контекстное меню и выполним пункт **Действия**, в результате чего откроется окно, как на рис. 7.43.

Как видно из рисунка, окно предназначено для написания кода **ActionScript**, описывающего действия с объектами. Не нужно пугаться и переживать, что вы не владеете этим языком. Надо щелкнуть на надписи **Фрагменты кода**, которая расположена в правом верхнем углу окна. В результате откроется одноименное окно, содержащее распространенные фрагменты кода **ActionScript** (рис. 7.44).

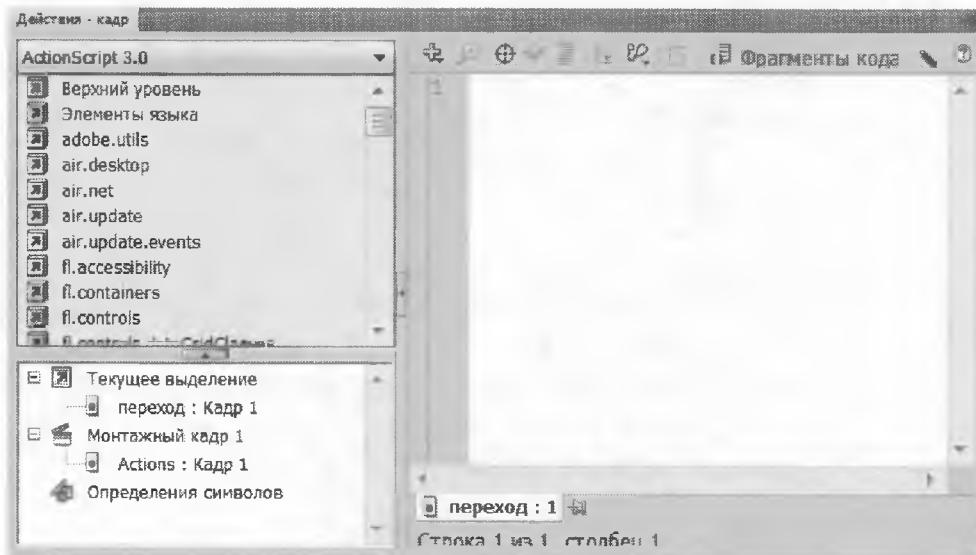


Рис. 7.43. Окно Действия, содержащее код ActionScript

Из пункта Действия данного окна выбираем фрагмент кода под ссылкой Нажмите для перехода к веб-странице. В результате предыдущее окно (см. рис. 7.43) получает текст кода (рис. 7.45).

При этом нужно внимательно читать пояснение к коду, которое закомментировано символами `/*` и `*/` и имеет серый цвет букв. Оно гласит:

```
/* Нажмите для перехода к веб-странице. При нажатии на указанный экземпляр символа производится загрузка веб-страницы в новом окне обозревателя.
```

Инструкции:

1. Замените `http://www.adobe.com` на адрес желаемой веб-страницы.

Не удаляйте кавычки `("")`.

`*/`

Соответственно, согласно инструкции изменяем в коде страницу `http://www.adobe.com` на необходимую нам страницу `index.html`. Если файл с роликом лежит в одном каталоге со страницей `index.html`, то исправленный код команды будет выглядеть следующим образом:

```
movieClip_4.addEventListener(MouseEvent.CLICK, fl_ClickToGoToWebPage_4);

function fl_ClickToGoToWebPage_4(event:MouseEvent):void
{
    navigateToURL(new URLRequest("index.html"), "_blank");
}
```

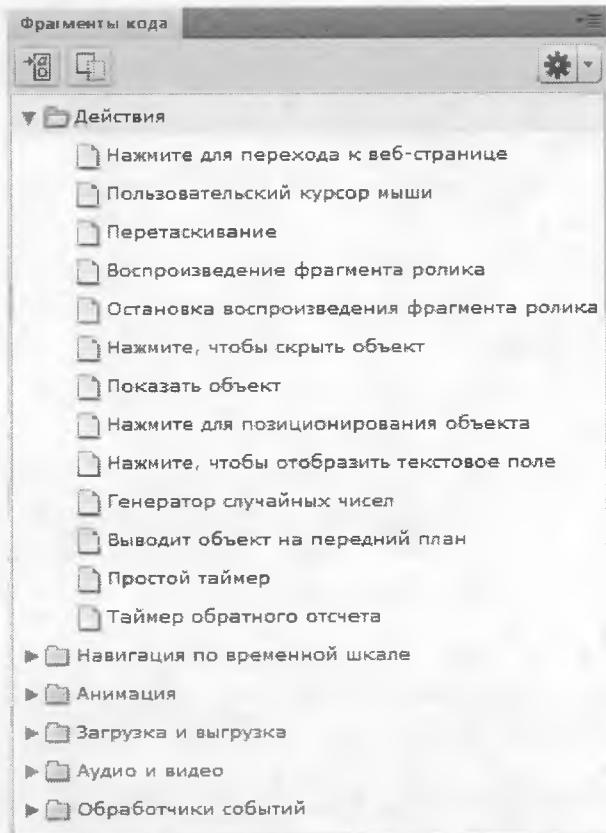


Рис. 7.44. Окно, содержащее готовые фрагменты кода

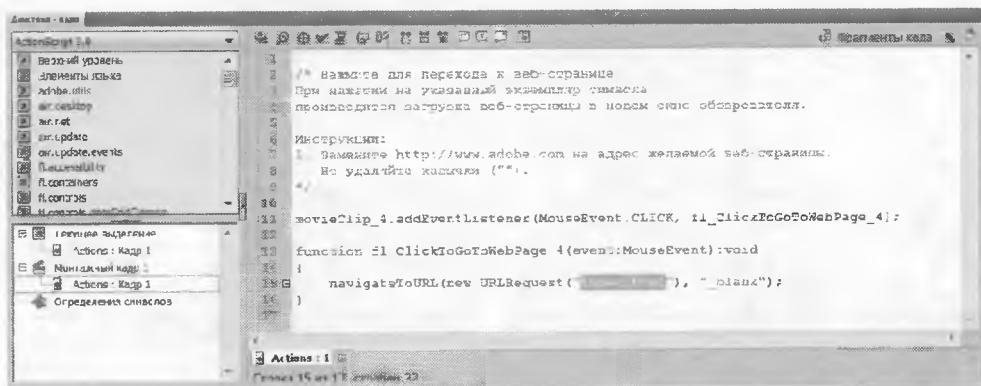


Рис. 7.45. Вставленный фрагмент кода в окне Действия

Если страница `index.html` находится в другой папке, то следует прописать относительный путь к ней от места расположения файла ролика.

Далее возьмем с помощью мыши слой переход и перетащим его на временной шкале таким образом, чтобы он оказался в самом низу, под всеми слоями (рис. 7.46).

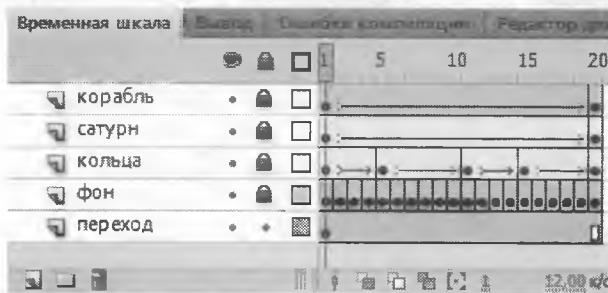


Рис. 7.46. Расположение слоев на временной шкале

Затем сохраним файл и перезапустим его не из программы, как мы обычно делаем, а из папки, в которой он хранится. При щелчке на области ролика загрузится главная страница сайта (рис. 7.47).



Рис. 7.47. Загрузка главной страницы сайта, инициированная щелчком мыши на области ролика. На переднем плане сам flash-ролик

Из описанных в этой главе примеров можно сделать простой вывод: создание элементарных flash-роликов не представляет особого труда, достаточно внимания, терпения и фантазии. Важно помнить, что для каждого отдельного анимируемого объекта необходимо создавать отдельный слой — это особенно критично при создании анимации формы. Если предложенного материала для вашей работы уже недостаточно, то, значит, пора осваивать узкоспециализированную литературу по Flash и языку ActionScript.

Глава 8

DREAMWEAVER — ГЛАВНЫЙ ИНСТРУМЕНТ ВЕБ-МАСТЕРА

8.1. Да здравствует прогресс в области сайтостроения!

Продолжим знакомство с некоторыми методами создания веб-страниц и сайтов. Во второй главе были рассмотрены способы написания веб-страниц в текстовом редакторе Блокнот с применением языка гипертекстовой разметки HTML. Четвертая глава посвящена знакомству с языком JavaScript, где мы создавали сценарии с помощью инструмента Notepad++. Из приведенных там сведений можно сделать вывод о том, что для написания веб-страницы достаточно простого текстового редактора. Позволю себе маленькую оговорку: при этом необходимо знать язык гипертекстовой разметки HTML на уровне его текущей спецификации, следить за его развитием и применять в деле все вводимые в язык новшества.

Если работа состоит только в создании и поддержке сайтов, то эти требования к знанию всей мощи языка оправданы и, как следствие, подкреплены должностным окладом. Но, как показывает практика, в небольших компаниях функционал создания и поддержки сайта считается дополнительной обязанностью специалиста, зачастую имеющего достаточно далекое представление обо всех тонкостях как языка HTML, так и сайтостроения в частности. Поскольку работа не волк, то есть выполнять ее необходимо, то самый простой выход — приобрести удобный специализированный инструмент, предназначенный для создания и редактирования веб-страниц. При его использовании можно свести к минимуму знания языка гипертекстовой разметки, опираясь только на общие положения.

Как говорилось ранее, для написания текста веб-страницы можно использовать текстовые, или HTML-, редакторы. В свою очередь, их можно разделить на два вида.

- **Кодовые редакторы** предлагают пользователю напрямую работать с HTML-кодом веб-страницы и при этом предоставляют дополнительные возможности: подсветку синтаксиса, проверку написания и т. д. Характерный пример кодового редактора — инструмент Notepad++, использованный нами при написании сценариев на языке JavaScript.

- **Визуальные редакторы**, в отличие от кодовых, предлагают пользователю различные инструменты для работы с объектами веб-страницы (текстом, таблицами, стилями и пр.) напрямую, минимизируя необходимые знания кода или приемов веб-построения. Эта особенность привлекает к визуальным редакторам максимальное количество нетехнических специалистов, так как позволяет при внесении изменений сразу же видеть результат работы. Характерными представителями визуальных редакторов являются программы Adobe Dreamweaver из пакета программ компании Adobe и проект с открытым кодом Kompozer, который можно скачать с сайта www.kompozer.net.

Следует отметить, что Adobe Dreamweaver сочетает в себе как кодовый, так и визуальный редакторы. При работе в ней веб-мастер может переключать режимы просмотра страницы в формате как есть или в режиме отображения кода, а также выводить на экран разбитое на две части окно, в одной части которого отображается код страницы, а в другой — ее вид. Безусловно, это делает данную программу одним из лучших инструментов как для начинающих веб-мастеров, так и для профессионалов. В свою очередь, проект с открытым кодом Kompozer позволяет получить практически тот же самый функционал, который предлагает Adobe, но совершенно бесплатно.

Поскольку книга построена на описании приемов построения веб-страниц с помощью программных продуктов компании Adobe, не будем отступать от правил и приступим к рассмотрению Adobe Dreamweaver. Триальную, или пробную, версию программы можно скачать с сайта компании Adobe, как скачивали программы Adobe Photoshop и Adobe Flash. Программа устанавливается стандартным для Windows способом. Если же у вас полный пакет программных продуктов, то при его установке проконтролируйте, выбрано ли приложение Adobe Dreamweaver в соответствующей строке.

Для удобства использования программы Adobe Dreamweaver необходимо сделать глобальные настройки, причем сразу же после ее установки. Дело в том, что при создании веб-страницы программа автоматически генерирует код, определяющий параметры ее просмотра. Это код символов, стиль страницы и пр. Но поговорим обо всем по порядку.

Первый запуск Adobe Dreamweaver. Начальные настройки

В процессе установки при первоначальном запуске программы откроется окно, где придется указать расширения для типов файлов, которые следует ассоциировать с данной программой. В этом случае имеет смысл оставить все предлагаемые варианты, если у вас нет дополнительных программ для работы именно с этим конкретным типом файла, либо можно сохранить установки по умолчанию.

При первом запуске программы, как и во всех продуктах Adobe, откроется приветственное окно, позволяющее выбрать действие — открыть или создать файл на основе уже существующих шаблонов или с пустых

страниц (рис. 8.1). Установив флажок **Больше не показывать**, можно отключить автоматическое открытие этого окна при последующих запусках программы.

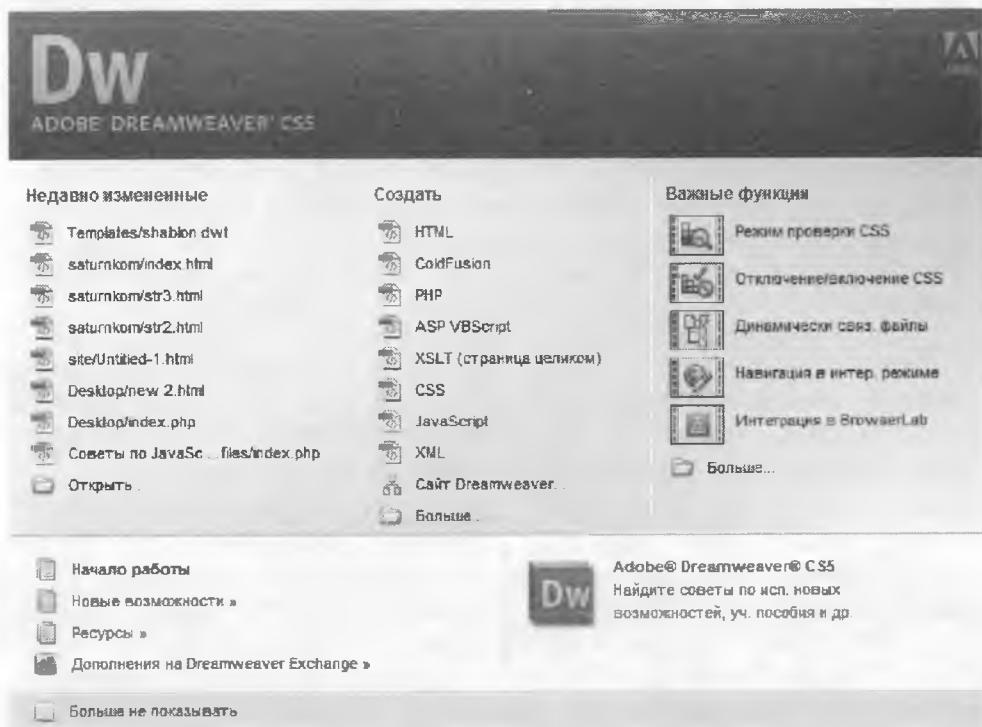


Рис. 8.1. Окно приветствия программы Adobe Dreamweaver

Напоминаю, что описываемые в этой книге программные продукты могут по версии отличаться от тех, что установлены на вашем компьютере, и иметь несколько другой интерфейс или локализацию. Однако вне зависимости от версии принципы действий при выполнении операций полностью аналогичны.

Выполним команду **Создать ▶ HTML** и перейдем к следующим глобальным настройкам среды.

Выбрав пункт меню **Правка ▶ Настройки**, откроем окно настроек системы (рис. 8.2).

Внимательно просмотрите все пункты, подберите и установите для себя настройки, которые посчитаете нужными. Если не вполне понятно, о чем, собственно, идет речь в окне, оставляйте настройки по умолчанию во избежание дальнейших проблем. Однако есть настройки, на которые следует обратить внимание.

- На вкладке Исправление кода нужно установить флажок Исправлять незакрытые теги и недопустимую вложенность тегов.
- На вкладке Просмотр в браузере следует щелкнуть на символе + и добавить те браузеры (помимо Internet Explorer), которые установлены в вашей операционной системе и с помощью которых вы будете просматривать при тестировании создаваемые веб-страницы (рис. 8.3).

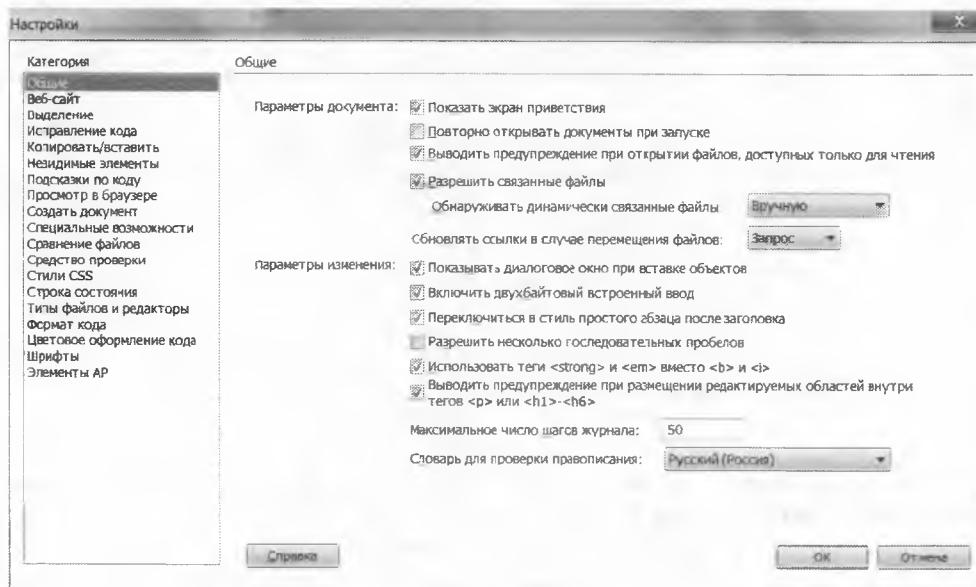


Рис. 8.2. Окно настроек системы

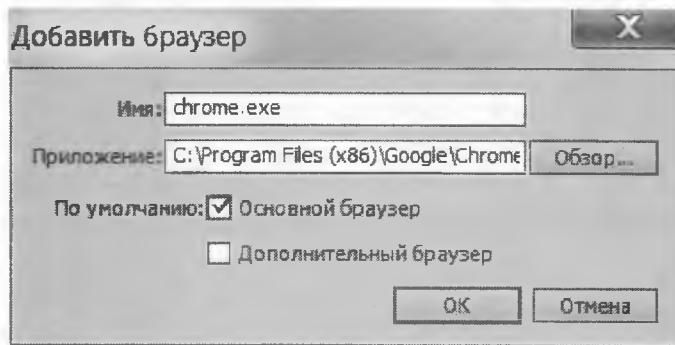


Рис. 8.3. Окно добавления браузера, необходимого при просмотре и тестировании веб-страниц

- В результате вкладка Просмотр в браузере примет вид, показанный на рис. 8.4.

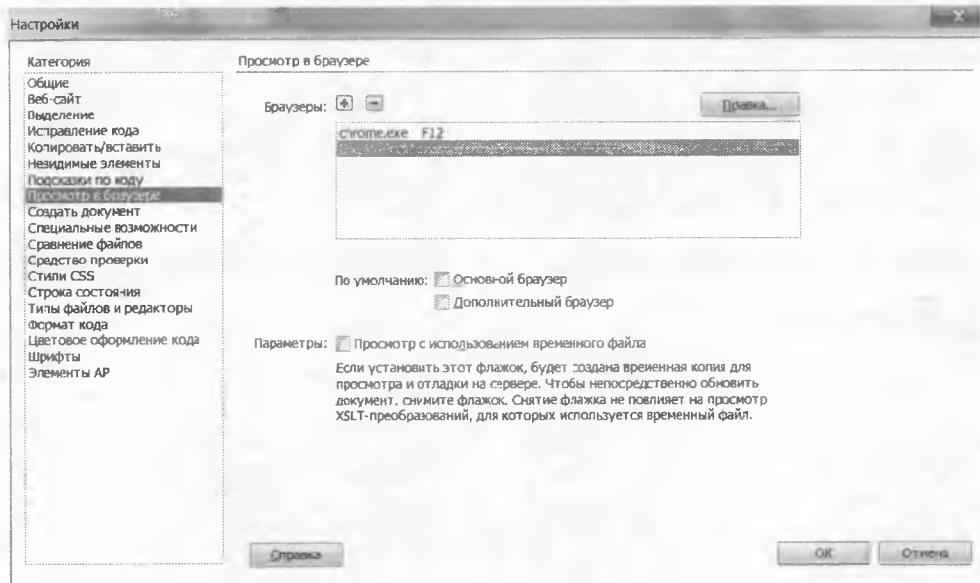


Рис. 8.4. Вид вкладки Просмотр в браузере после добавления дополнительного браузера

- На вкладке Создать документ нужно обязательно установить в окне Кодировка по умолчанию значение Кириллица Windows. Дело в том, что при создании страницы Adobe Dreamweaver генерирует метатег, содержащий информацию о коде символов, которыми написана страница, например, как на рис. 8.5.

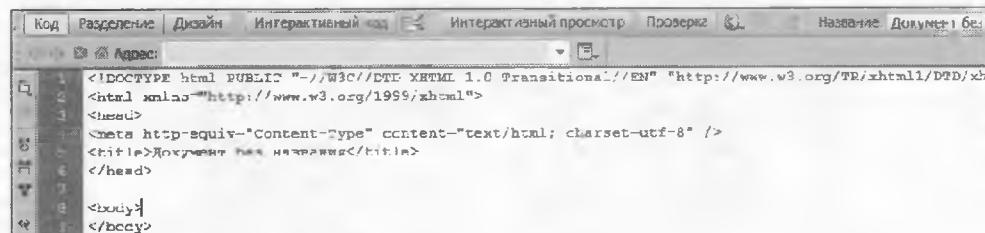


Рис. 8.5. Фрагмент кода страницы, сгенерированной Adobe Dreamweaver

Надпись `<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />` как раз и определяет стиль и кодовую таблицу для шрифтов страницы. Таким образом, изменяя значение поля Кодировка по умолчанию на значение Кириллица Windows, мы определяем генерируемую строку метатега как `<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />`, что, в свою очередь, позволяет браузеру не изменять настройку кодировки при про-

смотре веб-страниц. Если не сделать этого, то есть явно не указать код символов, то ничего страшного не произойдет. Однако некоторые браузеры, в частности Internet Explorer, не всегда корректно распознают кодировку символов и вместо них выводят на экран так называемые кро-козябры — сленговое название символов псевдографики. Хотя кодировка UTF-8 давно и успешно существует по всему миру, облегчая понимание различных шрифтов и, соответственно, языков, у компании Microsoft, как всегда, свое видение удобств коммуникации. По этой причине операционные системы от Microsoft и соответствующие сетевые инструменты настроены на использование именно корпоративных кодовых таблиц, в частности windows-1251. Это, в свою очередь, доставляет множество неудобств конкретным пользователям.

Итак, настройки выполнены, перейдем к работе со страницами.

Рабочая среда программы Adobe Dreamweaver

В общем и целом рабочая среда Adobe Dreamweaver показана на рис. 8.6.

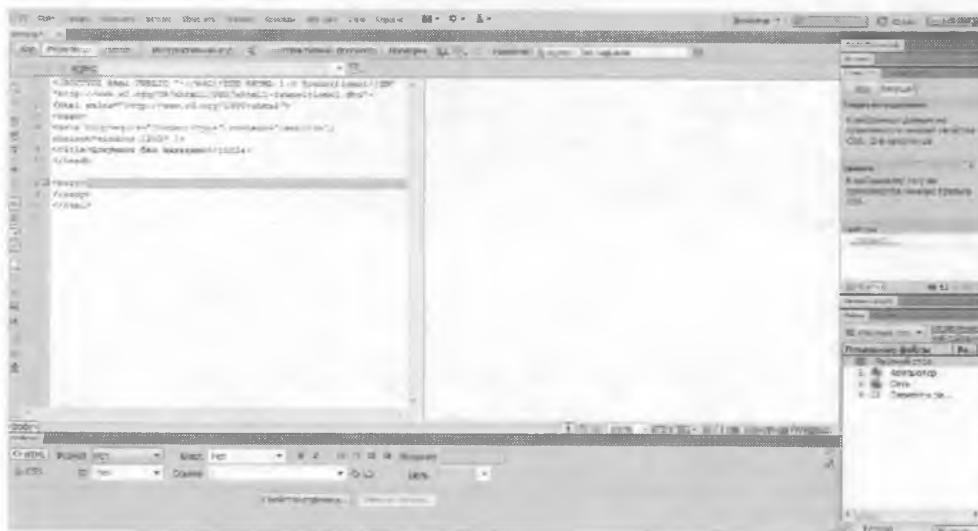


Рис. 8.6. Среда программы Adobe Dreamweaver

Меню для всех инструментов пакета от Adobe стандартное, и принцип его работы аналогичен описанному в предыдущих главах для Adobe Photoshop и Adobe Flash.

Рабочая область программы ограничена снизу окном свойств объекта, справа по умолчанию находятся дополнительные инструментальные окна, включить или отключить которые можно через пункт меню Окно. Область веб-страницы, показанная на рис. 8.6, разделена на две части: слева находится код страницы, а справа — ее дизайн. Такое разделение удобно, если у вас

широкоэкранный монитор. Кроме того, хорошо работать в такой области и при отладке страницы, так как при выделении редактируемого объекта в соседнем окне сразу выделяется и его код, что позволяет находить ошибки, имеющиеся на странице (рис. 8.7).

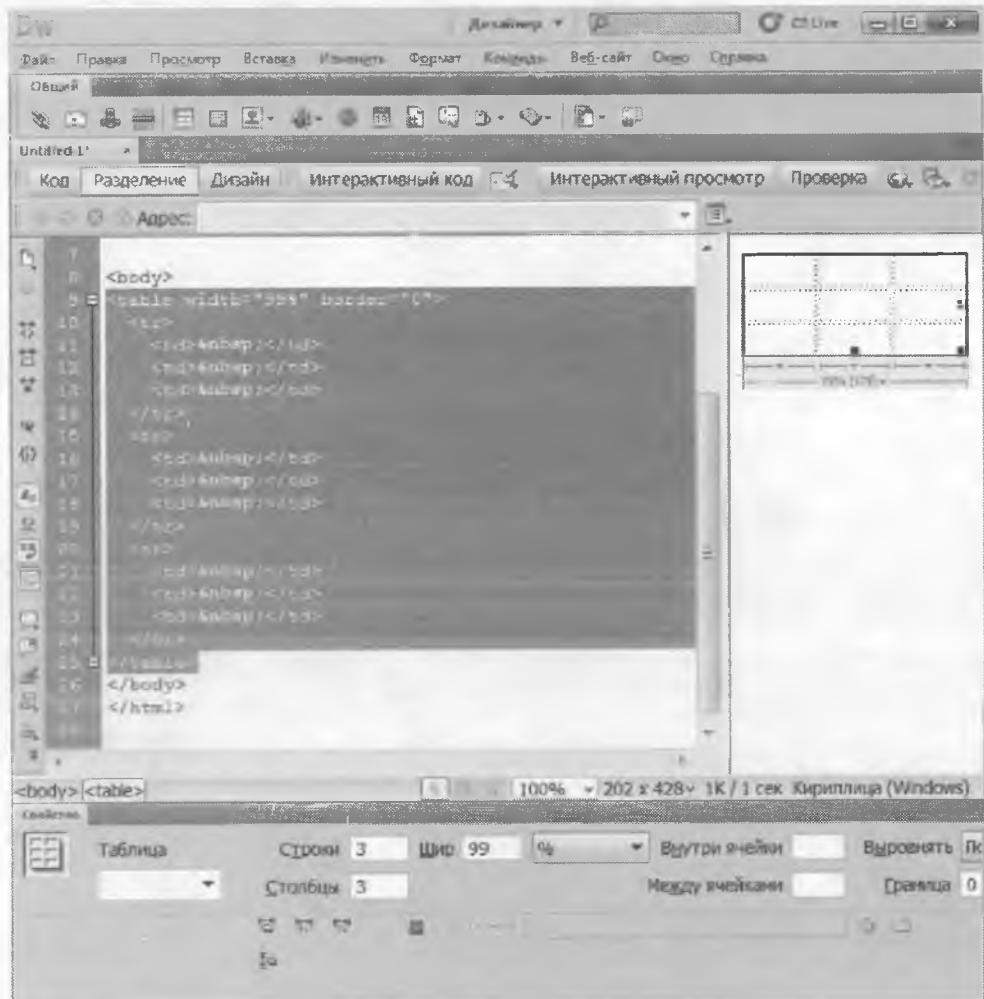


Рис. 8.7. Пример выделения кода объекта при редактировании дизайна объекта

Обычно по умолчанию в верхней части рабочей области находится панель инструментов Элементы (рис. 8.8). Если это не так, воспользуйтесь пунктом меню Окно ▶ Элементы. При необходимости эту панель, как и все другие, можно установить в любую удобную часть экрана. Панель Элементы позволяет вставлять на страницу различные объекты, не вдаваясь в написание их кода и не обращаясь к соответствующим пунктам меню.

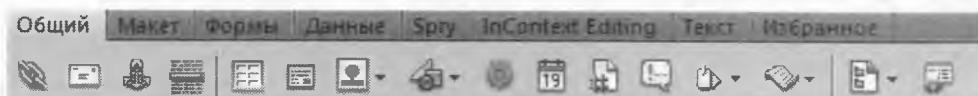


Рис. 8.8. Панель инструментов Элементы

Внизу под рабочей областью располагается панель свойств (рис. 8.9).

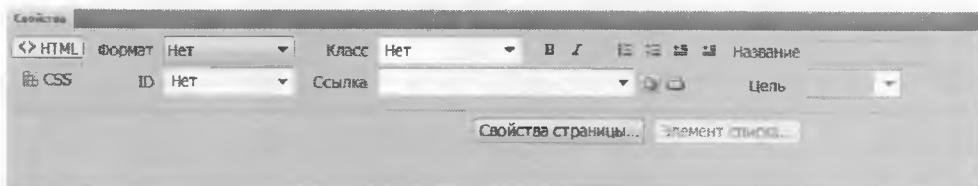


Рис. 8.9. Панель Свойства

На панели свойств в виде диалоговых окон и раскрывающихся меню отображаются те свойства активных в данный момент объектов, которые можно изменить, при этом автоматически изменяются параметры тегов в описании кода объекта.

Таким образом, настройка среды зависит от разрешения и ширины монитора, а также от удобства расположения тех или иных панелей инструментов под руками.

8.2. Сайт не на коленке

Создание сайта

Программа Adobe Dreamweaver позволяет создавать сайт с одновременным размещением в Интернете либо с локальным сохранением страниц на рабочем компьютере и последующим их размещением на веб-сервере с помощью протокола FTP. И в том и в другом случае необходимо выполнить первоначальную настройку сайта.

Выполнив команду **Сайт ▶ Новый сайт**, мы откроем окно создания нового сайта, диалоговые элементы которого и будем заполнять (рис. 8.10).

В соответствующих полях вкладки **Сайт** необходимо указать имя сайта — в нашем случае **saturnkom** — и его расположение. Обращаю ваше внимание, что по умолчанию создаваемые сайты сохраняются в папке **Мои документы** для локального профиля пользователя, в отличие от предложенного на рис. 8.10, где указана папка на диске D:.

На вкладке **Серверы** укажите данные размещения сайта в Интернете. Они полностью аналогичны данным, которые мы получили, когда размещали созданные нами веб-страницы в Сети с помощью FTP-клиента (см. главу 3).

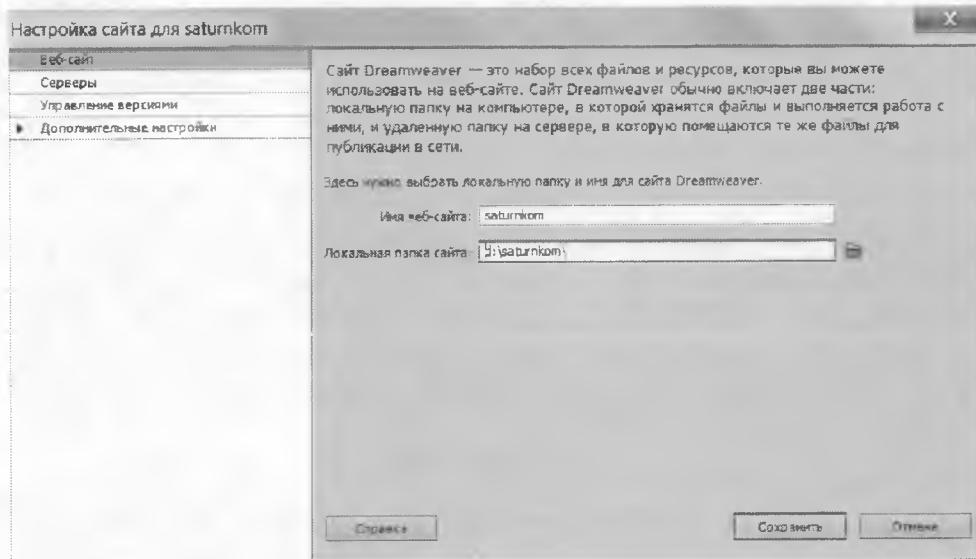


Рис. 8.10. Окно настройки нового сайта

- Доступ к FTP-серверу — `ftp.narod.ru`. Этот хост действителен для сайтов, созданных до 2010 года. Современный — на момент написания главы — хост будет выглядеть так: `saturnkom.ftp.narod.ru`. Как видите, появилось требование установки логина перед хостом.
- Порт, по которому осуществляется подключение, — 21.
- Тип установленного сервера — FTP.
- Имя (логин) пользователя берем из регистрационных данных — `saturnkom`.
- Паспорт (пароль доступа к сайту) пользователя тоже знаем — `*****`.

Таким образом, заполненная вкладка Серверы примет вид, как на рис. 8.11.

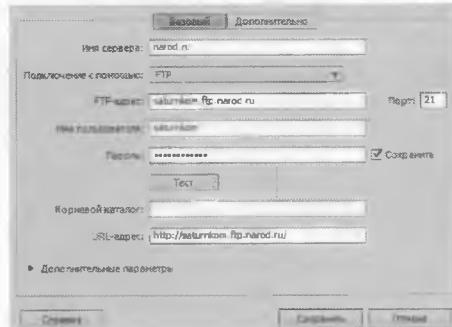


Рис. 8.11. Заполненная вкладка Серверы

При этом не забудьте подключиться к Сети и нажать кнопку Тест, чтобы проверить правильность подключения.

Далее перейдем на вкладку Дополнительные настройки (рис. 8.12).

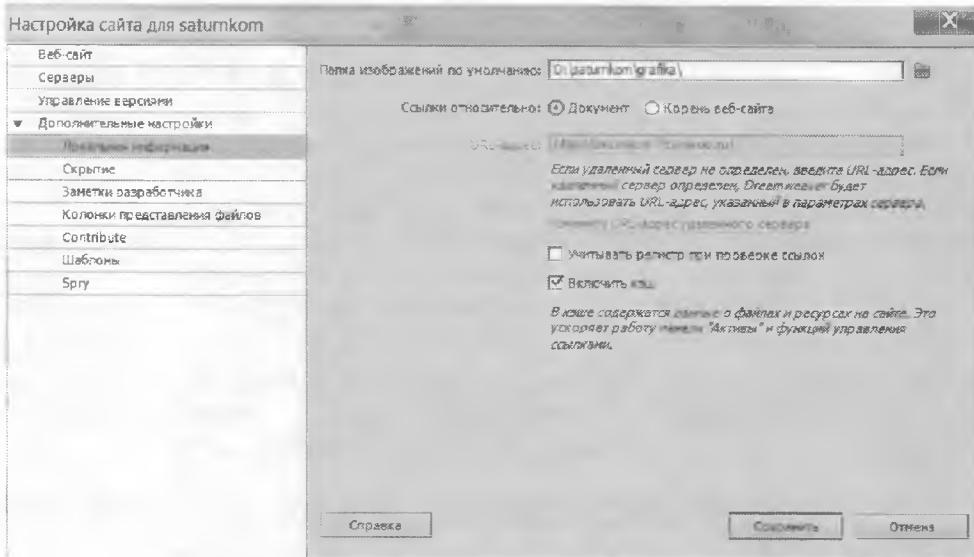


Рис. 8.12. Вкладка Дополнительные настройки

Укажем папку для хранения изображений и расположение ссылок в создаваемом сайте. Нажмем кнопку Сохранить и будем считать, что сайт нами создан, поэтому пора переходить к созданию веб-страниц, непосредственно несущих информацию.

Для контроля за созданным сайтом можно обращаться к окну Файлы (рис. 8.13).

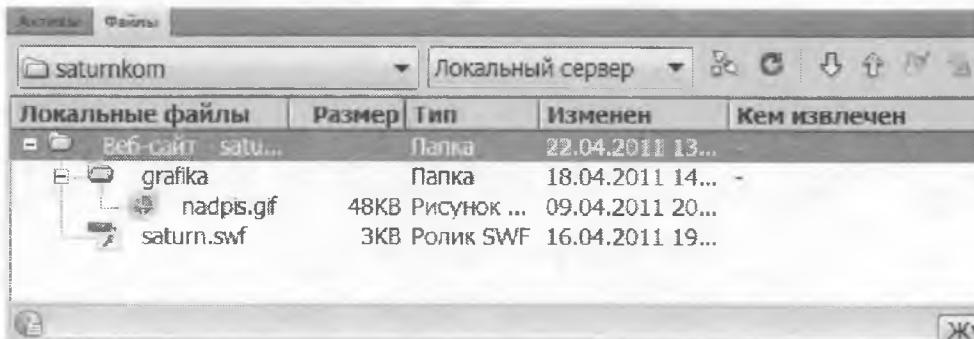


Рис. 8.13. Окно Файлы, содержащее информацию о файлах создаваемого сайта

Первая страница сайта как шаблон для последующих страниц

При создании веб-страниц можно воспользоваться готовыми шаблонами, которые находятся в библиотеке программы. Они имеют уже установленную разметку страницы и ее стилевое оформление.

Но мы не ищем легких путей и будем создавать свои страницы на основе резиновой таблицы, причем в той же последовательности, что и в главе 2. При этом дополнительная задача помимо знакомства с программой Adobe Dreamweaver будет состоять в том, чтобы сравнить трудозатраты на создание одинаковых сайтов и их размещение в Интернете. Поэтому начнем построение нашего сайта со страницы `index.html`, а чтобы в дальнейшем не возникало сложностей при создании следующих веб-страниц, оформим ее со всей возможной тщательностью, так как она будет шаблоном для остальных страниц сайта.

Итак, выберем пункт меню **Файл > Создать** и в появившемся окне зададим такие настройки: **Документ — Пустая страница, Тип страницы — HTML, Макет — <нет>**, а затем подтвердим создание нажатием кнопки **Создать** (рис. 8.14).

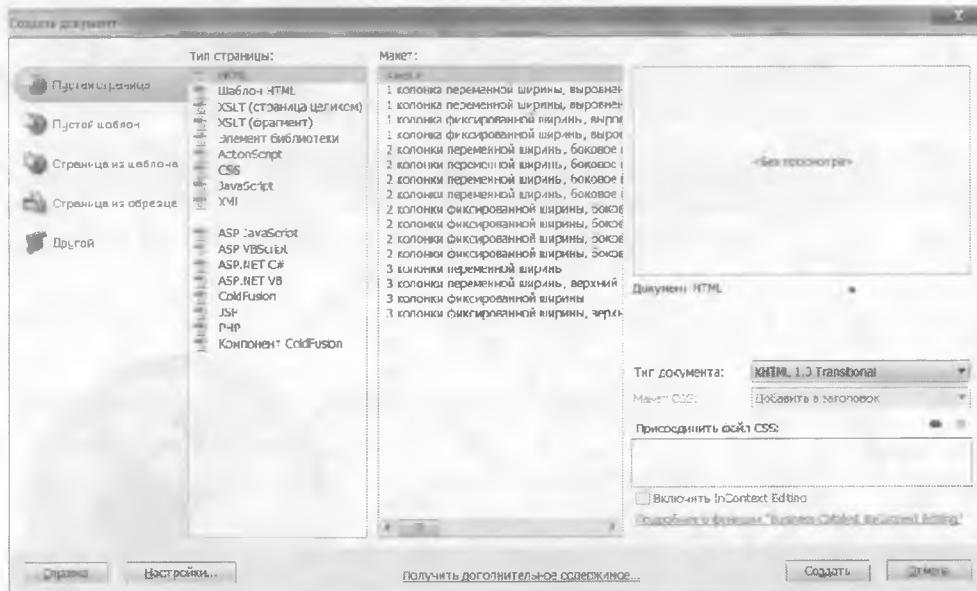


Рис. 8.14. Окно создания документа

Появившуюся страницу сразу сохраним под именем `index.html`, для чего выберем пункт меню **Файл > Сохранить как**. В открывшемся окне укажем имя для нашего шаблона (рис. 8.15).

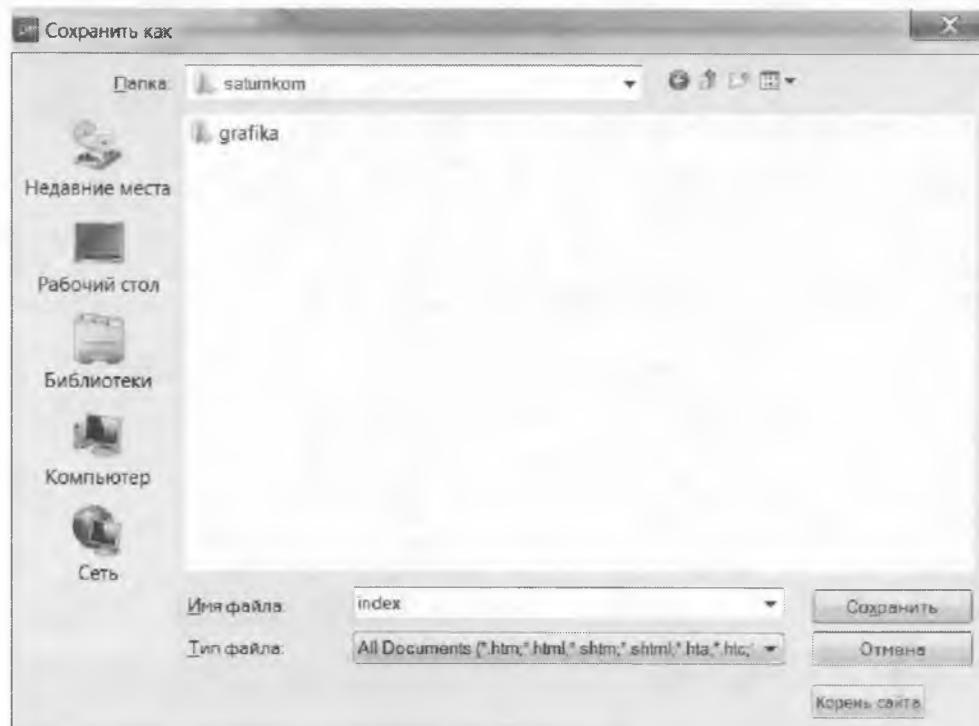


Рис. 8.15. Диалоговое окно сохранения файла в качестве шаблона

Определение свойств страниц

Теперь приступим к оформлению страницы.

Сначала перейдем в свойства страницы и зададим для нее значение фона цвета, для чего обратимся к панели свойств, расположенной внизу рабочей области, или вызовем контекстное меню страницы. В результате откроется диалоговое окно, представленное на рис. 8.16.

Как видно из рисунка, указывать цвет можно, непосредственно прописав его шестнадцатеричный код либо выбрав вариант из палитры встроенной библиотеки. Для этого необходимо щелкнуть кнопкой мыши на фигуре, имеющей в изображении темный треугольник, направленный вершиной вниз. При указании в качестве фона какого-либо графического файла нужно первоначально поместить его в папку графических объектов на сайте, а затем указать местоположение через окно, открывающееся нажатием кнопки Обзор. При этом следует помнить о том, что у графических файлов жестко заданные размеры, и поэтому фон может либо вылезать за пределы веб-страницы, либо, что недопустимо, быть фоном только для ее куска. Но если использовать маленькое изображение для фона, то можно замостить

веб-страницу картинками, указав метод повтора в окне Повторить, как мы делали это в главе 2.

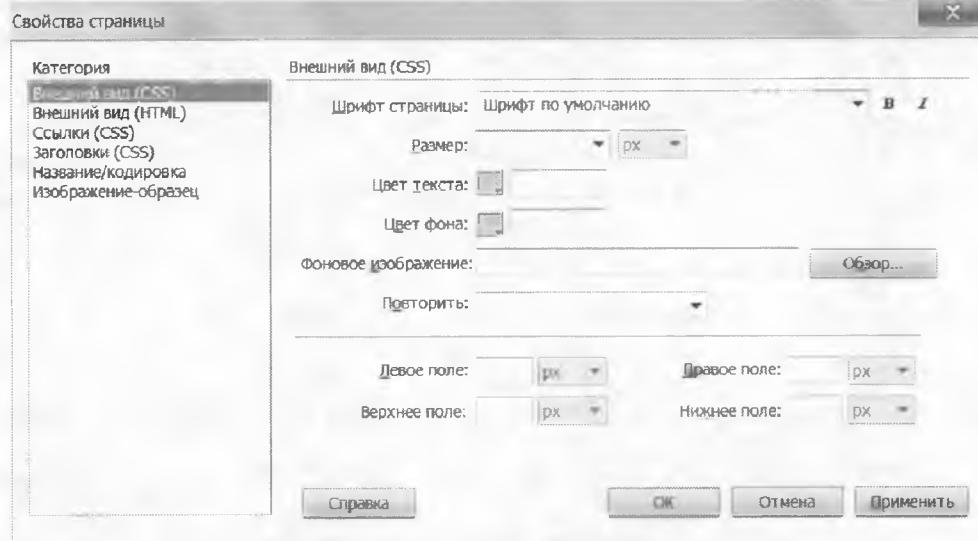


Рис. 8.16. Диалоговое окно Свойства страницы

Задав цвет фона и стили шрифта, можно увидеть изменения в коде страницы, если работать в режиме Разделение (рис. 8.17).

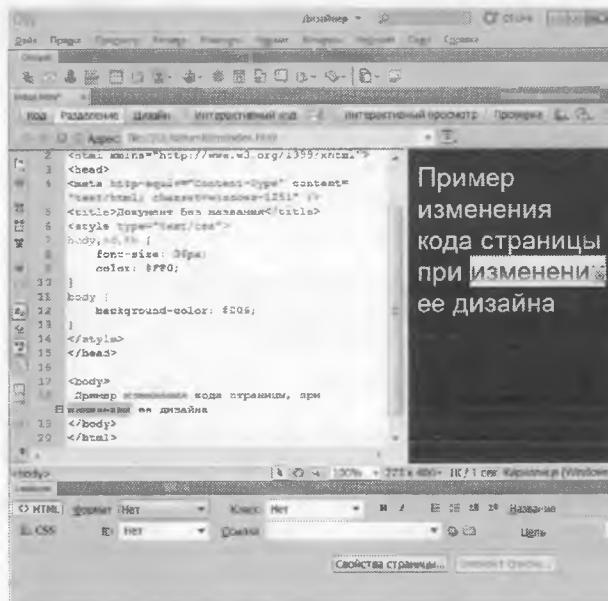


Рис. 8.17. Демонстрация изменения кода страницы

8.3. Таблицы и графика

Вставка и настройка таблиц

Напомню, что в главе 2 мы создавали горизонтальную таблицу для размещения пунктов меню. Пора от этого отказаться по той простой причине, что количество страниц в сайте будет неуклонно расти, а расширять ширину окна до бесконечности невозможно. Зато увеличение количества строк, содержащих пункты меню, ограничено только количеством информации на сайте.

Итак, планирование информации на веб-странице приблизительно будет выглядеть таким образом.

- *Вверху* графический заголовок, содержащий подготовленные ранее графические файлы.
- *Слева* построчно расположенные пункты меню, являющиеся гиперссылками на соответствующие страницы сайта.
- *Справа* область, содержащая информацию страницы.

Распределяем ячейки таблицы, в которых все это будет содержаться: столбцов — 4, строк для начала — 5. Для вставки таблицы можно выполнить команду **Вставка** ▶ **Таблица** либо воспользоваться инструментом **Таблица** на панели инструментов **Элементы**. При этом откроется диалоговое окно установки параметров таблицы (рис. 8.18).

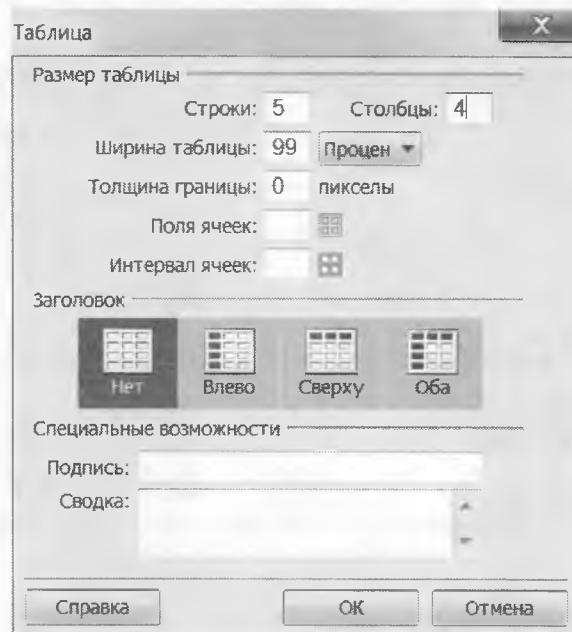


Рис. 8.18. Диалоговое окно вставки таблиц на страницу

Обратите внимание на следующие значения.

- Ширина таблицы – 99 % от ширины экрана. Это резиновая таблица. Почему не 100 %? Нужно, чтобы оставалась эстетическая рамка, обозначающая границы страницы.
- Толщина границы – 0 пикселов, чтобы граница таблицы была невидимой и не отображалась на экране компьютера.

Подтверждаем нажатием кнопки **OK** и получаем готовую таблицу с прозрачными границами (рис. 8.19). Осталось только в свойствах таблицы указать значение параметра **Выровнять**, вместо **По умолчанию** поставив **По центру**.

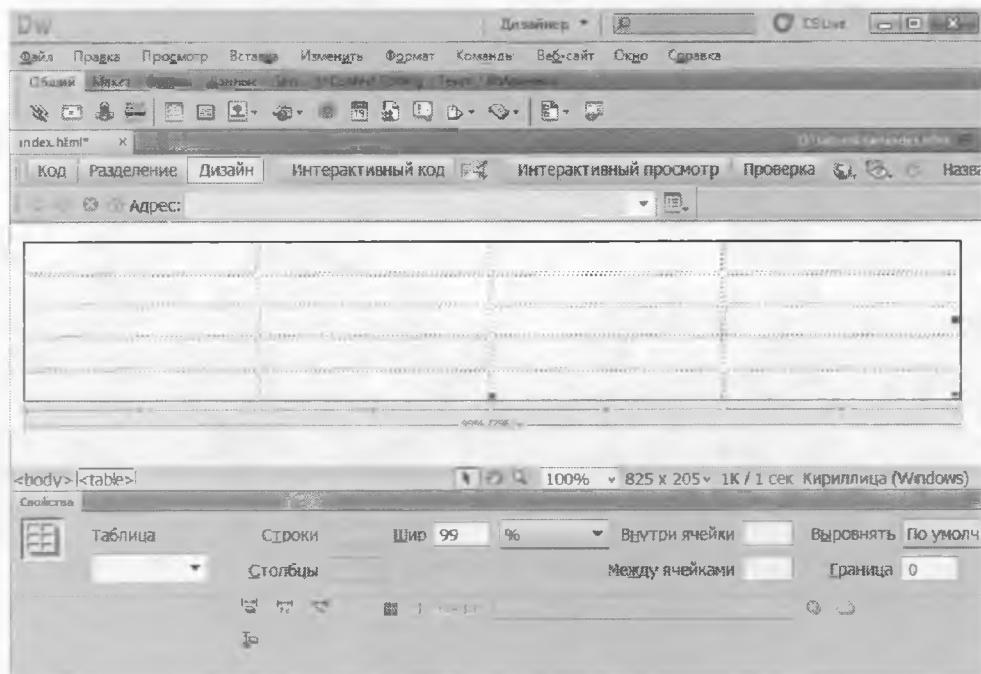


Рис. 8.19. Встроенная в веб-страницу таблица 4 × 5 с прозрачными границами

Добавление графических изображений на веб-страницу

Подготовимся к оформлению графического заголовка. Созданные в главах 5 и 6 графические файлы необходимо перенести в корневой каталог нашего сайта и разместить в соответствующих папках. Файл *nadpis.gif*, который содержит анимированный заголовок к нашему сайту, расположим в папке *grafika*, а файл с роликом *saturn.swf* придется оставить в корневом каталоге ресурса, так как при его записи мы делали гиперссылку на страницу *index.html* без непосредственного указания пути.

Для вставки графического объекта необходимо сначала установить в нужную ячейку курсор таким образом, чтобы он там начал мигать. После этого следует выбрать пункт меню **Вставка** ▶ **Изображение** или щелкнуть на соответствующем значке на панели инструментов **Элементы**. В результате выполненных действий откроется окно вставки графического объекта (рис. 8.20).

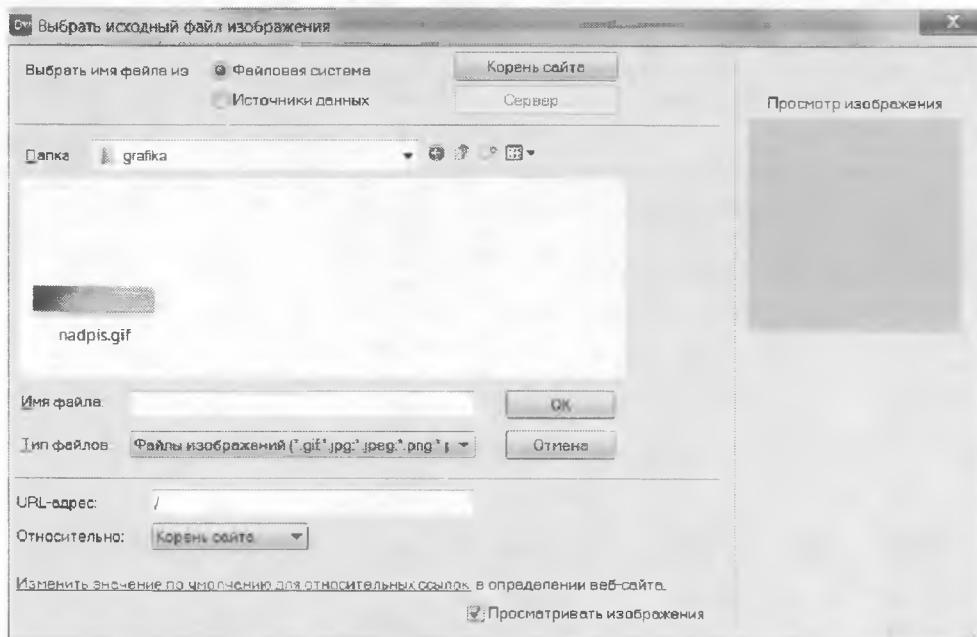


Рис. 8.20. Окно вставки графического объекта

После указания имени файла и пути к нему нажмем кнопку **OK**. Окно вставки закроется, и откроется новое окно, в котором можно задать дополнительные атрибуты тега **Image** (рис. 8.21).

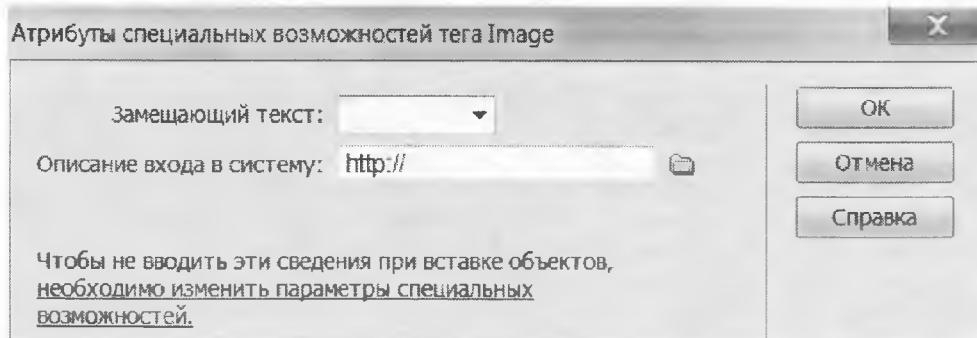


Рис. 8.21. Окно установки атрибутов тега **Image**

В появившемся окне в поле **Замещающий текст** необходимо прописать текст, который будет появляться в том случае, если по каким-либо причинам изображение не загрузится. В нашем случае надпись будет такая: **Компания «Сатурн», заголовок**. Применим изменения параметров тега нажатием кнопки **OK**.

Результат вставки графического файла можно увидеть на рис. 8.22.

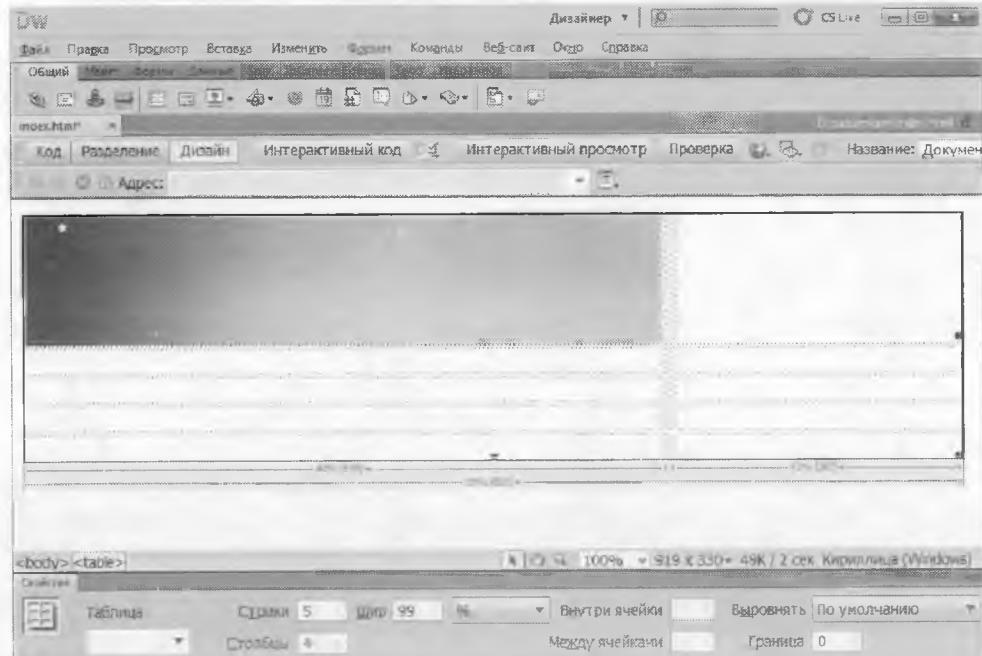


Рис. 8.22. Вставленный в ячейку таблицы графический объект

Для проверки того, как выглядит и действует в самой странице вставленный графический объект, достаточно щелкнуть кнопкой мыши на значке с изображением глобуса (рис. 8.23) и выбрать из списка браузер, в котором вы желаете просмотреть страницу. При этом не забывайте сохранять страницу каждый раз, когда программа попросит об этом, или перед закрытием документа.



Рис. 8.23. Выбор браузера для просмотра готовой страницы из предложенного списка

В результате веб-страница должна выглядеть следующим образом (рис. 8.24).



Рис. 8.24. Вид веб-страницы со вставленным графическим объектом

Вставка роликов Flash-анимации на веб-страницу

Для вставки ролика, содержащегося в файле `saturn.swf`, последовательность действий практически аналогична. Только нужно выполнить команду **Вставка** ▶ **Мультимедиа** ▶ **SWF** и вставить объект в крайний правый столбец верхней строки таблицы. В окне установки параметров атрибутов тегов (рис. 8.25) в поле названия пишем **Логотип**.

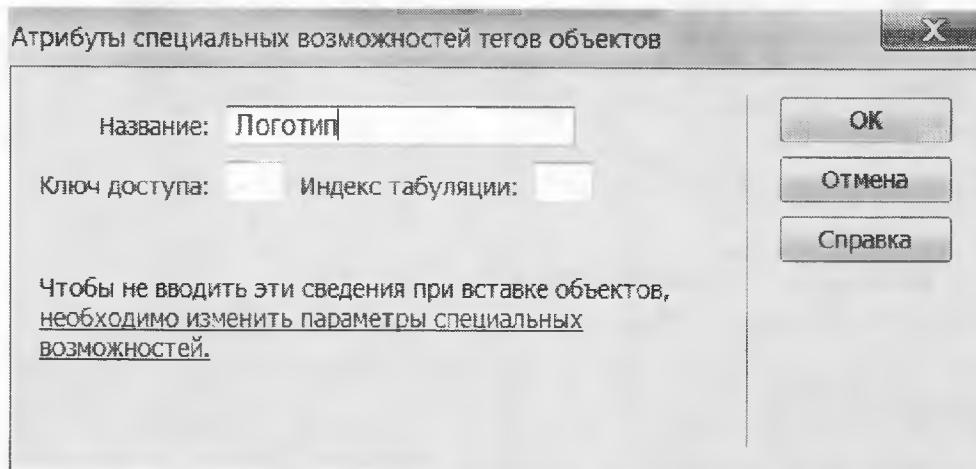


Рис. 8.25. Окно установки атрибутов тегов, появляющееся при вставке файлов типа SWF

Таким образом, дизайн страницы будет иметь вид, как на рис. 8.26, а сама страница будет выглядеть, как на рис. 8.27.

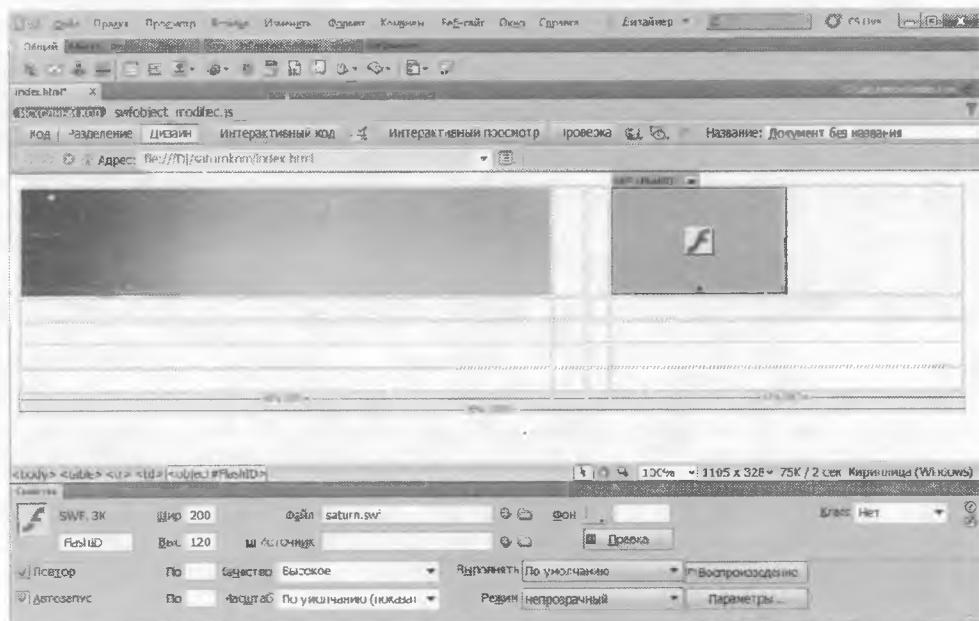


Рис. 8.26. Дизайн страницы



Рис. 8.27. Вид страницы с двумя встроеннымми графическими объектами

Объединение ячеек таблицы

Теперь перейдем к ячейкам нашей таблицы. В первой строке находится четыре ячейки: левая содержит заголовок, правая хранит логотип, а две средних ячейки — это непозволительная роскошь, ведь для создания «резиновости» таблицы достаточно одной ячейки. Следовательно, объединим две левые ячейки в одну. Почему именно две левых, а не две средних? Ширина объекта в левой ячейке — 400 пикселов, если вы помните, а под первой строкой находятся другие строки, левые ячейки которых мы хотели задействовать под пункты меню сайта, поэтому проще объединить две левые ячейки первой строки, чем впоследствии растягивать лежащие под ними столбцы.

Итак, выделим две левые ячейки первой строки и выберем пункт меню Изменить ▶ Таблица ▶ Объединить ячейки. Можно также воспользоваться контекстным меню выделенных ячеек. Теперь, когда ячейки объединены, зададим размеры для правой и левой ячеек, поймав мышью границы столбцов. Затем зададим выравнивание для содержимого правой и левой ячеек, спустившись к панели свойств. В результате дизайн таблицы должен выглядеть, как на рис. 8.28.

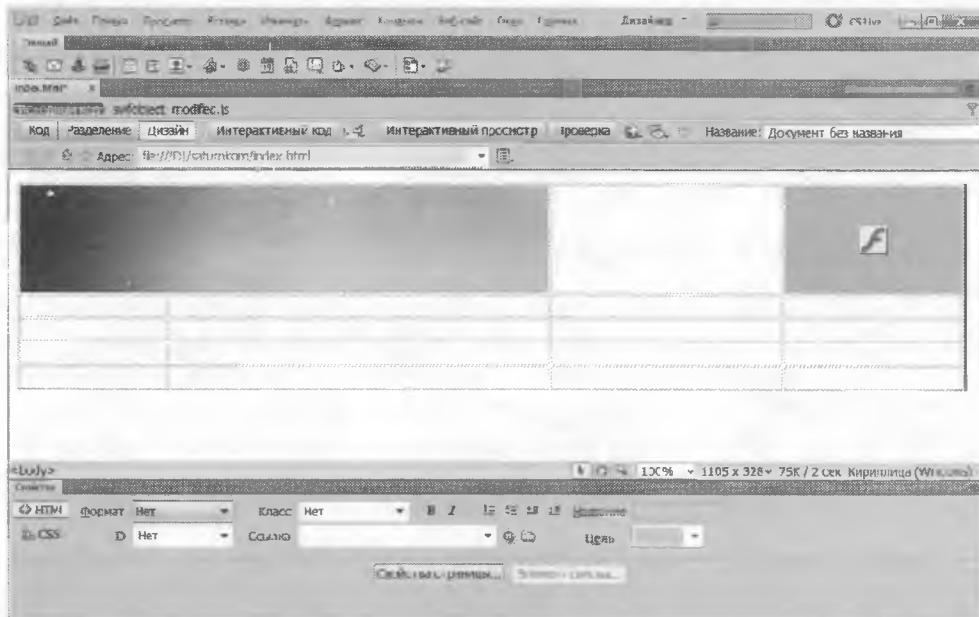


Рис. 8.28. Дизайн таблицы после объединения и выравнивания ячеек, содержащих графические объекты

Теперь, если просмотреть страницу в действии, то вне зависимости от размеров окна браузера графические объекты четко сохраняют свое положение относительно краев веб-страницы (рис. 8.29 и 8.30).



Рис. 8.29. Расположение графических объектов на странице при широком окне браузера

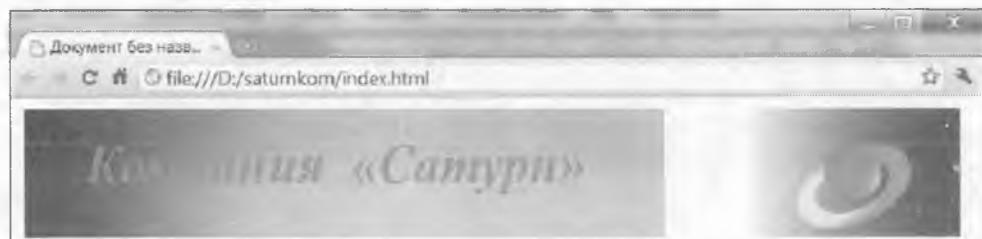


Рис. 8.30. Расположение графических объектов на странице при узком окне браузера

8.4. Форматирование текста. Организация гиперссылок

Работа с текстом на веб-страницах

Пора начинать работать с текстом. Нужно вставить на страницу нашу фирменную надпись: «Компания „Сатурн“ предоставит вам возможность провести отпуск или выходные дни на любой планете ближнего и дальнего освоенного космоса».

Для этого объединим все ячейки второй строки таблицы в одну ячейку описанным выше способом и внесем в нее необходимую надпись. Затем перейдем на панель Свойства, где можно задать стиль для надписи (рис. 8.31).

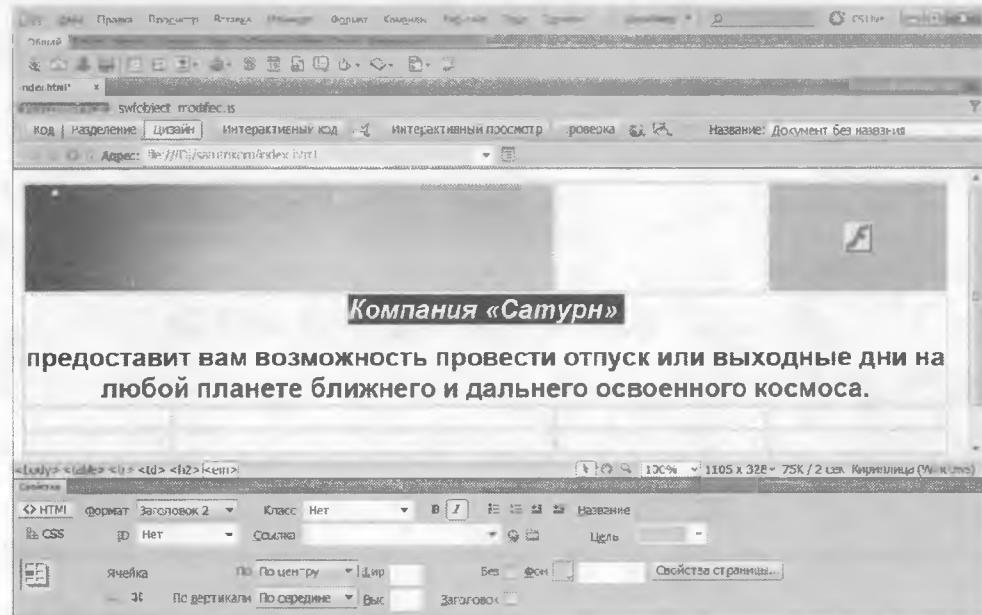


Рис. 8.31. Определение параметров надписи

Если нажать кнопку **CSS** вместо выбранной на рисунке кнопки **<>HTML**, то можно задать стиль выделенной надписи, который будет записан в отдельный файл стилей. Результат работы показан на рис. 8.32.

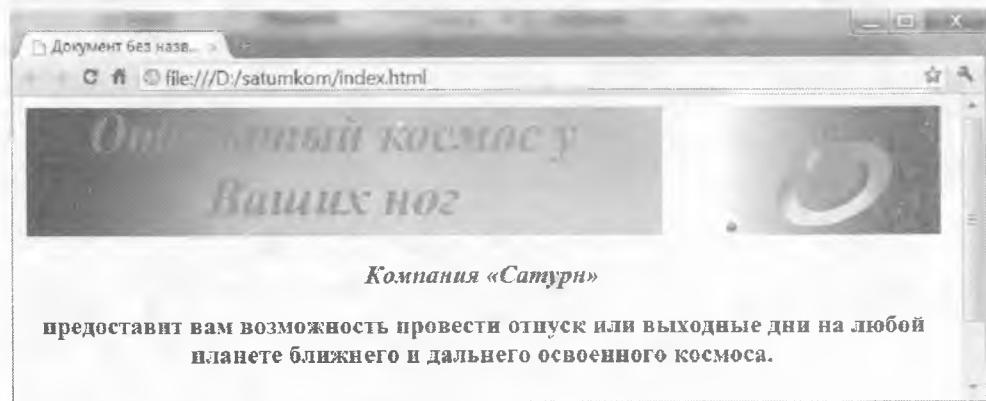


Рис. 8.32. Вставка надписи на веб-страницу

Таким же образом, как вставляли текст, можно вставить надписи пунктов меню. Поскольку у нас осталось еще три строки, то в левые ячейки этих строк и пропишем пункты меню: Главная страница, Услуги компании, Вопросы и ответы (рис. 8.33).

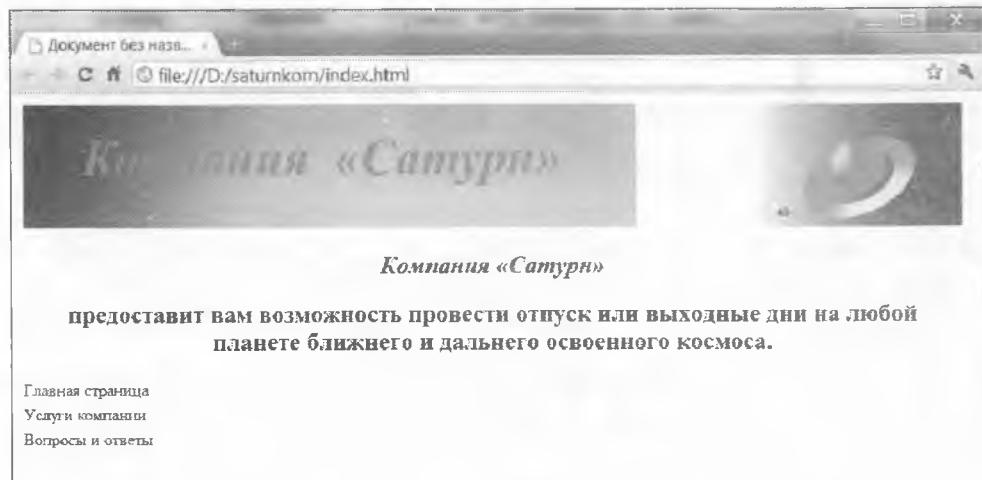


Рис. 8.33. Страница с пунктами меню

Для добавления новых строк в таблицу выполняется простая операция: выделяется необходимая строка или столбец, вызывается контекстное меню выделенного объекта и в нем выбирается пункт **Таблица > Вставить строки** (рис. 8.34).

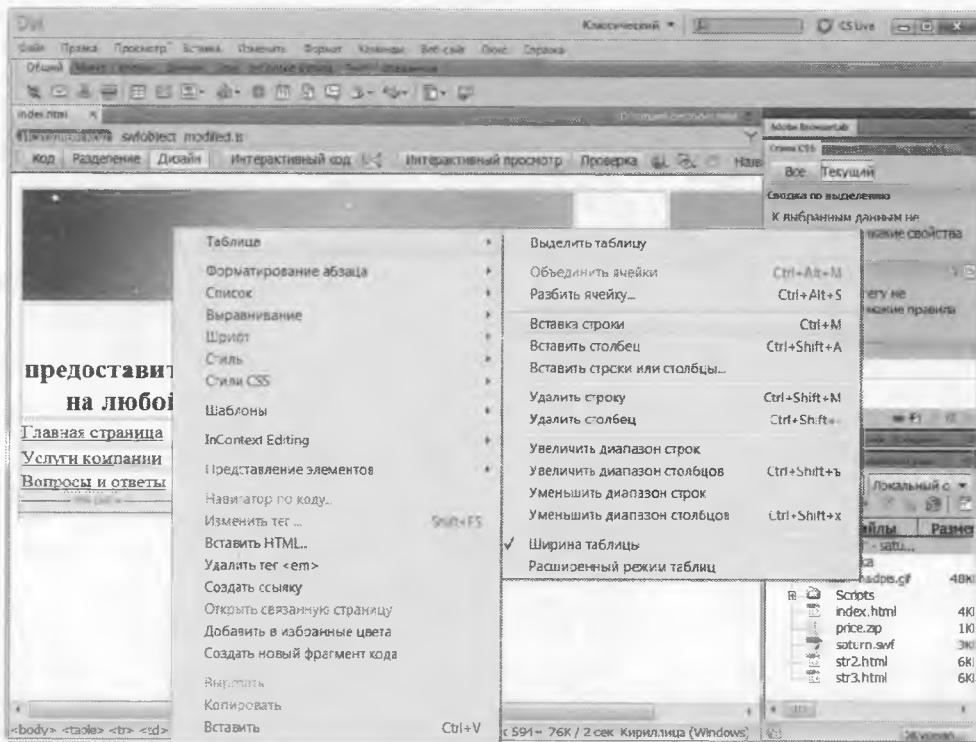


Рис. 8.34. Иллюстрация контекстного меню для работы со строками и столбцами таблицы

Шаблон на основе веб-страницы с именем `index.html` почти создан, осталось связать пункты меню гиперссылками со страницами, на которые они ссылаются. А для этого прежде всего необходимо создать сами страницы.

Сначала следует сохранить нашу страницу, для чего выберем пункт меню **Файл** \rightarrow **Сохранить**. Когда мы будем работать сразу с несколькими открытыми страницами, проще выполнить команду **Файл** \rightarrow **Сохранить все**, но только после того, как все страницы первоначально были сохранены под своими именами и в правильном месте.

Далее выбираем пункт меню **Создать** и в открывшемся окне указываем обычную страницу на основе HTML, как ранее было показано на рис. 8.14. Сохраняем ее под соответствующим именем — `str2.html`. Таким же образом создаем страницу с именем `str3.html`. Теперь, если посмотреть на верхнюю часть документа, то можно увидеть несколько вкладок, содержащих имена созданных и пока незаполненных страниц (рис. 8.35).



Рис. 8.35. Вкладки, содержащие имена открытых файлов

Щелкая на вкладке с именем файла, мы делаем активным соответствующий файл. Вкладка с именем активного файла при настройках по умолчанию всегда выглядит более светлой. Будьте внимательны и всегда контролируйте файл, с которым в данный момент работаете.

Создание гиперссылок на примере пунктов меню в странице-шаблоне index.html

Чтобы создать гиперссылки, делаем файл `index.html` активным, для чего щелкаем кнопкой мыши на его вкладке, расположенной в левом верхнем углу рабочей области.

Далее щелчком правой кнопкой мыши вызываем контекстное меню выделенного текста — в нашем случае надписи Главная страница — и выбираем пункт Создать ссылку. В появившемся окне указываем имя файла, на который и будет ссылаться данная надпись, у нас это файл `index.html` (рис. 8.36).

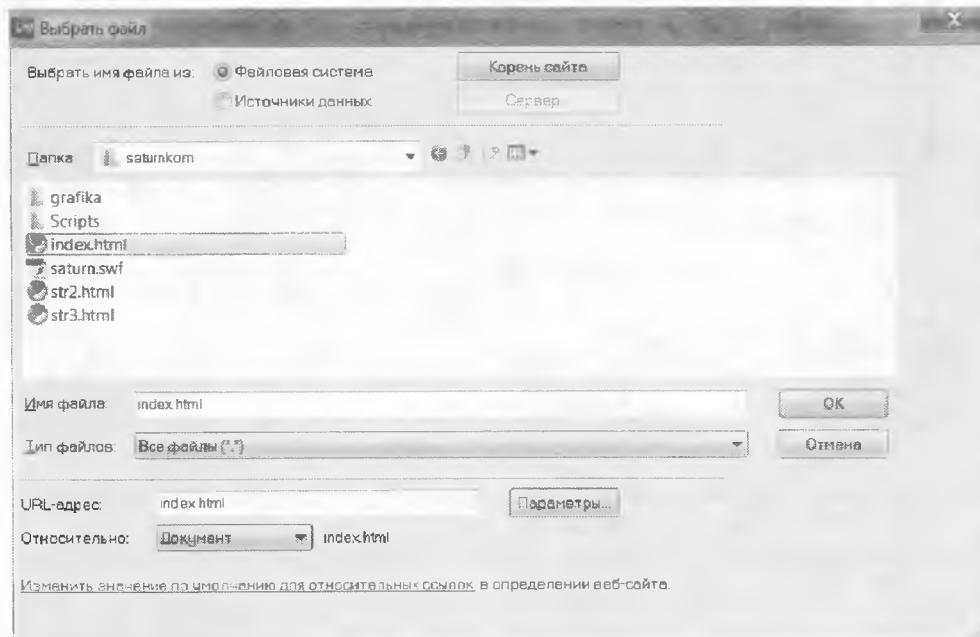


Рис. 8.36. Диалоговое окно выбора файла для связи с текстом — создания гиперссылки

При выполнении данной операции надпись, обозначающая гиперссылку, меняет цвет и подчеркивается — в зависимости от того, какие ранее были заданы настройки (рис. 8.37).

Повторив описанную процедуру еще дважды, определим файлы для других пунктов меню, связав вторую и третью надписи меню с файлами str2.html и str3.html соответственно. Таким образом, все пункты меню будут связаны гиперссылками со своими страницами (рис. 8.38).

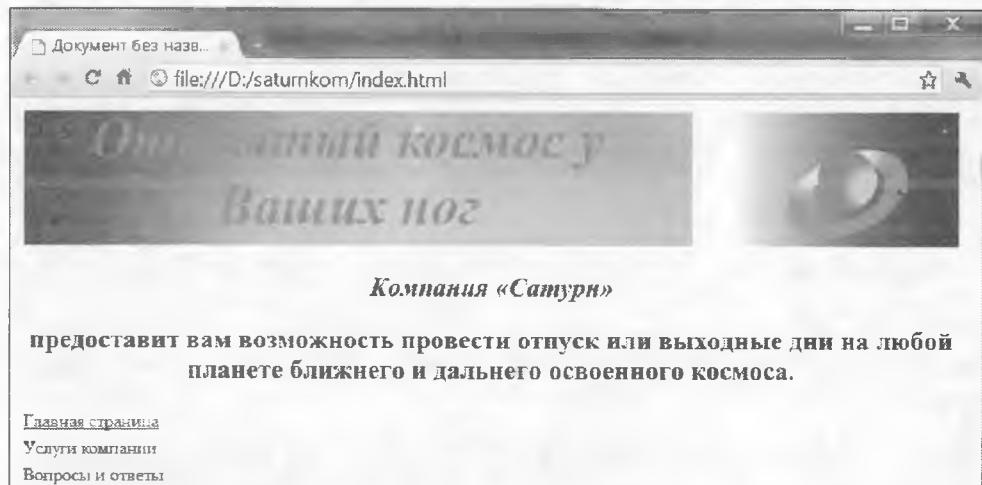


Рис. 8.37. Изменение вида надписи при создании гиперссылки

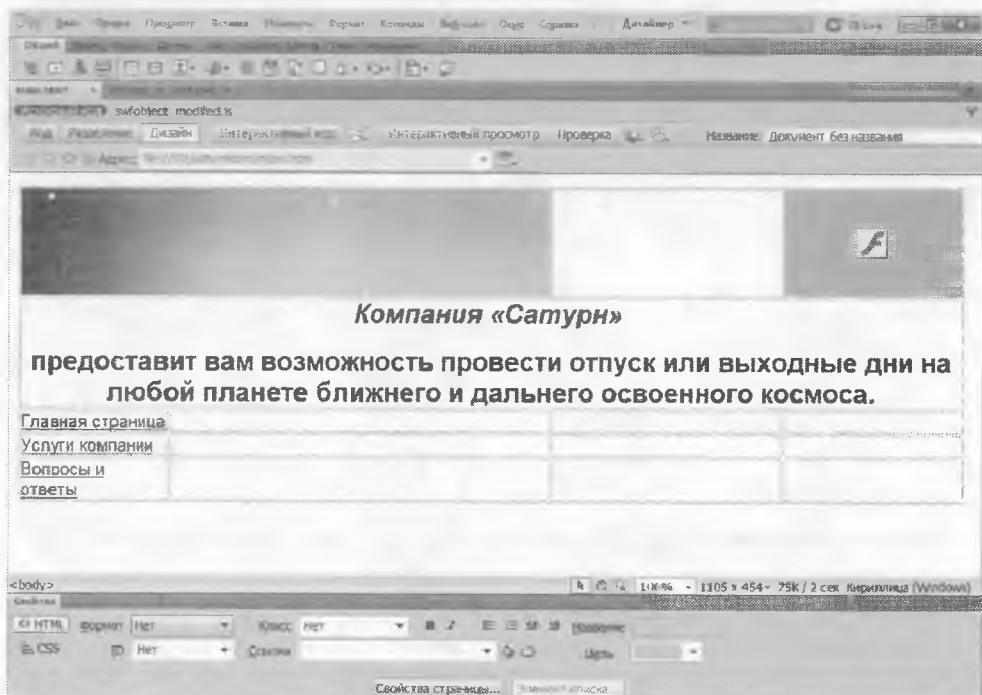


Рис. 8.38. Дизайн веб-страницы с обозначенными гиперссылками пунктов меню

8.5. Тиражирование веб-страниц

Перенос готового шаблона index.html на другие веб-страницы сайта

В файле `index.html` мы разрабатывали внешний вид веб-страницы, обязательный для всех страниц, и, надо заметить, мы в этом преуспели. Теперь настала пора просто скопировать готовую и заполненную таблицу на другие веб-страницы нашего сайта. Это можно сделать двумя способами. В первом случае нужно перейти в режим просмотра кода и скопировать весь HTML-код, лежащий между тегами `<body>` и `</body>`, а затем вставить его между такими же тегами других страниц. Во втором случае нужно, оставаясь в режиме дизайна, просто выделить таблицу со всем ее содержимым, скопировать и вставить на другие страницы (рис. 8.39).

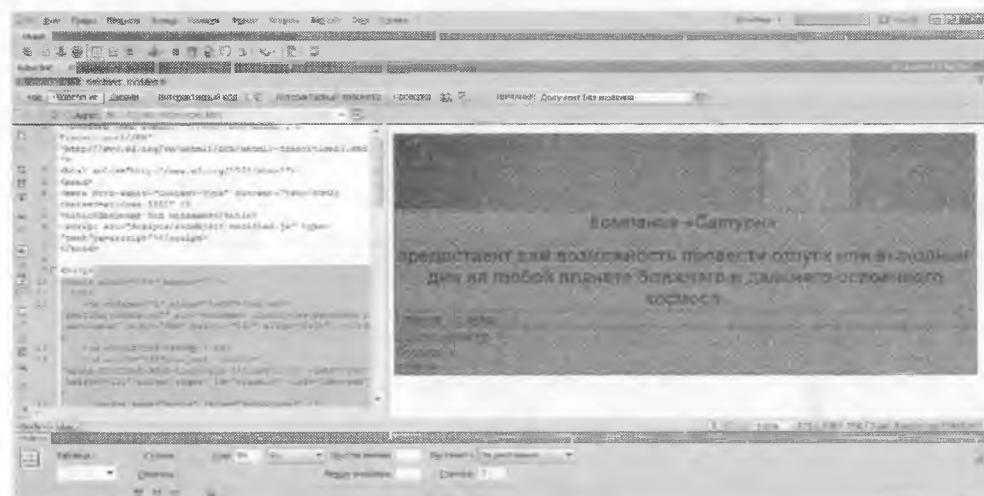


Рис. 8.39. Выделение содержимого страницы-шаблона для копирования на другие страницы сайта

Итак, тем или иным способом выделим таблицу, затем выберем пункт меню **Правка** ▶ **Копировать**. После этого перейдем на вкладку с именем файла `str2.html` и выполним команду **Правка** ▶ **Вставить** для активной страницы. Проделав такую же процедуру вставки для страницы `str3.html`, мы получим полностью идентичные по своему содержанию страницы. После вставки таблицы на страницу нужно выполнить пункт меню **Файл** ▶ **Сохранить все**.

Оформление содержания index.html — главной страницы сайта

Оформление страницы начинается, как вы помните, с оформления тега `<title></title>`, что находится в заголовке веб-страницы. Но при

использовании Adobe Dreamweaver нет необходимости искать этот тег в коде веб-страницы. Достаточно прописать Компания "Сатурн" в поле Название, которое расположено на панели Документ (рис. 8.40) и в котором по умолчанию при создании новой страницы всегда присутствует надпись Документ без названия. Вот вместо этой стандартной надписи и введем нужный нам текст.

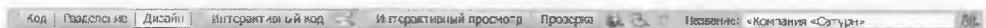


Рис. 8.40. Панель инструментов Документ с полем Название

Теперь нужно проверить, появилась ли наша надпись в строке заголовка веб-страницы (рис. 8.41).

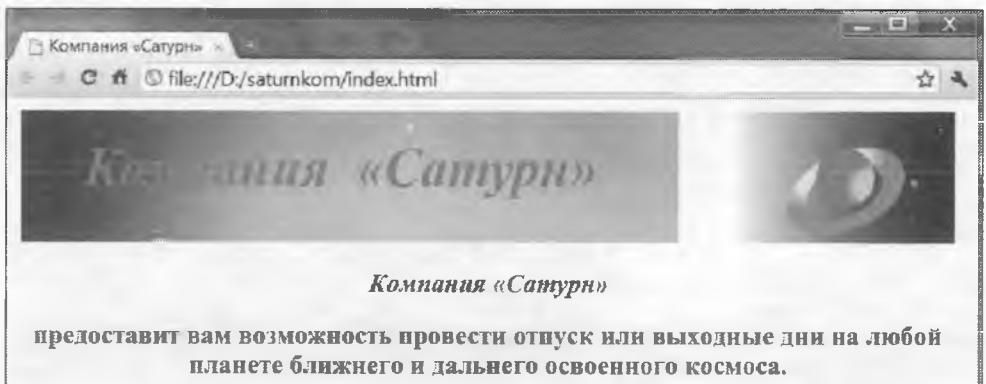


Рис. 8.41. Надпись Компания «Сатурн» в заголовке веб-страницы

Теперь посмотрим на саму веб-страницу в режиме дизайна (рис. 8.42) и увидим там пустую область, ограниченную ячейками таблицы.

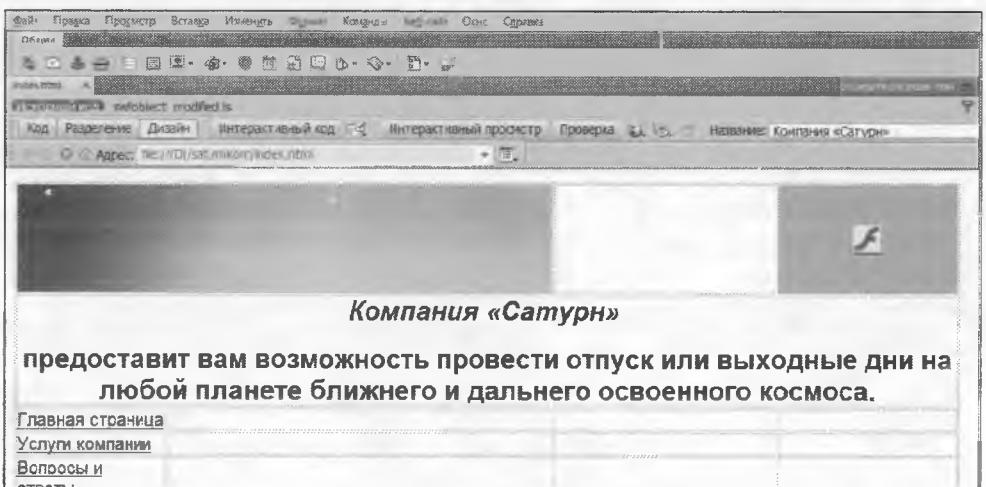


Рис. 8.42. Режим Дизайн веб-страницы

Оперируя ячейками таблицы — деля и объединяя их, добавляя или удаляя строки таблицы, можно творить жизненное пространство веб-страницы, заполняя его материалом.

8.6. Игра со сценариями на веб-страницах

Оживление веб-страницы. Добавление сценария «Простые часы» на index.html

Для придания активности веб-странице вставим на нее небольшой сценарий «Простые часы», описанный в главе 4.

Для этого нам необходимо работать не только в режиме **Дизайн**, но и в режиме **Код**, поэтому идеальный вариант — режим **Разделение**, так как он позволяет одновременно работать с исходным кодом страницы и видеть происходящие на ней изменения.

Обратимся к самому коду сценария «Простые часы». Он подробно описан в главе 4, поэтому здесь остановимся только на общем расположении частей сценария в теле веб-страницы.

Тело сценария, ограниченное тегами `<script></script>`, следует вставить в заголовок страницы (рис. 8.43).

```
<script>

function clock() {
    var time = new Date()
    var hour = time.getHours()
    var minute = time.getMinutes()
    var second = time.getSeconds()
    var temp = "" + hour
    temp +=(minute <10) ? ":0" : ":"
    temp +=(second <10) ? ":0" : ":"
    document.clock.digits.value = temp
    id =setTimeout("clock()",1000)
} </script>
```

Теперь для вызова сценария в момент загрузки страницы пропишем в теге `<body>` событие, вызывающее функцию `clock()` (рис. 8.44).

```
<body onLoad="clock()">
```

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html;
6 charset=windows-1251" />
7 <title>Компания &laquo;Сатура&raquo;</title>
8 <script src="Scripts/swfobject_modified.js" type="text/javascript">
9 </script>
10 <script>
11 function clock() {
12 var time = new Date()
13 var hour = time.getHours()
14 var minute = time.getMinutes()
15 var second = time.getSeconds()
16 var temp = "" + hour
17 temp += minute < 10 ? "0" + minute : minute
18 temp += second < 10 ? "0" + second : second
19 document.getElementById("clock").innerHTML = temp
20 }
21 <script>
22 </head>
23
24 <body onLoad="clock()">
25 <table width="99%" border="0">
```

Рис. 8.43. Вставка кода сценария «Простые часы» в заголовок веб-страницы

Рис. 8.44. Организация вызова сценария при загрузке веб-страницы

Осталось вставить форму, содержащую область вывода часов. Для этого выделим ячейку таблицы, в которой будет размещена область часов в ре-

жиме Дизайн, а в выделенную область раздела Код вставим код самой формы (рис. 8.45).

```
<form name="clock">

<input type="text" name="digits" size=8 value="">

</form>
```

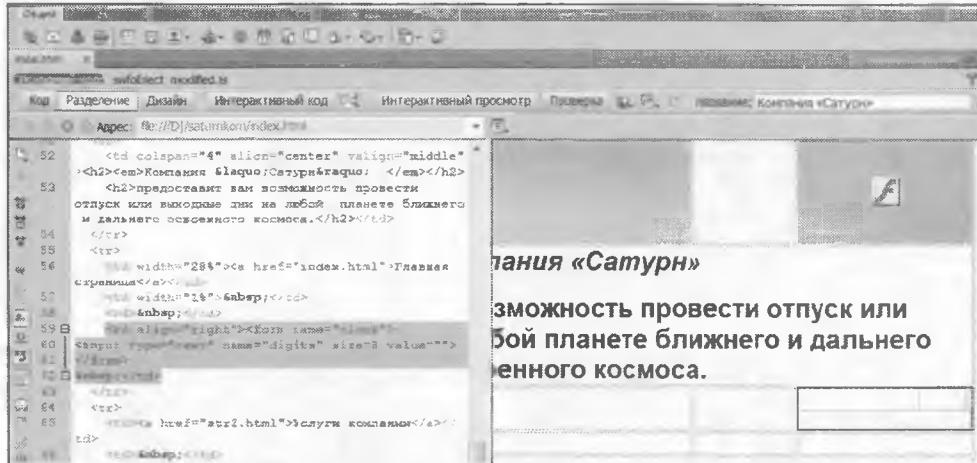


Рис. 8.45. Вставка кода формы, содержащей вывод области часов, в отведенную ячейку таблицы

Сохранив страницу, можно просмотреть, что получилось в результате (рис. 8.46).

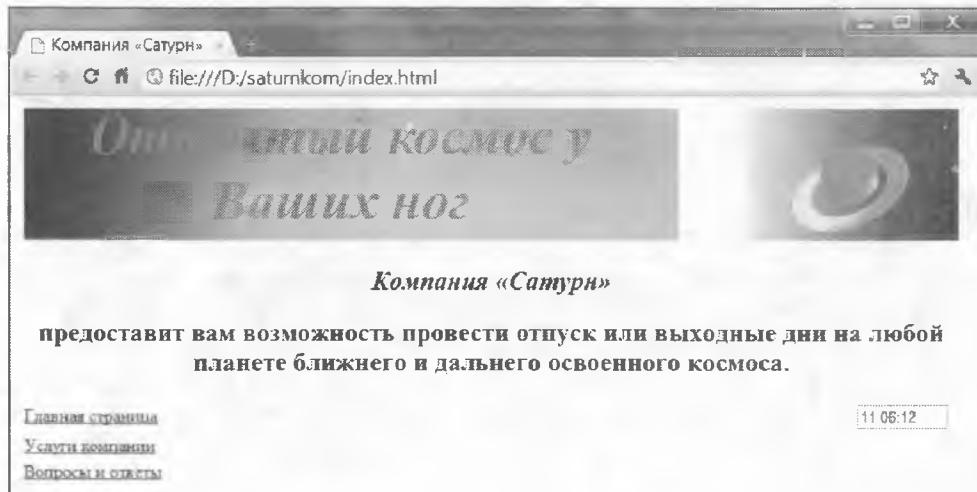


Рис. 8.46. Веб-страница со вставленным сценарием «Простые часы»

Далее заполняем свободную область веб-страницы необходимым материалом и переходим к оформлению следующих страниц.

Оформление страницы «Услуги компании». Вставка сценария «Текущая дата»

Если мы опять обратимся к главе 4, где происходило знакомство с языком программирования JavaScript и его конструкциями, то увидим, что для оформления страницы «Услуги компании» можно использовать сценарий «Текущая дата». Его код, как всегда, помещается в заголовок веб-страницы (рис. 8.47).



```

4 <meta http-equiv="Content-Type" content=
5 "text/html; charset=windows-1251" />
6 <title>Документ без названия</title>
7 <script src="Scripts/swfobject_modified.js" type=
8 "text/javascript"></script>
9 <script>
10 var time= new Date()
11 function vremya(){
12     var day=time.getDay()
13     var dayname
14     if (day==0) dayname="Воскресенье"
15     if (day==1) dayname="Понедельник"
16     if (day==2) dayname="Вторник"
17     if (day==3) dayname="Среда"
18     if (day==4) dayname="Четверг"
19     if (day==5) dayname="Пятница"
20     if (day==6) dayname="Суббота"
21     var monthnames=new Array ("Января", "Февраля",
22     "Марта", "Апреля", "Мая", "Июня", "Июля",
23     "Августа", "Сентября", "Октября", "Ноября",
24     "Декабря");
25     var month=time.getMonth()
26     var monthname=monthnames[month]
27     var year=time.getYear()
28     if (year<1000) year+=1900
29     var strokadate=dayname +', '+time.getDate()+' '+
30     monthname+', '+year;
31     document.write(strokadate) }
32 </script>

```

Рис. 8.47. Добавление сценария в заголовок веб-страницы

```

<script>

var time= new Date()

function vremya(){

var day=time.getDay()

var dayname

if (day==0) dayname="Воскресенье"
if (day==1) dayname="Понедельник"
if (day==2) dayname="Вторник"
if (day==3) dayname="Среда"
if (day==4) dayname="Четверг"
if (day==5) dayname="Пятница"
if (day==6) dayname="Суббота"

var monthnames=new Array ("Января", "Февраля", "Марта", "Апреля", "Мая",
"Июня", "Июля", "Августа", "Сентября", "Октября", "Ноября", "Декабря");

var month=time.getMonth()

var monthname=monthnames[month]

var year=time.getYear()

if (year<1000) year+=1900

var strokadate=dayname +', '+time.getDate()+' '+monthname+', '+year;
document.write(strokadate) }

</script>

```

Сам сценарий вызывается непосредственно из текста, размещенного на веб-странице. Однако его придется немного изменить по содержанию, то есть вместо:

"Компания Сатурн предлагает вам провести свои выходные дни или весь отпуск, совершив следующие увлекательные путешествия:"

сделаем следующее:

"Компания Сатурн предлагает вам провести свои выходные дни или весь отпуск с сегодняшнего дня <script>vremya()</script>, совершив следующие увлекательные путешествия:"

Перед вставкой текста в ячейки таблицы их нужно объединить. Кроме того, помните, что текст можно вставлять в таблицу в режиме **Дизайн**, а вызов сценария можно записывать только в режиме **Код**, иначе вместо выполненного сценария на веб-странице можно получить только текст вызывающего его события (рис. 8.48).

Перейдя в режим Код, можно исправить положение с вызовом сценария «Текущая дата» (рис. 8.49).

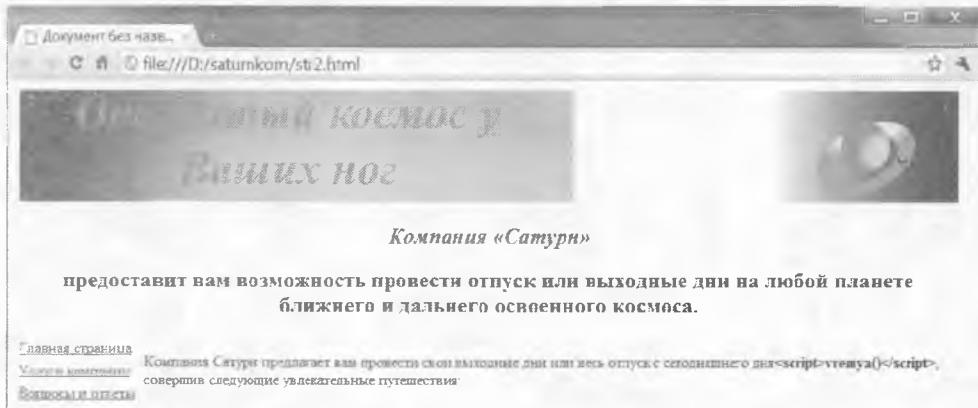


Рис. 8.48. Ошибка при записи события, вызывающего сценарий

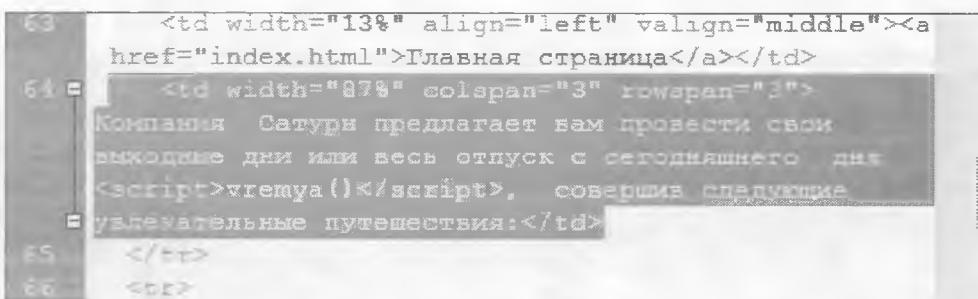


Рис. 8.49. Исправление ошибки вызова сценария в режиме Код

Результат выполненных действий с выровненным впоследствии текстом можно наблюдать на рис. 8.50.

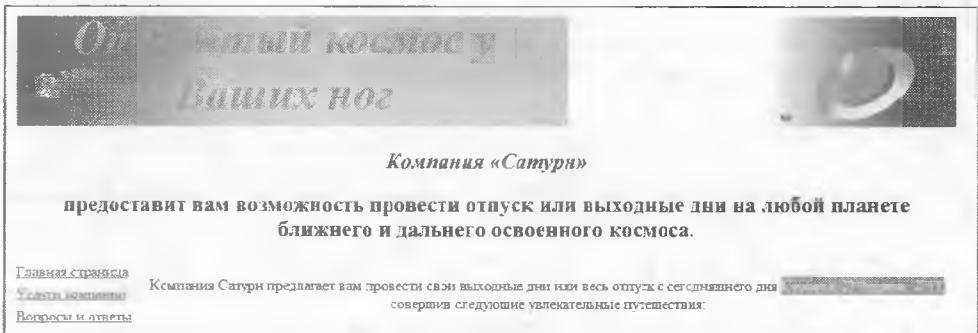


Рис. 8.50. Работа сценария «Текущая дата» на веб-странице

Теперь обычным способом вставляем тот текст, который нам нужен на данной странице. После этого форматируем его стандартными способами. И вот тут-то начинается путаница: все форматирование сразу сбивается и становится непонятно, что делать (рис. 8.51).

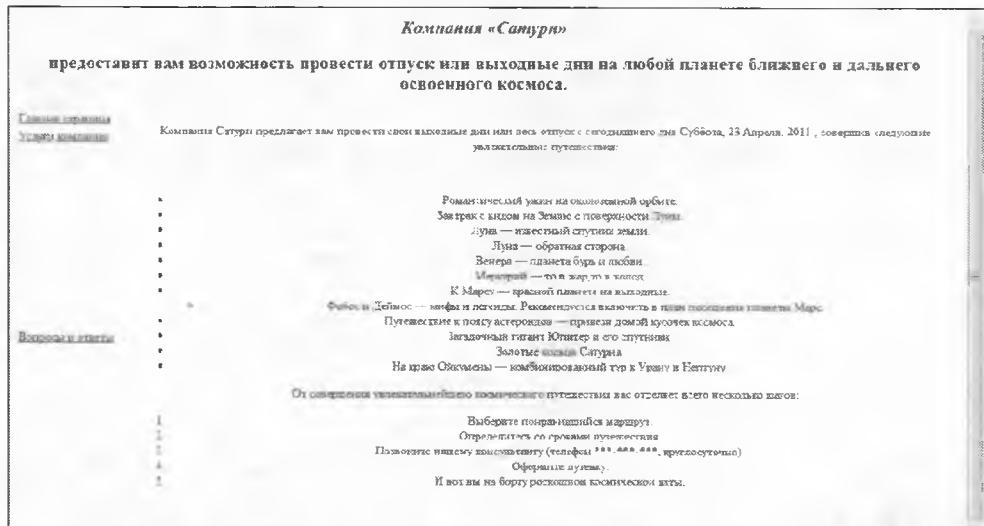


Рис. 8.51. Сбой пунктов меню при вставке дополнительного текста

Дело в том, что у нас всего одна большая объединенная ячейка, выделенная под текст. Ее нижняя граница является нижней границей всей таблицы, поэтому и нижняя ячейка левого столбца границы растягивается вниз вместе со всей таблицей, смещая пункт меню Вопросы и ответы.

Решений этой задачи как минимум два. Первое: вынести текст за пределы таблицы, так как он может самостоятельно перетекать при изменении ширины страницы. И второе: работать с ячейками таблицы, разделяя и объединяя их. Мы выберем второй вариант.

Прежде всего установим курсор в ячейку с пунктом меню Вопросы и ответы и выберем пункт контекстного меню Таблица ▶ Разбить ячейку. В появившемся окне укажем, что количество строк равно двум (рис. 8.52).

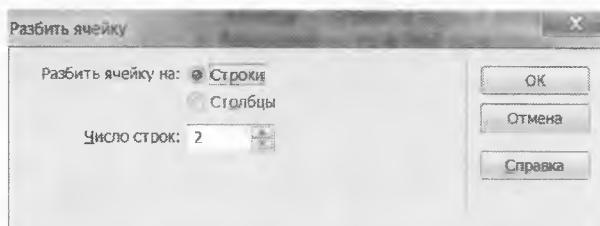


Рис. 8.52. Диалоговое окно Разбить ячейку

Разбив ячейку, мы подняли пункт меню Вопросы и ответы на положенное ему место (рис. 8.53).

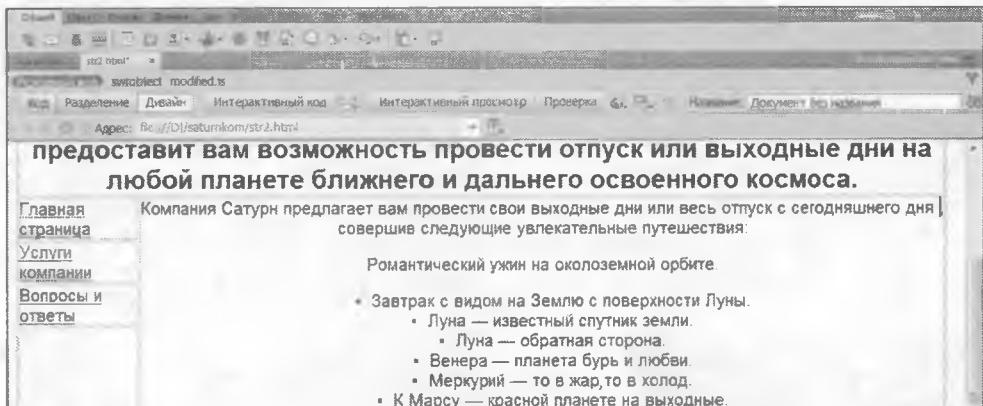


Рис. 8.53. Результат разбиения ячейки таблицы и возврат пункта меню на исходное место

Для форматирования текста на веб-странице необходимо пользоваться меню Формат программы Adobe Dreamweaver. Результат оформленной и готовой к работе страницы можно видеть на рис. 8.54.

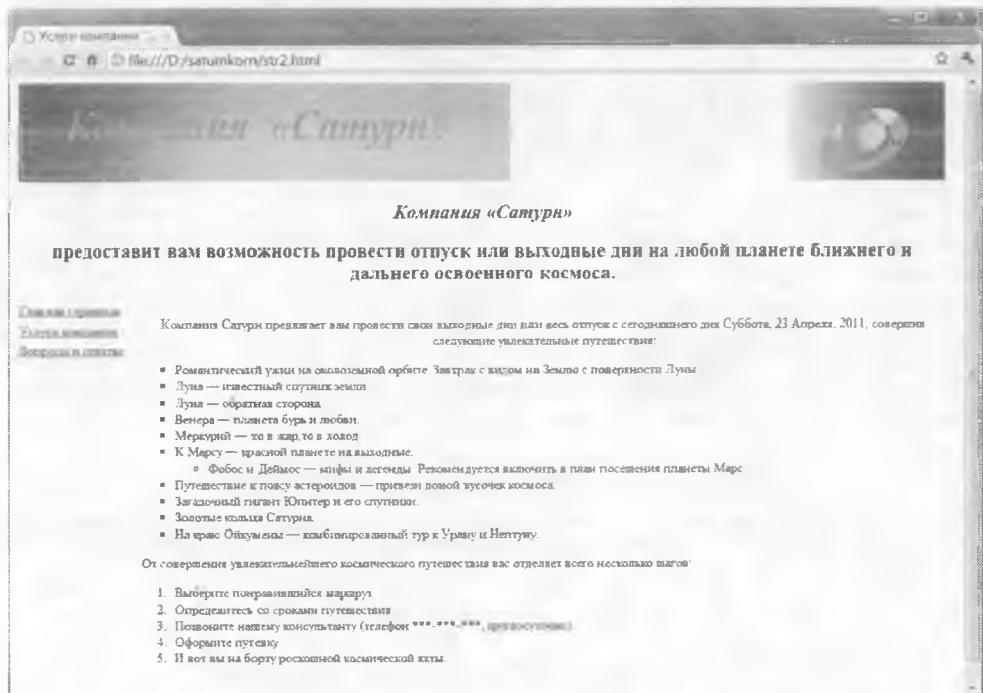


Рис. 8.54. Оформленная страница «Услуги компании»

Как видно из примера оформления данной веб-страницы, нет абсолютно ничего сложного как при форматировании текста, содержащегося на странице, так и при вставке в него сценария. Трудности, возникающие в момент работы с таблицами, легко разрешимы при включении простой логики.

8.7. Доделки

Оформление третьей страницы сайта. Электронная почта. Якори для организации внутренних ссылок и переходы к ним. На примере страницы str3.html разберем добавление ссылки на вызов почтового клиента, чтобы иметь возможность отправки почтового сообщения на электронный адрес компании. Как вы помните, текст пригласительного сообщения гласит: «Уважаемые посетители сайта компании „Сатурн“, если вы не нашли ответ на интересующий Вас вопрос, то, пожалуйста, обратитесь к нам письменно, активировав данную ссылку. Спасибо». Здесь ссылкой являлась выделенная фраза активировав данную ссылку.

Для организации ссылки на почтовый адрес необходимо выделить текст, который будет гиперссылкой, и выбрать пункт меню Вставка ▶ Ссылка на адрес электронной почты. В результате откроется диалоговое окно, где следует указать адрес электронной почты, на которую и должны идти письма (рис. 8.55).

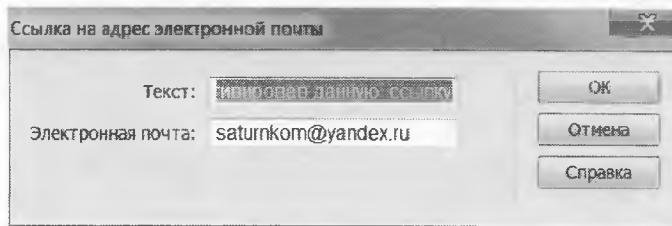


Рис. 8.55. Окно Ссылка на адрес электронной почты

Таким образом, надпись активировав данную ссылку станет выделенной и подчеркнутой, как и положено уважающей себя гиперссылке.

Для организации переходов между вопросами, как и в главе 2, где происходило знакомство со спецификацией языка HTML, сначала нужно установить якори. На них и будут уводить пользователя установленные гиперссылки. Это делается следующим образом: сначала следует выделить текст, который будет якорем, а затем выбрать пункт меню Вставка ▶ Именованная привязка. В открывшемся окне нужно заменить все пробелы символами подчеркивания либо вместо предложенного текста написать свой, согласно требованию спецификации языка — в одно слово (рис. 8.56).

После нажатия кнопки OK рядом с надписью появится значок якоря. Повторив вышеописанные действия столько раз, сколько у нас вопросов, мы получим веб-страницу, размеченную значками якорей (рис. 8.57).

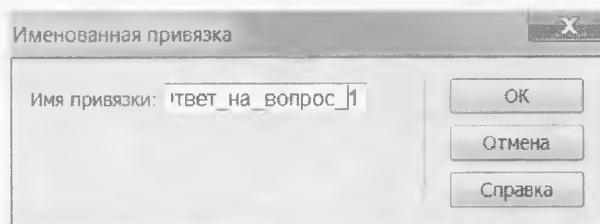


Рис. 8.56. Диалоговое окно Именованная привязка

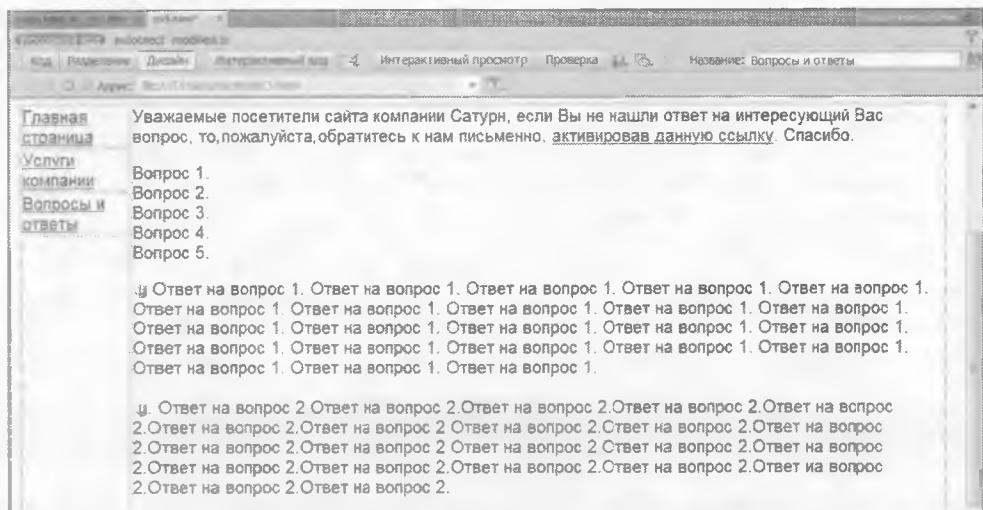


Рис. 8.57. Веб-страница с установленными якорями и ссылкой на вызов почтового клиента

Теперь необходимо связать текст гиперссылок с якорем. Для этого выделяем надпись **Вопрос 1**, которая будет гиперссылкой к первому якорю, и выбираем пункт меню **Вставка ▶ Гиперссылка**. В открывшемся окне указываем якорь, на который и будет выполнен переход при щелчке на гиперссылке (рис. 8.58).

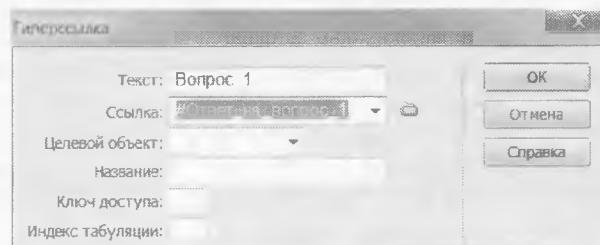


Рис. 8.58. Диалоговое окно Гиперссылка с указанием якоря в поле Ссылка

Повторив данные действия несколько раз, мы получим готовую к размещению на сервере страницу `str3.html` (рис. 8.59).

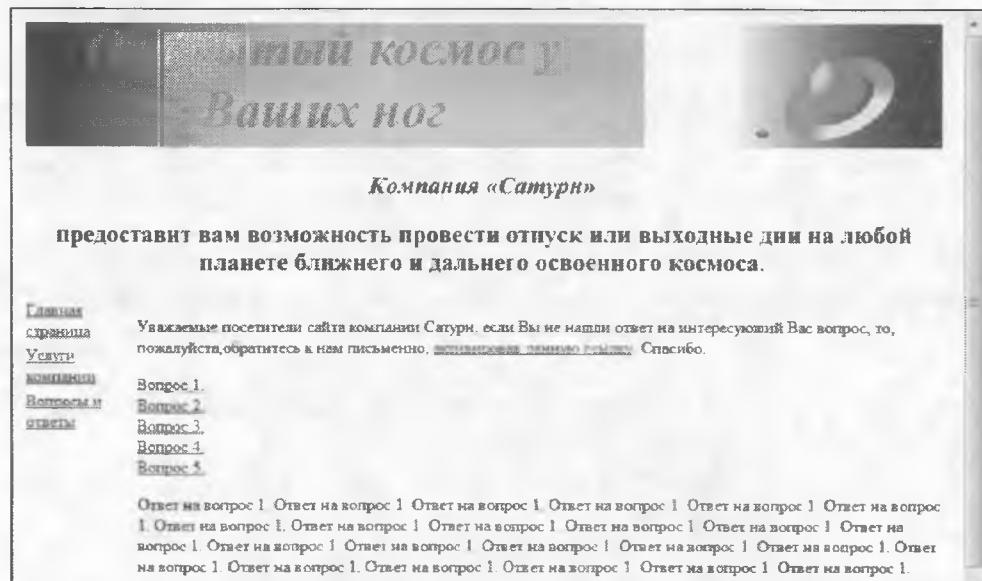


Рис. 8.59. Готовая к размещению на веб-сервере страница str3.html

8.8. Работа с хостингом штатными средствами

Загрузка сайта на хостинг через Adobe Dreamweaver

Поскольку сайт, состоящий из трех страниц, уже готов, пора подключиться к Сети и выложить его на хостинг. Это актуально для тех, кто работал в автономном режиме, то есть без связи с сервером, и сохранял файлы на локальном компьютере. Если при создании сайта в программе Adobe Dreamweaver вы были подключены к Сети постоянно, то страницы сохранились на сервер автоматически, при каждом сохранении на локальном компьютере, и теперь вы имеете страницы, размещенные на внешнем хостинге. В ином случае каждый раз при сохранении страниц сайта вы получали следующее сообщение (рис. 8.60).

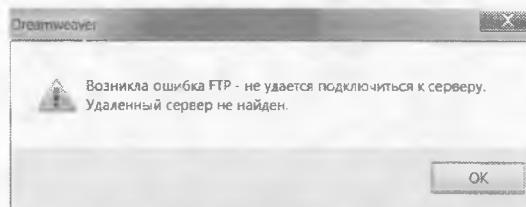


Рис. 8.60. Сообщение Adobe Dreamweaver о невозможности сохранить файлы на сервере удаленного хоста

В нашем случае для сохранения страниц на удаленном хостинге при подключении к Интернету нужно просто сохранить все страницы. Для этого выполним команду **Файл > Сохранить все**, и программа автоматически сохранит страницы сайта на удаленном хостинге.

Если при сохранении файлов на удаленном веб-сервере возникают проблемы, то следует заново проверить все сетевые настройки. Если же и повторная попытка не сделала удаленный сервер доступным, попробуйте установить либо снять дополнительные флажки в окне базовых настроек сайта, где описывается FTP-сервер. Обратите внимание на флажки **Использовать пассивный FTP** и **Использовать альтернативный метод перемещения файлов на FTP** (рис. 8.61).

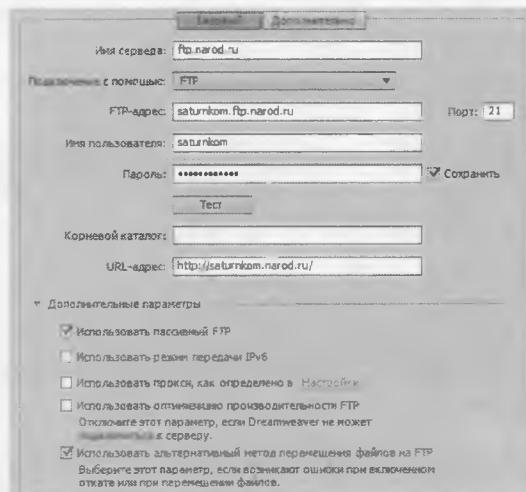


Рис. 8.61. Окно настроек доступа к удаленному хосту

Когда связь будет установлена, воспользуйтесь окном **Файлы** и нажмите кнопку **Синхронизировать с** (рис. 8.62).

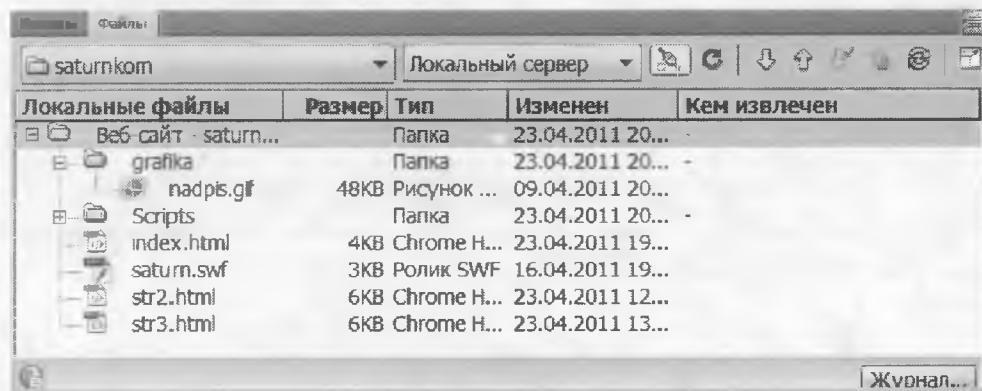


Рис. 8.62. Окно Файлы, предназначенное для управления файлами сайта

На заданный программой вопрос необходимо ответить положительно (рис. 8.63).

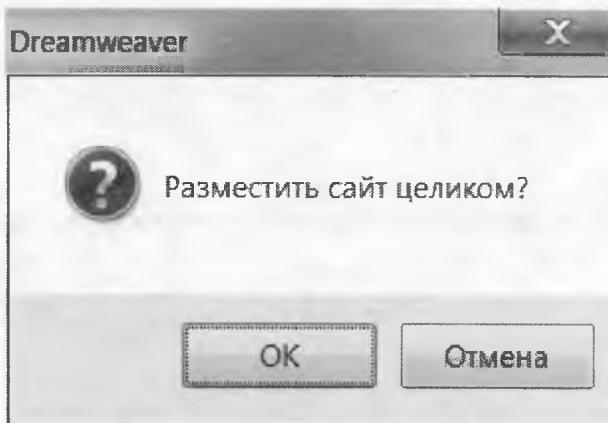


Рис. 8.63. Запрос программы, выводимый при синхронизации файлов на локальном и удаленном хостингах

После синхронизации можно набрать в адресной строке браузера URL-адрес сайта и наблюдать его во всей красе и доступности для пользователей (рис. 8.64).

На этом можно было бы закончить краткий экскурс в возможности визуального редактора Adobe Dreamweaver. Но, как показывает практика, не надо торопиться. Прошу вас уделить мне еще немного внимания.



Рис. 8.64. Веб-страница, размещенная на удаленном хостинге при синхронизации сайтов

Добавление новых страниц на сайт. Проверка гиперссылок

Рано или поздно сайт будет расширяться, то есть будут появляться новые страницы. Пока их немного, добавление очередной страницы не вызывает трудностей. В этом случае алгоритм действий веб-мастера достаточно прост.

1. Создать новую веб-страницу.
2. Скопировать на нее шапку-шаблон с любой другой существующей страницы.
3. На каждой веб-странице сайта добавить строку в столбец, содержащий пункты меню, куда и прописать новый пункт меню.
4. Организовать гиперссылки для нового пункта меню со всех страниц сайта и проверить их работоспособность.
5. Заполнить вновь созданную страницу сайта.
6. Провести синхронизацию локального сайта с сайтом, находящимся на удаленном хостинге.

Как видите, алгоритм достаточно прост, но это кажущаяся простота. Дело в том, что при увеличении количества элементов системы она становится все более и более громоздкой. Соответственно, уследить за постоянно расширяющимся сайтом становится все сложнее и сложнее, взять хотя бы проверку гиперссылок при организации пунктов меню для новых страниц.

Однако разработчики программы Adobe Dreamweaver учли и этот момент, дав веб-мастерам возможность автоматически проверять гиперссылки. Для проверки рабочих и битых гиперссылок сайта нужно выполнить пункт меню **Сайт** ▶ **Проверить ссылки на всем сайте**. В открывшемся окне Проверка ссылок можно просмотреть список работающих и неработающих гиперссылок, которые при необходимости легко исправить (рис. 8.65).

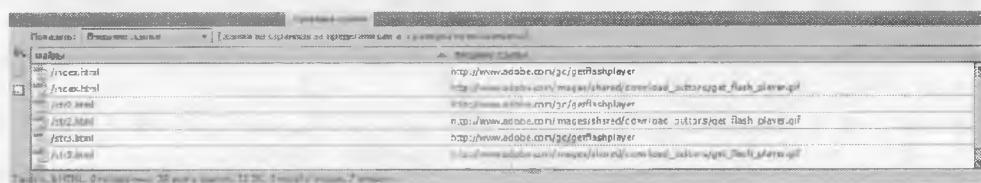


Рис. 8.65. Окно проверки гиперссылок на сайте

Для исправления гиперссылок автоматически по всему сайту нужно выполнить пункт меню **Сайт** ▶ **Изменить ссылку по всему сайту** и заполнить поля в открывшемся окне (рис. 8.66).

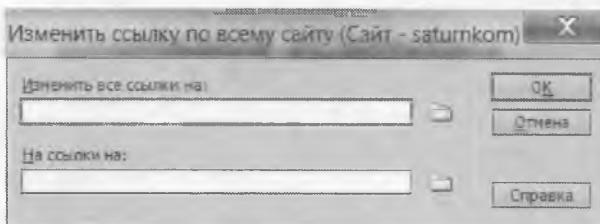


Рис. 8.66. Окно изменения гиперссылок по всему сайту

Гиперссылки на скачивание файлов

Очень часто возникает необходимость разместить на сайте определенные файлы таким образом, чтобы пользователь мог их скачать для последующей работы. Это довольно легко сделать.

1. Предварительная подготовка файлов.

- Файлы, имеющие большой объем, архивируются.
- Если подготовленный архивный файл очень большой, то его следует разбить на многотомный архив и указать гиперссылки на каждый архивный том отдельно.
- Взаимосвязанные папки или группы файлов архивируются, так как по Сети можно передавать только файл, а не папку. В противном случае пользователь скачает только ярлык папки, а не ее содержимое.

2. Размещение подготовленных файлов в специальный каталог с установленными правами доступа 777.
3. Определение в тексте страницы слова (или фразы), которое и будет целевой гиперссылкой на скачивание файла. Окно, открываемое при установке параметров гиперссылки, представлено на рис. 8.67.

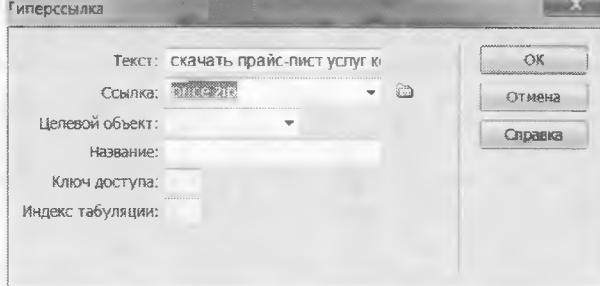


Рис. 8.67. Окно определения параметров гиперссылки при указании на файл для скачивания в качестве целевого объекта

Глава 9

CMS JOOMLA!

9.1. О системах управления содержимым

В предыдущих главах в общих чертах были описаны инструменты и методы, которые используются при создании веб-страниц и сайтов. Предложенной информации вполне достаточно для создания небольшого ресурса маленькой компании с минимальными затратами. Если сайт будет расширяться, то количество инструментов и методов создания различных элементов также будет увеличиваться, соответственно, будут расти требования к уровню подготовки веб-мастера — сначала одного, а затем еще нескольких. В свою очередь, возрастут как временные издержки на обслуживание сайта, так и денежные, а это не всегда оправдано. К такому выводу давно пришли крупные компании, оценив параметры эффективности и обслуживания корпоративных сайтов, а затем подтянулись и энтузиасты. Таким образом, сегодня у нас есть не просто выбор, а достаточно широкий выбор специализированных инструментов, значительно облегчающих создание и поддержание работоспособности как личного сайта, так и сайта компании любого уровня.

Итак, CMS — системы модификации контента. Для русского уха более привычный перевод звучит так: системы управления содержимым. Что это? Специальный программный продукт, разработанный для автоматизации создания и обслуживания сайтов. При использовании CMS достаточно в начале создания сайта самостоятельно сделать настройки или, если нет времени на изучение системы, пригласить специалиста, а затем поручить наполнение страниц любому сотруднику компании, имеющему навыки работы с офисными приложениями. При этом нет необходимости выделять для него рабочее место, ведь сайт находится в Сети и, соответственно, управляемся может с любого компьютера, подключенного к ней, хотя бы из интернет-кафе, что находится в соседнем доме.

Систем модификации контента много. Практически каждый платный хостинг и все чаще бесплатные, например www.ucoz.ru, имеет честь предложить своим клиентам специально разработанное приложение для удобства создания и обслуживания сайта на данном хостинге. При таком огромном выборе различных CMS почему именно Joomla!?

Joomla! — прежде всего программа с открытым кодом, написанным на языке программирования PHP, что, в свою очередь, дает вам право и воз-

можность изменять эту систему в зависимости от ваших предпочтений. Немаловажно также, что, как и большинство программ с открытым кодом, проект Joomla! разрабатывается группой энтузиастов, в состав которых можно органически влиться, и поэтому сама система распространяется полностью бесплатно.

Требования у Joomla! довольно скромные, и созданный в ней сайт с успехом может мигрировать с хостинга на хостинг, например при смене провайдера. Огромный плюс системы Joomla! состоит в том, что в Сети распространено огромное количество шаблонов сайтов, написанных специально для этой системы. И, пожалуй, самое главное то, что у CMS Joomla! достаточно простая структура управления страницами и интуитивно понятный интерфейс управления.

Если вкратце рассматривать возможности Joomla!, то получится следующий небольшой список.

- Полноценное создание сайта и управление им через веб-интерфейс, позволяющий контролировать основные элементы сайта.
- Функция управления пользователями-авторами контентом сайта и возможность модерирования их сообщений.
- Иерархическое управление группами пользователей и элементами управления меню сайта.
- Автоматизация регистрации пользователей на сайте и полная защита пользовательских данных.
- Способность создавать статьи и автоматически публиковать или удалять их на сайте в заданное время.
- Наличие в интерфейсе программы визуального WYSIWYG-редактора, который позволяет размещать информацию на веб-страницах без знания языка HTML.
- Поддержка новостных лент.
- Создание опросов-голосований по материалам сайта и их обработка.
- Размещение рекламных баннеров на веб-страницах и управление ими.
- Организация гостевой книги или форума сайта.
- Расширение способностей сайта путем подключения дополнительных плагинов и, как следствие, организация на сайте, например, магазина сетевой торговли.
- Мультиязычность системы Joomla! и полностью русифицированный интерфейс.
- Работа в различных операционных системах: как Windows, так и Linux различных версий и сборок.

Для краткого ознакомления, я думаю, достаточно. Теперь имеет смысл, не откладывая дела в долгий ящик, перейти к изучению самой системы Joomla!.

Принцип действия систем управления содержимым

В отличие от сайтов, написанных на языке гипертекстовой разметки HTML, ресурсы, создаваемые в системах управления содержимым, строятся с помощью программных средств. Соответственно, принципы построения и формирования для последующего отображения веб-страниц будут различаться.

Рассмотрим рис. 9.1, на котором показан весь путь веб-страницы от пользователя до пользователя, написанной на языке гипертекстовой разметки HTML и хранящейся на жестком диске компьютера или веб-сервера.

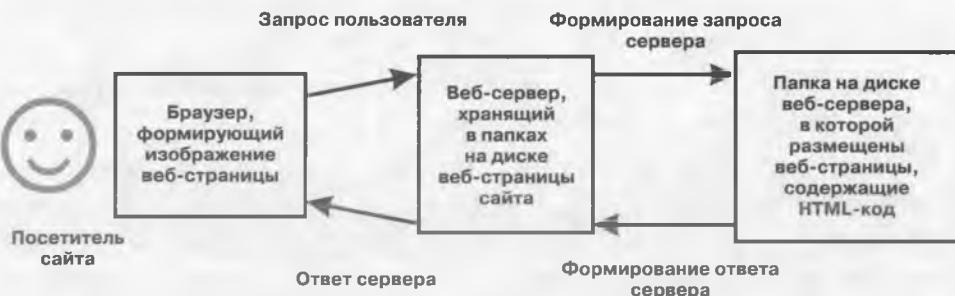


Рис. 9.1. Формирование страницы, написанной на языке HTML, при ее хранении на обычном хостинге

При выборе пользователем какой-либо гиперссылки или наборе адреса сайта в адресной строке браузера начинает работать следующая схема.

1. Браузер формирует запрос к веб-серверу на предоставление информации, находящейся по заданному адресу.
2. Веб-сервер, в свою очередь, разыскивает в указанной папке веб-страницу с заданным именем, формирует ее код для ответа и пересыпает его по запросу браузеру.
3. Получив ответ веб-сервера, браузер интерпретирует код веб-страницы в визуальные образы на экране монитора.
4. Пользователь доволен и наслаждается.

Если мы рассматриваем способ формирования веб-страницы средствами любой системы модификации контента, то данная схема будет выглядеть по-другому (рис. 9.2).



Рис. 9.2. Формирование страницы средствами CMS Joomla!

Чем же обусловлено различие передачи информации в рис. 9.1 и 9.2? Ведь браузер в обоих случаях получает HTML-код веб-страницы. Все дело в различной организации хранения данных, содержащихся на веб-страницах.

В случае классической веб-страницы составляющие ее данные вместе с элементами форматирования хранятся непосредственно в теле самой веб-страницы в виде совокупности текста и HTML-кода и не требуют посредников при передаче содержимого браузеру.

При использовании систем управления содержимым подход принципиально другой. Вся информация, содержащаяся на веб-страницах, хранится в виде обычного текста в базе данных SQL-сервера — специальной программы хранения данных, обеспечивающей формирование ответов на запросы из этих сведений. CMS Joomla!, как посредник между веб-сервером и SQL-сервером, формирует запрос к определенным ячейкам базы данных SQL-сервера, получает из них текстовый контент веб-страницы, прогоняет его через шаблон и на выходе передает веб-серверу сгенерированный HTML-код веб-страницы, где согласно ранее настроенному шаблону выполнено форматирование содержимого.

Как видите, эта чуть более усложненная по сравнению с классической схема хранения и генерирования веб-страниц дает на выходе сплошные плюсы.

- Достаточно один раз оформить шаблон, и все запрашиваемые страницы будут выглядеть как одна. Для смены дизайна или форматирования сайта можно изменить только один шаблон, а не все страницы сайта.
- При увеличении или уменьшении количества страниц и соответствующих пунктов меню само меню будет автоматически генерироваться для всех страниц сайта.
- Используя настройки шаблона, можно ограничивать доступ к определенной информации на сайте для различных категорий пользователей.

Таким образом, удобства в создании и использовании сайта на основе систем модификации контента превышают главный их недостаток — наличие SQL-сервера и поддержку веб-сервером языка программирования PHP. Иными словами, для создания сайта на базе CMS Joomla! бесплатный хостинг narod.ru нам не подойдет, поскольку на нем пока не работает SQL-сервер и он не поддерживает PHP.

Сразу же становится острым вопрос с хостингом. Существующие бесплатные хостинги, как правило, не выполняют резервного копирования баз данных и обычно не несут ответственность за их сохранность при форс-мажорных обстоятельствах. При их использовании все операции по техническому обслуживанию и резервному копированию баз данных ложатся на плечи веб-мастера. Хотя данная операция достаточно проста, на нее, как обычно, всегда не хватает времени. Соответственно, в этом случае при обрушении сайта вы останетесь с архивом базы данных неизвестно какой давности.

Коммерческие хостинги регулярно выполняют резервное копирование баз данных владельцев сайтов, и в случае не только технических сбоев, но и собственной неаккуратности всегда можно получить несколько архивных копий базы данных, сохраненных за оговоренный в договоре промежуток времени.

Пока же вопрос с хостингом сайта решается, попробуем создать сайт на основе системы Joomla! на персональном компьютере. Разве такое возможно? Конечно.

ХАМРР, или друзья Apache

Для создания сайта на персональном компьютере и последующей его демонстрации людям, принимающим решение о платном или бесплатном хостинге, необходимо сначала создать среду, в которой и будет жить создаваемый сайт.

Среда для системы управления содержимым Joomla! должна включать в себя следующее.

- Веб-сервер для приема запросов от браузера и формирования ему ответов по ним.
- SQL-сервер для хранения и обработки базы данных с контентом сайта.
- Интерпретатор языка программирования PHP, возможностями которого пользуется сама Joomla!.

Все эти три составляющих можно скачать по отдельности и настраивать каждый компонент самостоятельно, но этот способ трудоемкий, требует определенных базовых знаний для решения возникающих конфликтов

и в целом длительный. Поэтому воспользуемся уже готовым решением и посетим сайт <http://www.apachefriends.org/ru/xampp.html> (рис. 9.3).



Рис. 9.3. Русскоязычная страница проекта apachefriends.org

Проект [apachefriends.org](http://www.apachefriends.org) уже тем хорош, что представляет собой скомпонованный и заранее настроенный дистрибутив необходимых нам программ: веб-сервера Apache, SQL-сервера MySQL и интерпретаторов языков PHP и Perl. Дистрибутив проекта реализован в различных версиях для разных операционных систем, и вся установка ансамбля инструментов сводится к установке дистрибутива. Инструкции по пошаговой инсталляции данного комплекта программ для выбранной вами операционной системы находятся тут же, на сайте проекта. Вне зависимости от системы работа всего комплекса программ является стабильной и устойчивой. Обращаю ваше внимание на то, что дистрибутив XAMPP предназначен для проектирования и отладки сайтов на локальном компьютере и не рекомендован к использованию в качестве серверной сборки, так как не имеет жесткой защиты SQL-сервера от несанкционированного доступа. Поэтому, когда вы дорастете до создания своего портала на собственном сервере, компоновать среду вам придется из отдельных программок-серверов, делая особый упор на информационную безопасность сайта.

Пока же скачайте дистрибутив XAMPP с сайта проекта и установите данный комплекс программ согласно инструкциям для вашей операционной системы. Мы будем рассматривать работу комплекса XAMPP для операционной системы Windows.

После установки программы в области уведомлений появится характерный значок оранжевого цвета со стилизованной белой буквой Х. Такой же ярлык вы увидите на Рабочем столе, и в меню программ будет организована группа Apache Friends. Таким образом, запустить панель управления можно разными способами. После запуска контрольной панели вы увидите ее в нижнем правом углу экрана, как на рис. 9.4.

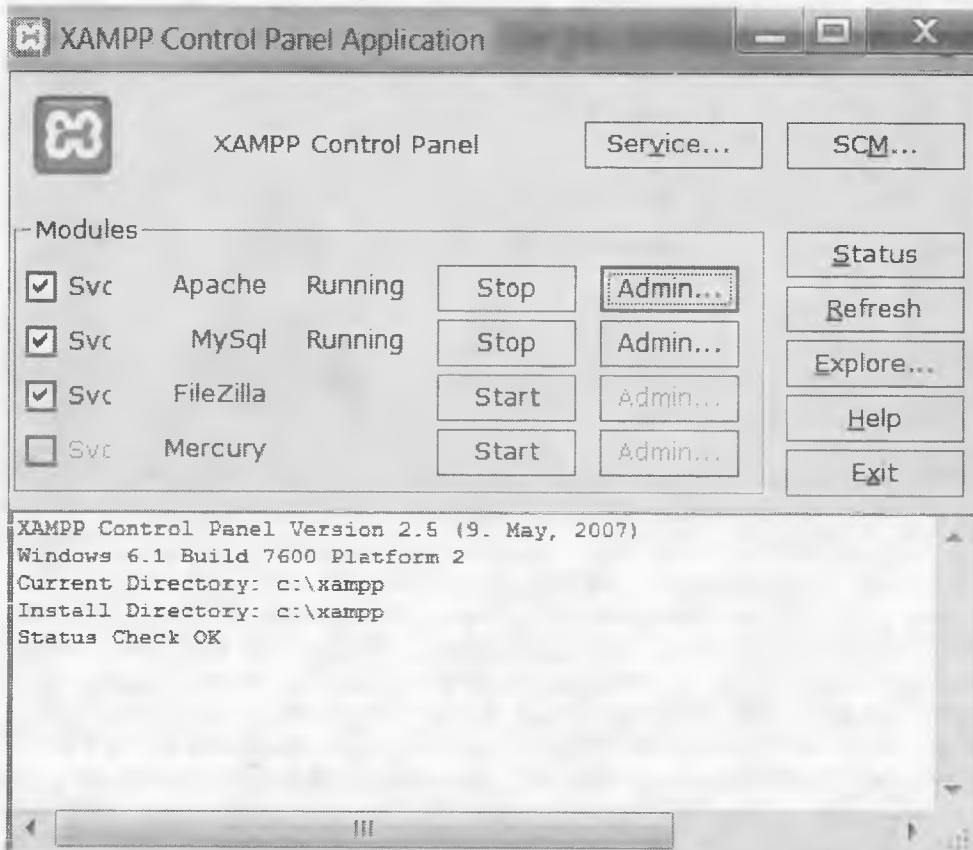


Рис. 9.4. Контрольная панель управления XAMPP

При нормальной работе службы серверов Apache и MySQL включены, сервер FileZilla отключен и при необходимости может быть подключен в соответствии с настройками, которые описаны на локальной странице веб-сервера Apache. Приступим к проверке работы служб. Для этого нажмем кнопку Admin для службы веб-сервера Apache и в открывшемся окне

выберем предложенный язык, на котором впоследствии будет выводиться информация (к сожалению, русского языка в локализации нет и, судя по всему, не предвидится), и перейдем к странице <http://localhost/xampp/index.php> (рис. 9.5).

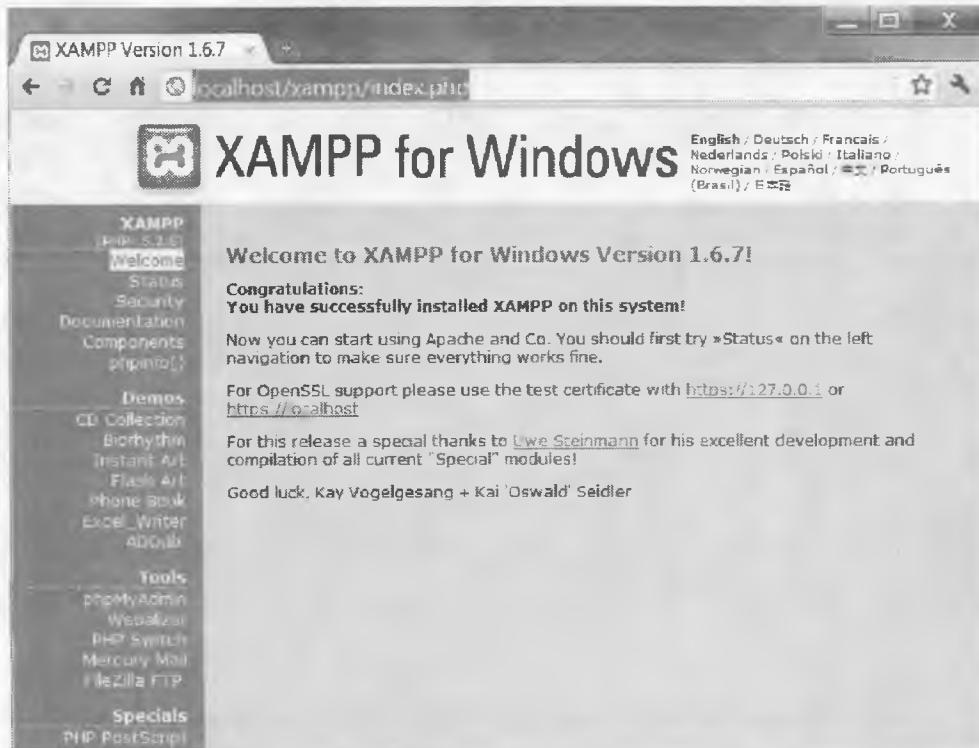


Рис. 9.5. Страница xampp/index.php на <http://localhost> при первом запуске программ

Обратите внимание на следующее: здесь указан хостинг <http://localhost>, или <http://127.0.0.1>, что то же самое. Это означает, что информацию браузеру необходимо искать не во внешней Сети, а на локальном компьютере. Таким образом, при обращении к сайту, организованному на данном локальном компьютере, следует обращаться через `localhost`.

Выберем строку `phpMyAdmin` из меню `Tools`, и откроется окно управления базой данных SQL-сервера (рис. 9.6).

Окно `phpMyAdmin` позволяет получить доступ к базе данных сайта и управлять ею. Однако сайта еще нет, поэтому нет и базы данных, соответственно, и управлять нечем. Скажу больше — Joomla! сама создаст свою базу данных для сайта и сама будет ее обслуживать, а нормальный пользователь просто не будет знать о ней ничего. Знакомство с этим окном обусловлено тем, что после создания и демонстрации сайта на локальном компьютере его придется переносить на веб-хостинг. А это возможно двумя способами. Первый

заключается в создании сайта заново на веб-хостинге, заодно можно потренироваться в освоении скользких мест. Второй подразумевает простое копирование содержимого базы данных сайта со своего компьютера и добавление его в базу данных на веб-хостинге. Подавляющее большинство платных хостингов поддерживает эту систему для управления клиентскими базами данных, соответственно, освоив работу с `phpMyAdmin` на своем компьютере, вы не растеряетесь и при работе с платным хостингом.

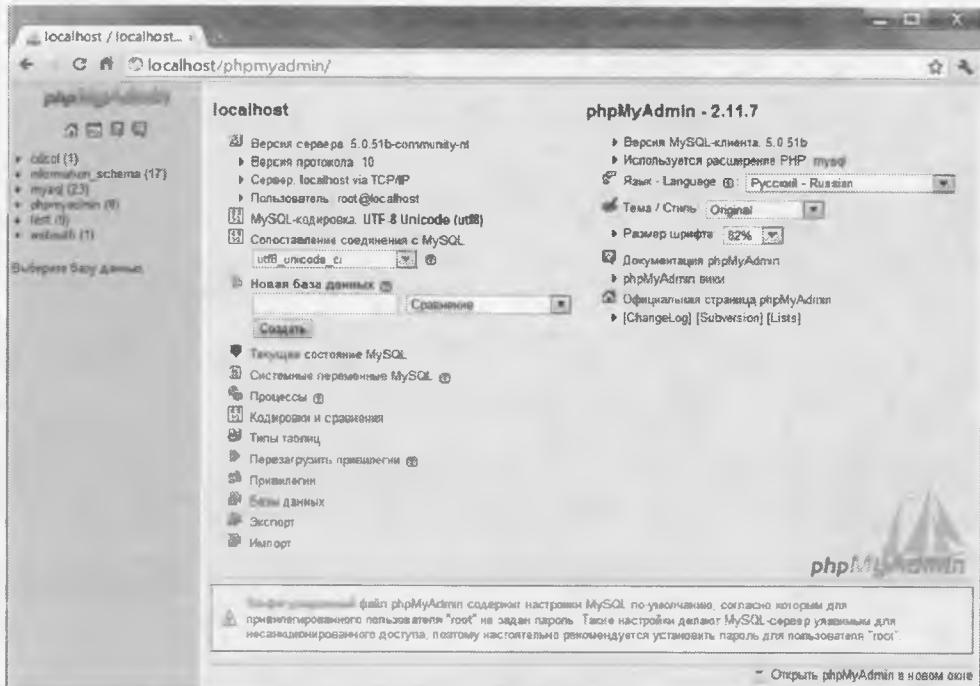


Рис. 9.6. Окно `phpMyAdmin`, предназначенное для управления базой данных с контентом сайта

Из настроек данного окна стоит запомнить следующее: SQL-сервер, к которому будет обращаться CMS Joomla!, называется `mysql`, пароль главного суперпользователя `root` не определен, локальный адрес расположения — `localhost`. Именно эти данные нам понадобятся при установке CMS Joomla!.

9.2. Установка и настройка

CMS Joomla!: где взять и куда положить

Саму систему управления содержимым можно получить с главного сайта проекта <http://www.joomla.org/> или с одного из его русскоязычных зеркал, например <http://www.joomla.ru/> (рис. 9.7).

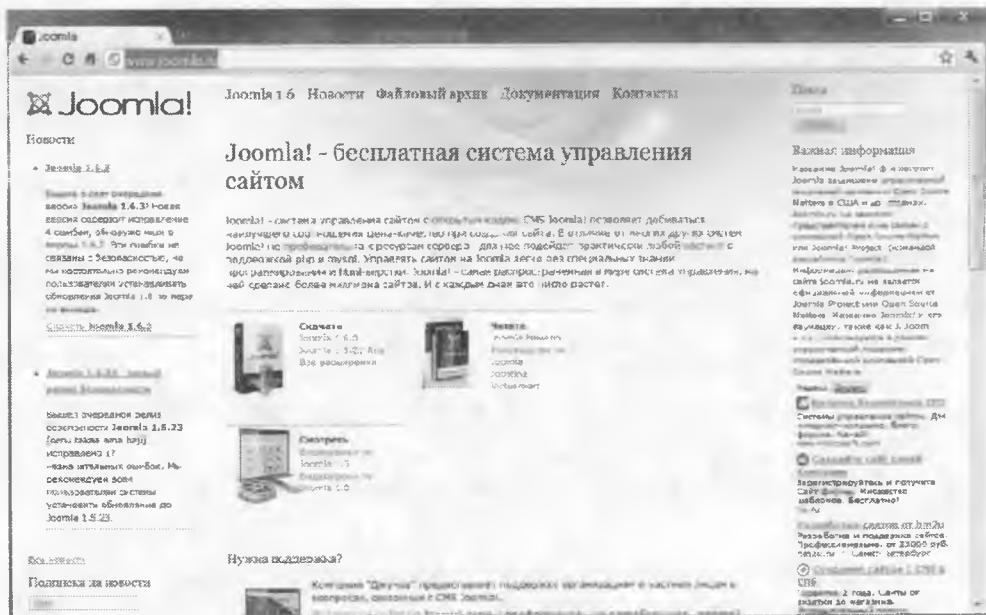


Рис. 9.7. Сайт <http://www.joomla.ru>, русское зеркало проекта <http://www.joomla.org/>

При выборе версий для скачивания указывайте стабильную сборку и не забудьте про обновления. Для Windows это будет ZIP-архив, а для Linux — любой из предложенных архивных файлов, кроме ZIP-архива Windows, разумеется. Имеет смысл обратить внимание на возможность локализации версии CMS Joomla!, на момент написания книги локализована только версия 1.5.22 RUS.

Для установки самой системы ее необходимо распаковать из архива и поместить в каталог, в котором она и должна размещаться. Если это будет внешний хостинг, то имя доступного каталога вам предоставит провайдер. В нашем случае при работе с XAMPP все создаваемые сайты хранятся в папках со своими именами в каталоге C:\xampp\htdocs, который был создан при установке. Именно в этот каталог, предварительно сохранив в нем папку saturnkom, распаковываем полученный архив с файлами (рис. 9.8).

Архивный файл можно распаковать и поместить в указанный каталог непосредственно, как и любой архив в операционной системе Windows. Однако можно потренироваться и, предварительно настроив FTP-сервер и клиент FileZilla, закачать распакованный архив в данный каталог, пропустив установку CMS Joomla! на удаленный веб-сервер.

Работа выполнена, теперь можно приступать к непосредственной установке и настройке сайта.

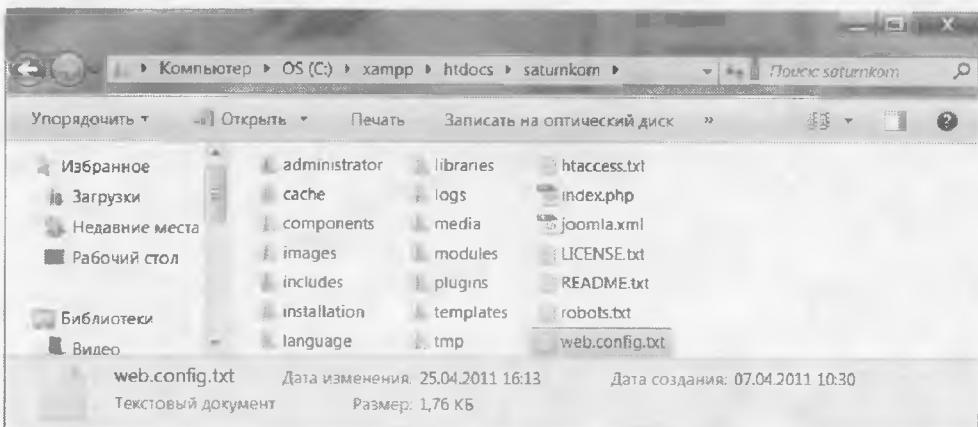


Рис. 9.8. Расположение распакованного архива CMS Joomla! на локальном компьютере при установке с использованием XAMPP

Первичная установка и настройка сайта в CMS Joomla! на локальном компьютере

Поскольку CMS Joomla! имеет сетевой интерфейс управления, то все работы по настройке самой программы и последующее управление и модификация сайта выполняются только через ее веб-интерфейс, с помощью браузера. Первоначальная настройка сайта состоит из нескольких шагов, в которых пользователь указывает те или иные параметры, в дальнейшем используемые системой по умолчанию. Если установку проводить не на локальный компьютер, а сразу на удаленный веб-сервер, то нужно подготовить FTP-клиент к работе (см. главу 3), так как в процессе установки Joomla! потребует определить права доступа к определенным каталогам, а впоследствии и удалить каталог `installation`.

Приступим. Наберем в адресной строке браузера адрес `http://localhost/saturnkom` и в открывшемся окне выполним **первый шаг** установки — выберем язык. В нашем случае задан русский язык (рис. 9.9). Теперь нажмем кнопку **Далее**.

Второй шаг — это проверка системы на соответствие требованиям CMS Joomla!. Внимательно отнеситесь к информации в окне браузера. В верхней области с параметрами не должно быть слова «нет» или любых красных надписей, иначе Joomla! не будет работать (рис. 9.10).

Поскольку установка проводится на локальный компьютер и XAMPP при этом настроен из-под операционной системы Windows, то в нашей установке упущен один главный шаг. Его сразу увидят пользователи Linux, а мы столкнемся с этой проблемой при размещении сайта на удаленном хостинге.

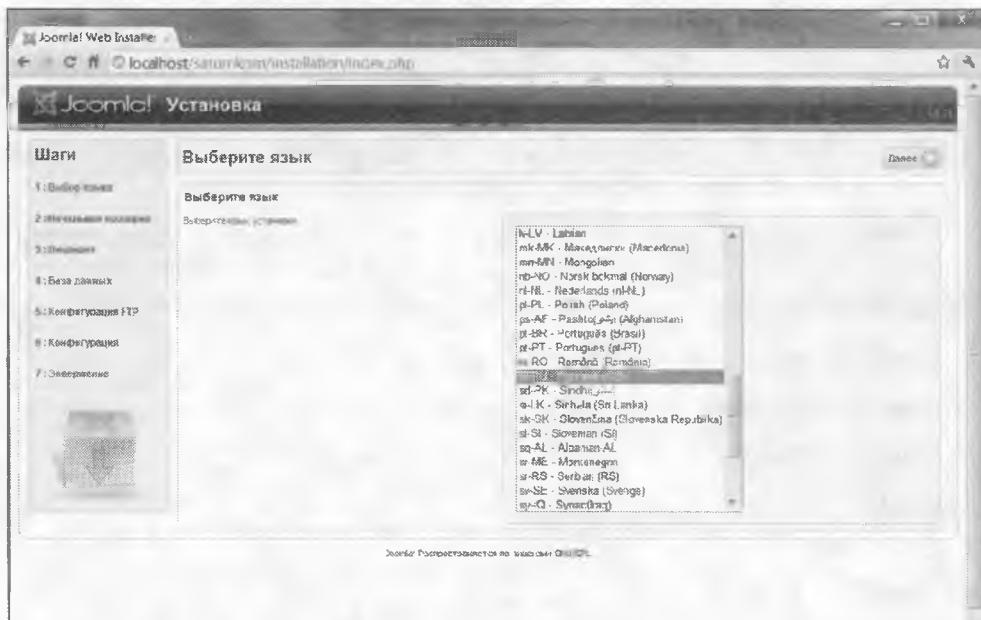


Рис. 9.9. Шаг первый – выбор языка интерфейса системы

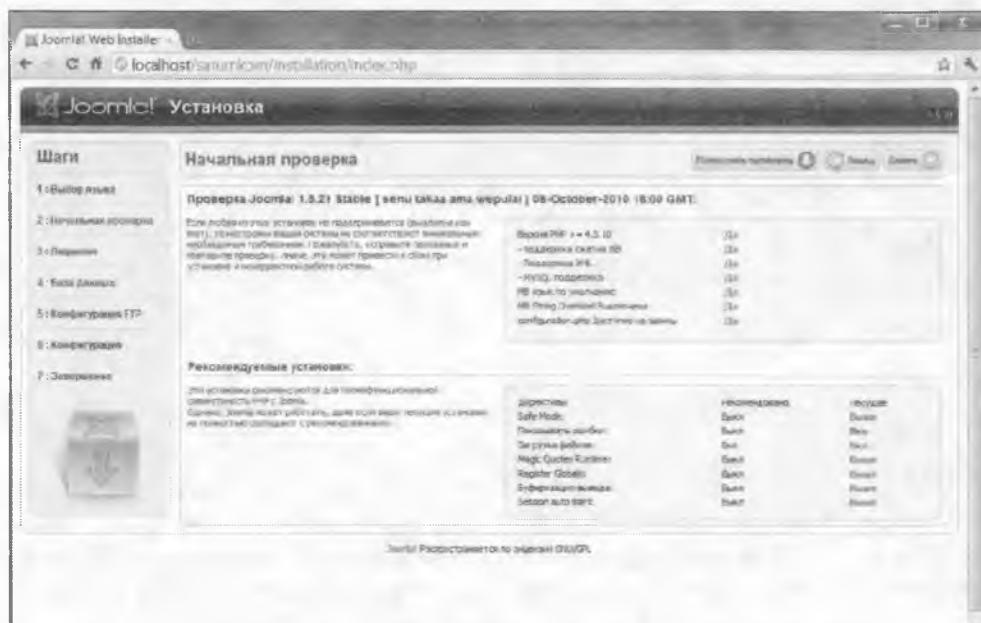


Рис. 9.10. Шаг второй – подтверждение работоспособности системы

Дело в том, что в подавляющем большинстве сетевых веб-серверы работают под управлением UNIX-подобных операционных систем, в которых политика безопасности несколько отличается от политики безопасности Windows. Для изменения либо просто открытия какого-либо файла в UNIX-подобных системах необходимо, чтобы у файла было настроено право доступа к нему, разрешающее или запрещающее какие-либо изменения.

Поэтому, чтобы избежать красных надписей, нужно предварительно задать права доступа 777 на изменения файлов для следующих каталогов:

- administrator/backups/;
- administrator/components/administrator/cache/;
- administrator/modules/;
- administrator/templates/;
- cache/;
- components/;
- images/;
- images/banners/;
- images/stories/;
- language/;
- plugins/;
- plugins/content/;
- plugins/editors/;
- plugins/editors-xtd/;
- plugins/search/;
- plugins/system/;
- media/;
- modules/;
- templates/;
- tmp/.

Кроме того, для файла configuration.php самой системы Joomla! необходимо установить права доступа 666, а для каталога, в котором будет размещена CMS, также должны быть заданы права доступа 777.

При этом обратите внимание, что файл конфигурации configuration.php создается при настройке CMS Joomla!, то есть сначала необходимо поместить распакованный архив с системой на удаленный веб-сервер.

Затем с помощью FTP-клиента, в нашем случае FileZilla, нужно устанавливать права доступа к каталогам — сначала к корневому, затем к остальным, указанным выше. Только после этого можно приступать к установке, в процессе которой по требованию системы необходимо определить права доступа к файлу конфигурации `configuration.php`. В результате выполнения описанного алгоритма мы и получим окно без красных записей, как на рис. 9.10.

Если при соблюдении вышеуказанной процедуры все-таки возникают трудности, то имеет смысл обратиться за помощью к вашему хостинг-провайдеру, так как глобальные настройки сервера персонально под требования вашего сайта может изменить только он. Но это бывает очень редко, почти никогда — CMS Joomla! без особых проблем ставится на любой хостинг, удовлетворяющий ее минимальным требованиям.

Третий шаг предполагает ознакомление с лицензией GNU/GPL, оригинальный текст которой приведен в окне браузера (рис. 9.11).



Рис. 9.11. Шаг третий — знакомство с лицензией GNU/GPL

Четвертый шаг позволит создать и настроить базу данных хранения контекстного содержимого сайта (рис. 9.12).

В появившемся окне необходимо заполнить следующие поля либо по аналогии с нашими — для локального компьютера, либо используя данные, которые были получены от хостинг-провайдера:

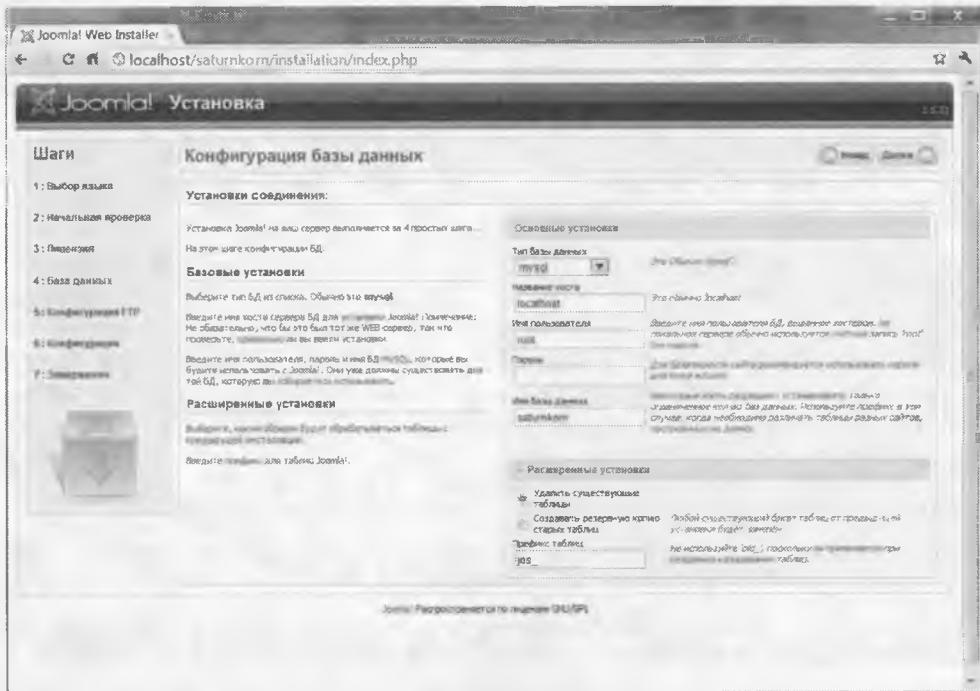


Рис. 9.12. Шаг четвертый — создание и конфигурация базы данных на SQL-сервере

- Тип базы данных — в нашем случае `mysql`. При размещении на удаленном веб-сервере — ранее полученные данные.
- Имя сервера базы данных — в нашем случае `localhost`. При удаленном размещении — данные, полученные от хостинг-провайдера.
- Имя пользователя — `root` в нашем случае или полученное от провайдера удаленного веб-сервера.
- Пароль — оставляем пустое поле в нашем случае, вводим полученный при регистрации пароль для доступа к базе данных.
- Имя базы данных — `saturnkom` для нас, или придумываем что-то для своего сайта на удаленном веб-сервере. При этом нужно помнить, что имена базы данных для одного и того же сайта как на локальном компьютере, так и на удаленном веб-сервере, должны совпадать. В ином случае при переносе сайта с локального компьютера на удаленный хост вы не сможете их синхронизировать и вам придется создавать сайт заново.
- Префикс таблицы — можно установить тот, что указан по умолчанию, но лучше его сменить. Обычно указывают три любые буквы и знак подчеркивания после них. Префикс предназначен для того, чтобы машина не путала таблицы одного сайта пользователя с таблицами другого сайта этого же

пользователя. Обычно это применяется тогда, когда провайдер отводит для пользователя только одну базу данных, а сайтов у него несколько. Таким образом, использование разных префиксов позволяет машине ориентироваться в базе данных и определять подачу контента для каждого сайта из своей отдельной таблицы, заданной своим собственным префиксом. Кроме того, изменение префикса способствует увеличению степени защиты базы данных от несанкционированного доступа. Обычный префикс базы данных стандартен и широко известен, измененный же префикс известен только вам, соответственно, злоумышленник, обращаясь по известному префиксу к таблицам вашей базы данных, их просто не найдет и не взломает.

■ **Действия с уже имеющимися таблицами** — я рекомендую установить первый флажок, так как во время учебы постоянно придется что-то добавлять, удалять, и в конце концов вы либо сами запутаетесь в переименованных таблицах, либо превысите дисковое пространство, отведенное веб-хостером под сайт.

Шаг пятый позволяет настроить FTP-клиент, присутствующий в CMS Joomla! (рис. 9.13).

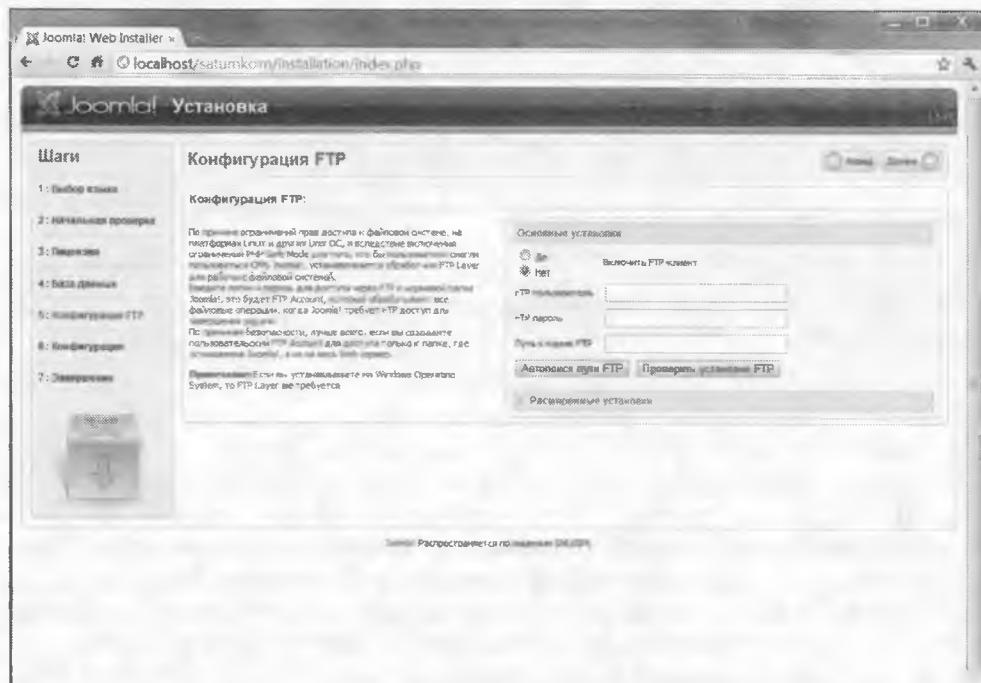


Рис. 9.13. Шаг пятый — настройка FTP-доступа к сайту встроенными средствами Joomla!

Поскольку мы работаем на локальном компьютере, то однозначно нужно установить переключатель в положение **Нет**. Для удаленного веб-сервера

можно определить включение FTP-доступа как Да, только не вводите пароль — система не рекомендует этого делать для предотвращения несанкционированного доступа к дисковому пространству, на котором размещен ваш сайт. Лучше запомнить или записать пароль и вводить его только по запросу системы, и только в том случае, если вы работаете в режиме административного редактирования.

Шестой шаг определяет название сайта и политику доступа к сайту администратора (рис. 9.14).

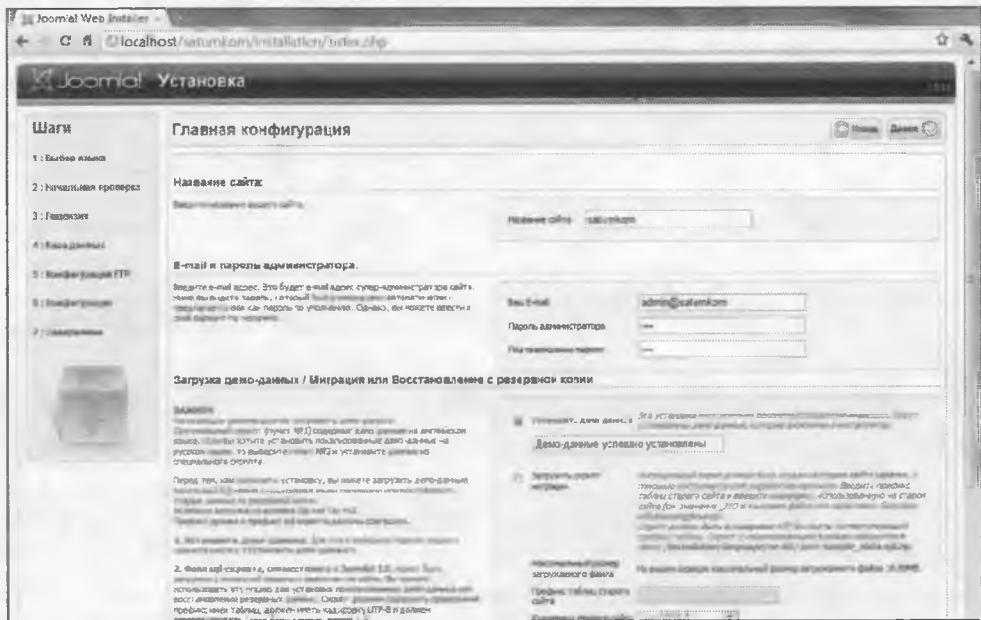


Рис. 9.14. Шаг шестой — определение конфигурации сайта

В основных настройках записывается имя сайта, электронный почтовый адрес администратора сайта и пароль доступа к сайту в режиме административного редактирования. В обязательном порядке следует установить переключатель в положение **Установить демо-данные** и проконтролировать получение ответа в виде надписи **Демо-данные успешно установлены**, как на рис. 9.14. При указании почтового ящика следует помнить, что он будет находиться в свободном доступе не только для гостей вашего сайта, но и для спам-роботов. По этой причине имеет смысл указывать не личный домашний электронный почтовый адрес, а почтовый адрес секретаря фирмы или человека, отвечающего за пересылку и сортировку почты внутри офиса, чтобы выполнялось первоначальное отсеивание спама.

На **седьмом шаге** установка завершается (рис. 9.15).

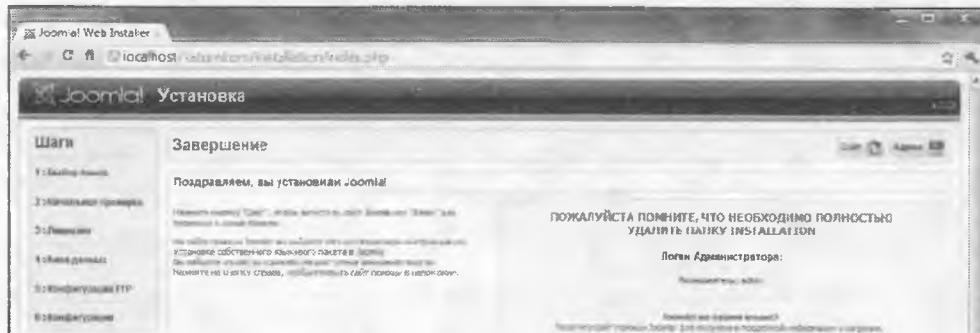


Рис. 9.15. Шаг седьмой – завершение установки CMS Joomla!

При завершении установки в обязательном порядке удалите каталог `installation`, иначе вы не сможете запустить систему. Если же система вдруг запустится, что может случиться только во время сбоя, то вы лишите свой сайт безопасности, оставляя все регистрационные данные для свободного доступа любому, кто случайно или умышленно забредет на выделенное вам дисковое пространство, будь то поисковый робот или специалист службы техподдержки. Удалив директорию `installation`, вы можете сразу просмотреть свой созданный сайт либо перейти к панели управления им, для чего достаточно щелкнуть на том или ином значке в правом верхнем углу экрана (рис. 9.16 и 9.17).



Рис. 9.16. Готовый сайт, сгенерированный CMS Joomla!

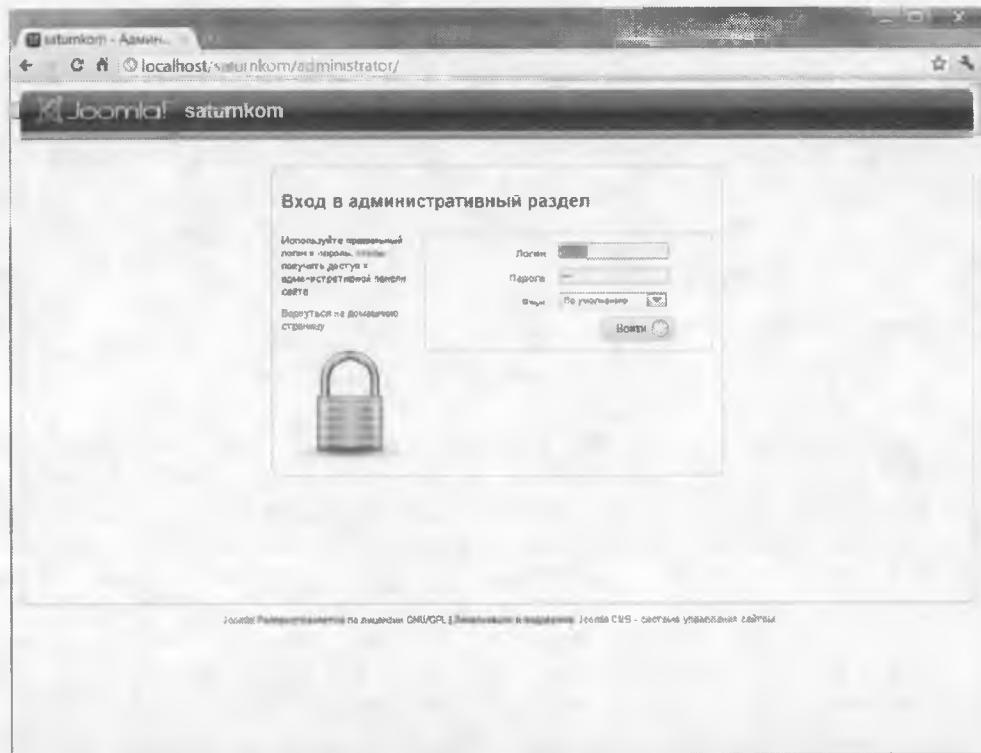


Рис. 9.17. Вход в панель управления сайтом на основе CMS Joomla!

Как видно, создание сайта на основе CMS Joomla! не представляет особых трудностей, главное — четко и последовательно выполнять инструкции, появляющиеся в каждом новом окне установки системы. Кроме того, перед этим необходимо получить от хостинг-провайдера исходные данные для их указания при регистрации. Заполнение сайта содержимым тоже не вызовет никаких проблем, только сначала обратимся к теории.

9.3. Структура

Каталоги CMS Joomla! и их содержимое

Прежде всего рассмотрим, в каких каталогах CMS Joomla! размещает информацию о сайте и какого характера эта информация. Поскольку сама система содержит более 3200 файлов, вкратце коснемся только основных каталогов структуры. Более того, вручную исправлять содержимое файлов и каталогов не только нет необходимости, но и зачастую противопоказано для целостности и безопасности самой системы, поскольку все управление сайтом и дополнительными элементами производится через веб-интерфейс.

Сама базовая структура CMS Joomla! после установки на локальный компьютер и удаления каталога `installation` представлена на рис. 9.18.

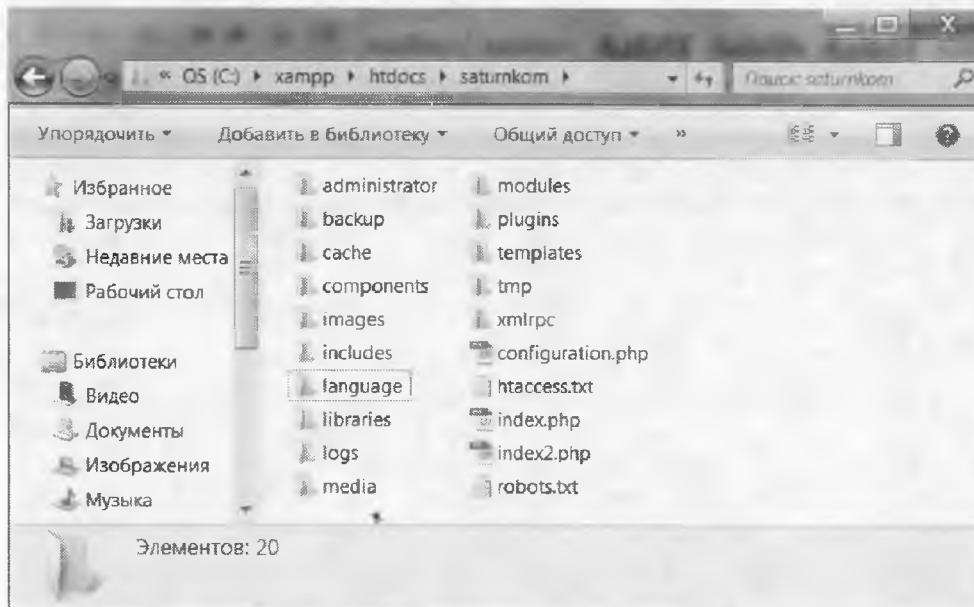


Рис. 9.18. Базовая структура системы управления содержимым Joomla!

Поскольку названия каталогов Joomla! тщательно продуманы, то при владении английским языком можно догадаться, что размещено в том или ином каталоге данного перечня.

- Корневой каталог CMS Joomla! – содержит главный файл `index.php`, запускающий установку системы после ее загрузки на веб-сервере. Если же файл `index.php` обнаружит существование файла `configuration.php` (с указанными в нем необходимыми для работы параметрами), то Joomla! будет запущена в работу. Таким образом, для переустановки Joomla! достаточно удалить файл `configuration.php` и запустить `index.php`, после чего и произойдет повторная установка Joomla!. Но при этом помните, что в корневом каталоге должен быть установлен каталог `installation`.
- `administrator` – содержит зеркало дерева каталогов, так как включает интерфейс администратора, то есть обычный сайт Joomla!. Таким образом, возможности интерфейса администратора могут быть увеличены путем установки соответствующих расширений, как и для самого сайта CMS Joomla!.
- `cache` – согласно имени, используется системой для кэширования веб-страниц, повышая производительность системы, так как хранит наиболее популярные страницы сайта, которые отправляются в браузер пользователя прямо из этого каталога, без обращения к PHP-транслятору и SQL-серверу.

- **components** — предназначен для хранения компонентов CMS Joomla!, которые могут отображаться на сайте с помощью связанных с ними модулей. По умолчанию CMS Joomla! содержит 11 компонентов, например компонент новостной ленты, голосований, поиска и т. п. Причем дополнительные необходимые компоненты можно скачать и установить.
- **images** — из названия каталога понятно, что он содержит все картинки, используемые в интерфейсе административного управления сайтом, а также картинки созданных статей, которые были загружены на сайт во время их написания. Подкаталог `\smilies` каталога `images` может содержать смайлики, подкаталог `\stories` — картинки, связанные со статьями, подкаталог `\banners` — файлы баннеров.
- **includes** — содержит PHP-файлы, которые используются для автоматического вложения контента в веб-страницы.
- **installation** — предназначен для начального этапа установки CMS Joomla!, поэтому содержит файлы, необходимые для начальной конфигурации системы при ее установке. Данный каталог должен быть удален сразу после установки, об этом вас попросит сама система на последней странице установки.
- **language** — содержит языковые файлы, используемые для перевода сайта на другие языки. Языковой файл — это обычный текстовый файл, сохраненный в кодировке UTF-8. Имена подкаталогов, содержащих файлы локализации для того или иного языка, стандартизированы. Первые две буквы — это код языка (по стандарту ISO-639-2), после тире следуют еще два символа — код страны (по стандарту ISO-3166). Таким образом, локализация для Великобритании хранится в каталоге с именем `en-GB`.
- **libraries** — предназначен для размещения основных библиотек функций, используемых CMS. Другими словами, Joomla!-сайт — по сути, это простое веб-приложение, которое применяют библиотеки этого каталога.
- **media** — предназначен для размещения мультимедиа-файлов, которыми вы насыщаете свой сайт.
- **modules** — содержит модули для отображения той или иной информации. Их можно вызывать из шаблонов, установленных в Joomla!, и, как выше было сказано, модули часто отображают информацию, предлагаемую компонентом.
- **plugins** — каталог для размещения плагинов, или расширений. Плагины предназначены для увеличения возможности CMS Joomla! на более низком уровне, чем компоненты, и управление ими производится через меню **Расширения** панели управления.
- **templates** — каталог для хранения папок с различными шаблонами, устанавливаемыми для сайта. Имена подкаталогов должны совпадать с названиями шаблонов.

- tmp — каталог для хранения временных файлов.
- xmlrpc — содержит мощную функцию CMS Joomla! — удаленный вызов процедур посредством XML. Это позволяет администратору с одного сервера управлять несколькими серверами, а посетителю дает возможность пользоваться специально подключенными модулями блогов.

Вот и вся основная информация о базовой структуре каталогов системы управления содержимым Joomla!. Она необходима пользователю лишь в крайних случаях, когда непосредственно требуется отредактировать соответствующий код с помощью простейшего текстового редактора или вставить графический объект, используя FTP-клиент для его размещения на веб-хостинге.

Планирование сайта на основе CMS Joomla! Разделы и категории, статический контент

Поскольку сайт предназначен для демонстрации определенного контента пользователям, необходимо понимать структуру размещения контента в системе Joomla! и учитывать это при планировании, а затем и наполнении сайтов.

В отличие от ранее рассмотренных методов построения сайтов классическим способом — путем написания HTML-кода веб-страниц и сохранения их в файлах и папках, имеющих почти бесконечную древовидную иерархию, ограниченную только длиной имени файла, сайт, созданный в CMS Joomla!, имеет абсолютно другую структуру.

Вместо неограниченной вложенности папок CMS Joomla! имеет строго закрепленный уровень иерархии, раз и навсегда установленный разработчиками. Дело в том, что в отличие от сайтов, написанных не в CMS, основным объектом в Joomla! является не файл, содержащий веб-страницу и ее контент, а статья. В свою очередь, статья собирается как мозаика из различных источников: текстовый контент — из таблиц базы данных SQL-сервера, графические объекты получают ссылки на заранее размещенные файлы в каталоге *images*, как и мультимедийные объекты. Далее эта смесь различных объектов проходит через шаблон и форматируется им в веб-страницу, содержащую только HTML-код веб-страницы. Если вдуматься в описанную модель, то видно, что нет не только необходимости, но и самой возможности структурировать файлы в иерархическую древовидную систему — самих файлов нет, они в виде веб-страницы появляются лишь в окне браузера посетителя. Структурирование контента в CMS Joomla! относится к статьям как к базовому элементу, содержащему текст веб-страницы.

Итак, все наполнение сайта, созданного с помощью CMS Joomla!, организовано в *двухуровневую* иерархию. Основа контента — это статья. Структура статей сходной тематической направленности представляет собой

категорию. В свою очередь, категории объединяются в *разделы*. Таким образом, любая статья относится к определенному разделу и входит в одну из его категорий. Принадлежность статьи к разделу и категории определяется специальными полями таблицы базы данных сайта, расположенных на SQL-сервере. Отсюда правило, которого необходимо четко придерживаться: до создания и расположения статьи на сайте необходимо определить раздел и категорию раздела, в который она будет входить. Если раздел или категория еще не созданы, то их нужно создать. Когда статья удаляется, она помещается в «корзину» до подтверждения полного удаления из таблицы базы данных. Так вот, если вы будете удалять категорию или раздел, в котором есть живые или помещенные в «корзину» статьи, то до полного удаления статей из таблицы базы данных вам не удастся удалить ни категорию, ни раздел. Кроме того, следует учитывать тот факт, что при создании элементов меню они привязываются к конкретным разделам, категориям или статьям.

Итак, иерархию CMS Joomla! можно представить следующим образом: Раздел ▶ Категория ▶ Статья.

Не бывает правил без исключений, и для CMS Joomla! это тоже справедливо — если нет возможности отнести статью к какому-либо разделу, а создавать его нет смысла, как, например, на веб-странице с информацией о контактах и проезде, то возможно создание некатегоризированных, или статических, статей. Такие статьи не относятся ни к одному из существующих разделов, но являются доступными, как обычные веб-страницы. Отсюда следующее предупреждение: не злоупотребляйте созданием некатегоризированных статей. Хотя они очень упрощают процесс бездумного создания непланируемого сайта, при увеличении количества информационных статей все труднее и труднее управлять содержимым ресурса. Рано или поздно вы не сможете контролировать потоки появляющихся веб-страниц, особенно когда авторами информационных статей будут не совсем компетентные люди. Поэтому важно планировать, планировать и еще раз планировать.

9.4. Первичные действия по созданию сайта на основе CMS

Администрирование сайта

Перед началом работы рекомендуется открыть на одной вкладке веб-обозревателя сам создаваемый сайт, набрав в адресной строке его адрес, в нашем случае <http://localhost/saturnkom/>. На второй вкладке следует открыть административную панель управления, в нашем случае нужно ввести адрес <http://localhost/saturnkom/administrator>. Не следует забывать, что при каждом сохранении изменений, выполняемых на административной панели браузера, необходимо обновлять просматриваемые веб-страницы на панели сайта в принудительном порядке, чтобы контролировать вносимые изменения.

При входе в панель администратора откроется следующее окно (рис. 9.19).

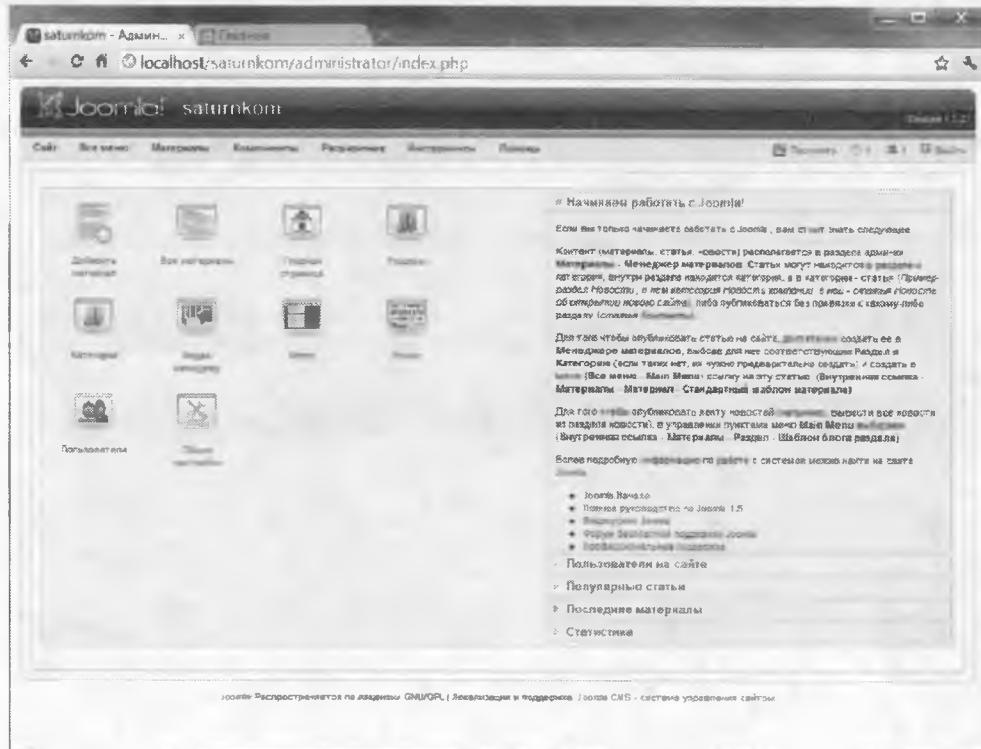


Рис. 9.19. Панель управления сайтом в режиме администратора в CMS Joomla!

В процессе установки CMS Joomla! на предпоследнем шаге мы задали установку демонстрационных данных. Исходя из наших установок, CMS Joomla! самостоятельно подключила все имеющиеся модули, создала пункты меню и наполнила страницы контентом-образцом, благодаря чему получился готовый сайт (см. рис. 9.16). Не требуется заново создавать всю структуру, необходимо лишь использовать под наши нужды то, что уже создано.

Для заполнения страниц сайта нужным содержимым предварительно следует удалить демонстрационный контент. При этом нужно помнить, что удаление всего начинается с полного удаления статьи. Если вместо переключателя у названия статьи стоит замок, то статья в данный момент открыта для редактирования, и перед удалением ее необходимо закрыть. Если вы уверены в том, что со статьей в данный момент никто не работает и ее попросту забыли закрыть при редактировании в последний раз, то можно сбросить блокировку всех открытых элементов сайта, выполнив пункт меню **Инструменты** ▶ **Сброс блокировок**.

Как видно из рис. 9.19, административная панель управления содержит как пункты меню управления элементами сайта, так и дублирующие их значки.

Для управления материалами сайта нужно воспользоваться менеджером материалов. Для его вызова выполните команду Материалы ▶ Менеджер материалов либо просто щелкните на значке Все материалы. В результате откроется окно, в котором и отображаются все находящиеся на сайте демонстрационные статьи (рис. 9.20). Поскольку статей очень много, то они выводятся по оконно, по умолчанию по 20 строк в окне. Для перехода между окнами пользуйтесь кнопками перехода, которые расположены в нижней части окна.

Рис. 9.20. Окно менеджера материалов со статьями, установленными CMS Joomla! по умолчанию

Удалять статьи можно по одной, а можно все сразу, для чего необходимо установить флажок в головке таблицы материалов. После выделения всех статей щелкните на значке В корзину и повторите это столько раз, сколько нужно для полной очистки сайта от демонстрационных статей.

После удаления статей в корзину следует перейти в окно менеджера корзины и повторить приведенный выше алгоритм действий, полностью удаляя статьи уже из корзины (рис. 9.21).



Менеджер корзины: [Материалы]

Фильтр

Помощь | Помощь

Заголовок № Родитель Баннер

1 № 1 без категории

2 Каталог

3 Контакт

4 Сервис. ОСТ

5 Сайт группы

6 Новый интернет-магазин

7 сайт по полной работе на joomla

8 Joomla! Commerce

9 Авт-маник

10 Большой интернет-магазин

11 Гольф-8 - открытие 2009

12 Из него отпадаете стоимость сайта

Рис. 9.21. Окно менеджера корзины

Последовательно выполняя команду **Материалы** ▶ **Менеджер категорий и Материалы** ▶ **Менеджер разделов**, нужно удалить сначала категории, а затем и разделы сайта. После удаления можно перейти на главную страницу сайта, что открыта на соседней вкладке веб-обозревателя, и, обновив ее, убедиться, что остался только шаблон, а демонстрационные статьи удалены, то есть сайт чист и готов к работе.

Создание новых разделов и категорий

Для создания новых разделов необходимо вернуться на вкладку администратора и открыть в ней окно менеджера разделов. Щелкните на значке **Создать** в правом верхнем углу экрана. В зависимости от локализации версии CMS Joomla! значки могут иметь различное название, например **Новый**, поэтому вникайте в смысл надписей.

При создании раздела активизируется окно **Раздел**, в котором требуется указать параметры создаваемого раздела (рис. 9.22).

Определяем параметры полей.

- Содержимое поля **Заголовок** будет отображаться в меню, поэтому не должно быть слишком длинным, иначе не поместится в отведенное ему место и будет просто обрезано. В нашем случае это **Компания "Сатурн"**.
- Содержимое поля **Псевдоним** будет отображаться только при выборе раздела, то есть когда пользователь в него уже вошел, поэтому таких жестких ограничений по длине к нему не предъявляют. В нашем случае псевдоним звучит как **Компания "Сатурн". Философия и миссия компании**.

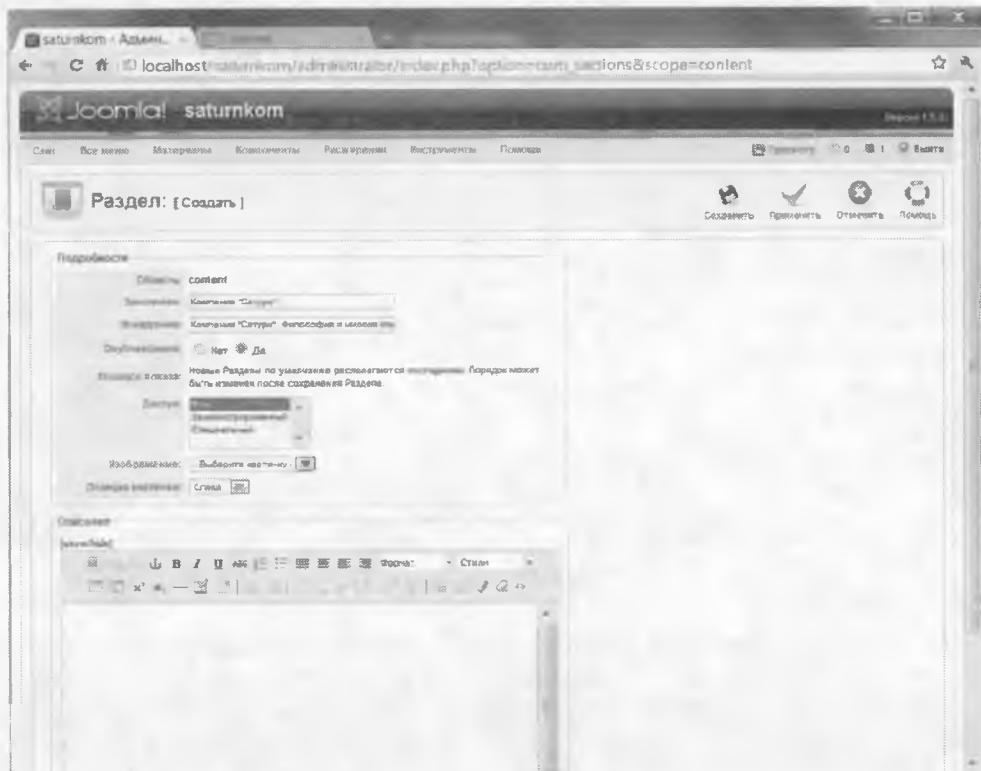


Рис. 9.22. Окно Раздел для указания параметров создаваемого раздела

- Ключ Опубликовано позволяет разрешить или запретить публикацию статей всего раздела вне зависимости от их количества.
- Поле Порядок показа будет активно только после создания раздела, и в нем можно указать, в какой последовательности располагаются создаваемые разделы, и при необходимости изменить порядок отображения разделов.
- Поле Доступ определяет категорию пользователей, допущенных к материалам данного раздела. Оно также является очень мощным инструментом формирования аудитории сайта.
- Окно Изображение позволяет выбрать значок для раздела. Значки расположены в каталоге image\stories, и при необходимости в дополнение к установленным по умолчанию можно добавить свои дополнительные графические файлы со значками.
- Поле Позиция картинки определяет позицию значка раздела.
- Большое текстовое поле Описание содержит более подробное описание раздела, чем окно Псевдоним. Имеет смысл заполнить его информацией, показывающей пользователям, какие статьи им ожидать в данном разделе.

После заполнения параметров раздела нужно щелкнуть на значке Сохранить. Окно Раздел, создание закроется, и вы вернетесь в окно менеджера разделов (рис. 9.23). Обратите внимание, что раздел Компания "Сатурн" создан, опубликован и доступен для чтения для всех посетителей вашего сайта.

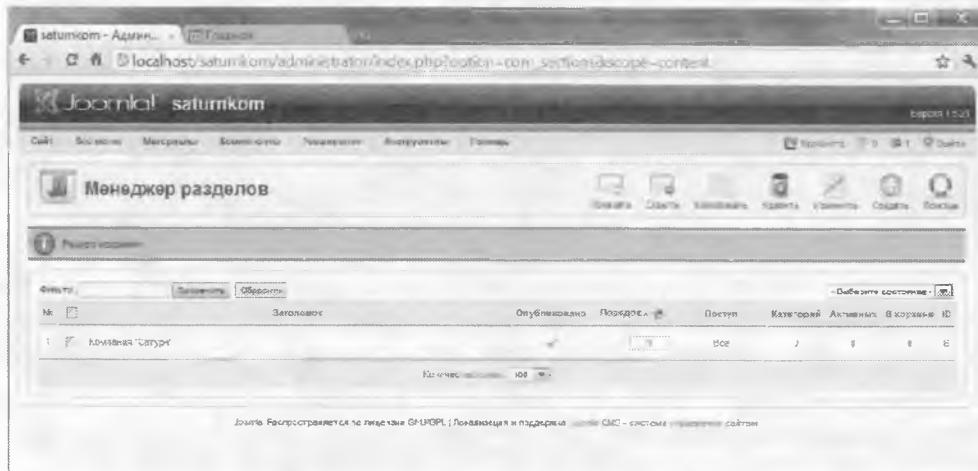


Рис. 9.23. Окно менеджера разделов с вновь созданным разделом Компания "Сатурн"

Для создания категории необходимо выбрать пункт меню Материалы ▶ Менеджер категорий. В появившемся окне перед созданием новой категории нужно обозначить, к какому разделу она относится и стоит ли ее опубликовывать (рис. 9.24). В нашем случае раздел указан как Компания "Сатурн" и категория опубликована. Затем следует щелкнуть на значке Создать.

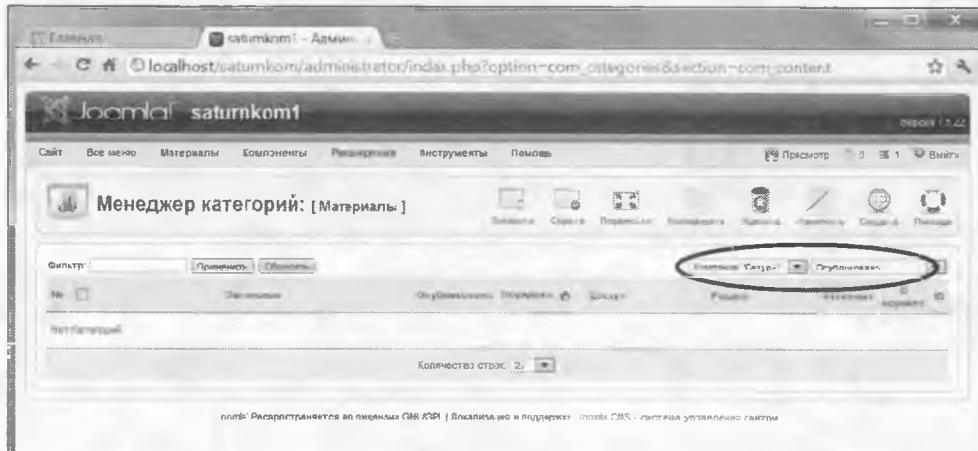


Рис. 9.24. Окно менеджера категорий с назвланием раздела и указанием на публикацию

Окно создания категории аналогично окну создания разделов — поля имеют такое же название и смысл, единственное отличие состоит в том, что поле **Псевдоним** можно не заполнять, это сделает сама CMS Joomla! в автоматическом режиме. После создания раздела и категории можно приступать к созданию статей и их публикации.

Создание и публикация категоризированных статей в CMS Joomla!

Прежде чем приступать к добавлению статей в созданный ранее раздел, определимся с теми текстовыми редакторами, которые предлагает CMS Joomla! для редактирования статей. Чтобы выбрать редактор, устанавливаемый по умолчанию, воспользуйтесь меню **Сайт** ▶ **Общие настройки** (рис. 9.25). В списке представлены три варианта ввода текста: можно выбрать один из двух текстовых редакторов или указать, что работа будет проходить без такового. Во втором случае в тексте статьи можно помещать просто HTML-код веб страницы, который находится между тегами `<body>` и `</body>`.

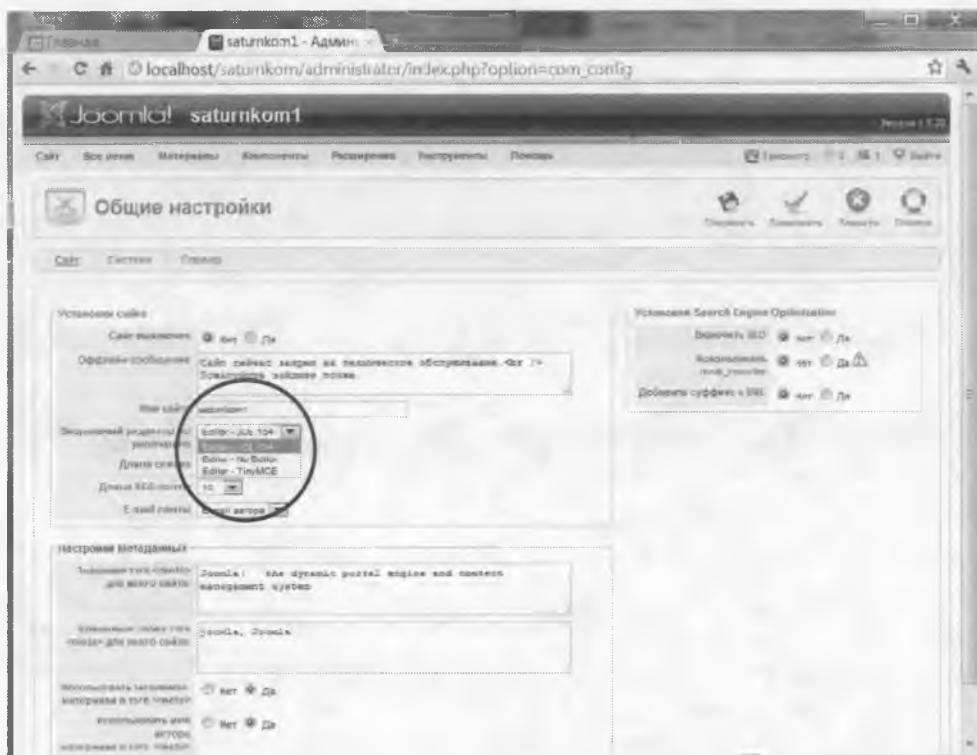


Рис. 9.25. Окно Общие настройки со списком выбора текстового редактора для создания статей и его установки по умолчанию

Попробуйте любой из предложенных текстовых редакторов, ведь при необходимости их настройку всегда можно изменить. После выбора редактора и выхода со страницы в обязательном порядке не забывайте щелкать на значке **Сохранить**, иначе изменения в состоянии страницы не будут внесены и окно окажется под замком, то есть без прямого доступа. В результате придется сбрасывать блокировку уже для всех статей, несмотря на то что с ними могут в данный момент работать пользователи.

Для создания новой статьи по аналогии с описанным выше нужно обратиться к пункту меню **Материалы** ▶ **Менеджер материалов** либо просто щелкнуть на значке **Добавить материал**, который находится на главной странице панели управления CMS Joomla!.

При создании статьи, а также при ее редактировании можно определять настройки в следующих полях.

- **Заголовок** — отображается на странице со статьей и в пункте меню.
- **Псевдоним** — альтернативный заголовок для статьи, который используется так называемыми сценариями дружественного поиска. Если поле **Псевдоним** оставлять пустым, то CMS Joomla! в автоматическом режиме установит значение, предпочтительное для поисковых роботов.
- **Раздел** и **Категория** — определяют принадлежность данной статьи к соответствующему разделу и категории соответственно.
- **Опубликовано** — определяет разрешение публикации данной статьи.
- **На главной** — дает указание, размещать или нет данную статью на главной странице сайта.

В поле **Параметры Статьи** можно оставить имеющиеся по умолчанию записи или указать информацию в следующих полях (рис. 9.26).

- **Автор** — содержит информацию о зарегистрированном авторе, в нашем случае автор пока один — **Administrator**.
- **Псевдоним автора** — содержит псевдоним, который будет отображаться, если задано показывать имя автора.
- **Доступ** — определяет доступ к данной статье различным категориям пользователей.
- **Дата создания**, **Дата публикации** и **Истекает** — позволяют изменять параметры времени создания, публикации и удаления данной статьи. Благодаря этим параметрам статьи могут быть созданы заблаговременно, а появляться и удаляться на веб-страницах сайта будут в строго определенное время. Например, еженедельная рубрика статей, посвященная рыбной ловле, обновляется по четвергам, а посвященная охоте — по вторникам.

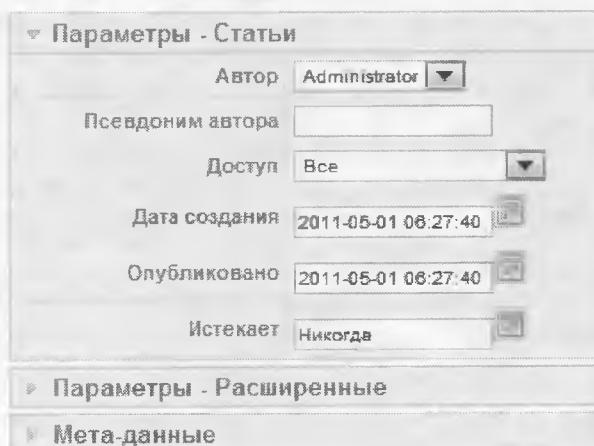


Рис. 9.26. Окно Параметры Статьи

В окне Параметры Расширенные настраиваются расширенные параметры статьи, которые будут выведены по требованию пользователя при нажатии кнопки Подробно внизу текста статьи (рис. 9.27). Обычно достаточно значений, установленных по умолчанию, — Использовать глобальные, но при необходимости параметры можно изменить.

- **Заголовок** — отображает или скрывает заголовок статьи.
- **Заголовок как ссылка** — разрешает использовать заголовок как ссылку. При активизации ссылки осуществляется переход к подробной части статьи.
- **Вступительный текст** — отображает вступительный текст выше самой статьи. Вступительный текст — это анонс статьи, который выводится в списке статей при просмотре содержимого категорий совместно с надписью ссылки Подробнее, при выборе которой будет выведен полный текст статьи. Для отделения вступительного текста от остального текста статьи необходимо нажать кнопку **Разрыв страницы**, которая находится под окном статьи, и в открывшемся диалоговом окне указать требуемые параметры.
- **Название раздела** и **Название раздела как ссылка**, а также **Название категории** и **Название категории как ссылка** — определяют возможность указания при просмотре статей на раздел и категорию, в которых находится статья. Если они определены как ссылки, то пользователь при их задании может увидеть перечень всех статей, входящих в данный раздел или категорию, в зависимости от того, какая из ссылок была выбрана.
- **Рейтинг** — показывает рейтинг статьи и определяется самими посетителями сайта.
- **Имя автора** — разрешает или запрещает демонстрацию имени автора.
- **Дата и время создания** — разрешает или запрещает демонстрацию времени создания статьи.

- **Дата и время последнего изменения** — также разрешает или запрещает демонстрацию времени изменения статьи.
- **Иконка PDF** — показывает или скрывает значок PDF-версии статьи. При щелчке на нем пользователю предоставляется статья в PDF-формате.
- **Иконка печати** — показывает или нет значок печати для статьи.
- **Иконка E-mail** — также выводит или нет значок запуска электронной почты.
- **Язык материала** — определяет язык, на котором написана статья, на основании установленных ранее языковых пакетов.
- **Ключевая сноска** — в данное поле можно ввести текст, который будет использоваться как перекрестная ссылка на эту статью из других статей в пределах сайта.
- **Альтернативный текст** — задает текст для ссылки **Подробнее**, тем самым указывая, какой текст будет отображаться вместо слова **Подробнее** при разделении статьи.

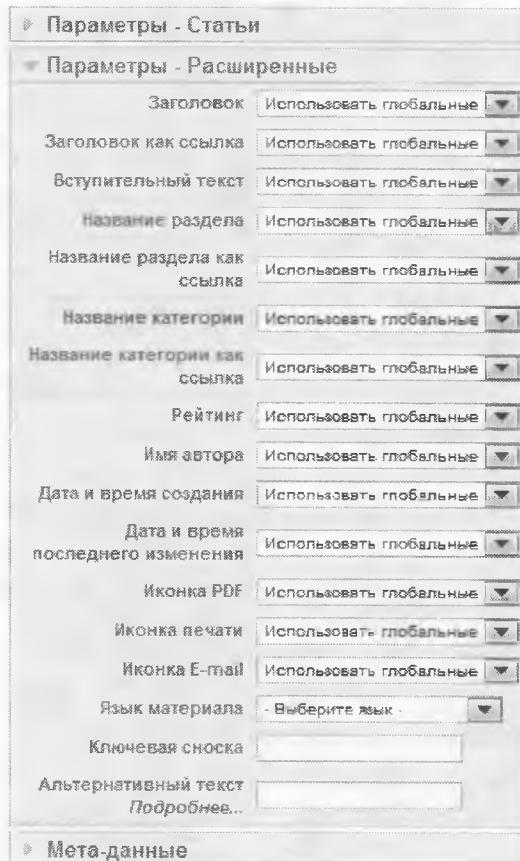


Рис. 9.27. Окно Параметры Расширенные

Напоследок рассмотрим окно **Мета-данные** (рис. 9.28). Здесь прописываются параметры, позволяющие воздействовать на поисковые машины с целью увеличения релевантности статьи на соответствие определенному запросу. Информация, составляющая область метаданных и ограниченная тегами `<meta></meta>`, не видна пользователям, но превосходно различается поисковыми роботами. Это можно использовать для ее индексации в базе данных поисковых машин. Предлагается заполнить следующие поля.

- **Описание** — задает описание статьи, которое отображается поисковыми машинами сразу после заголовка статьи.
- **Ключевые слова** — список ключевых слов, характеризующих содержимое статьи.
- **Параметр Robots** — задает параметры, согласно которым выполняется индексирование статьи поисковыми системами.
- **Автор** — поле для указания имени автора статьи.

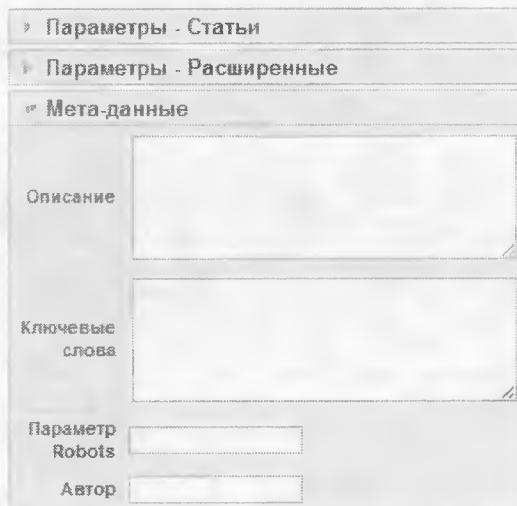


Рис. 9.28. Окно Мета-данные

Окно **Мета-данные** рекомендуется заполнять сразу при создании статьи, поскольку, как показывает практика, после написания статьи о ней все забывают, и метаданные впоследствии не редактируются.

После краткого знакомства с необходимыми параметрами, которые следует указывать при создании статьи, можно приступить и к самому созданию.

Нажав кнопку **Добавить материал** на панели управления либо выбрав пункт меню **Материалы** ▶ **Менеджер материалов** ▶ **Создать**, откроем окно создания статьи (рис. 9.29) и заполним ключевые поля согласно изложенному материалу.

Параметры статьи оставляем установленными по умолчанию.

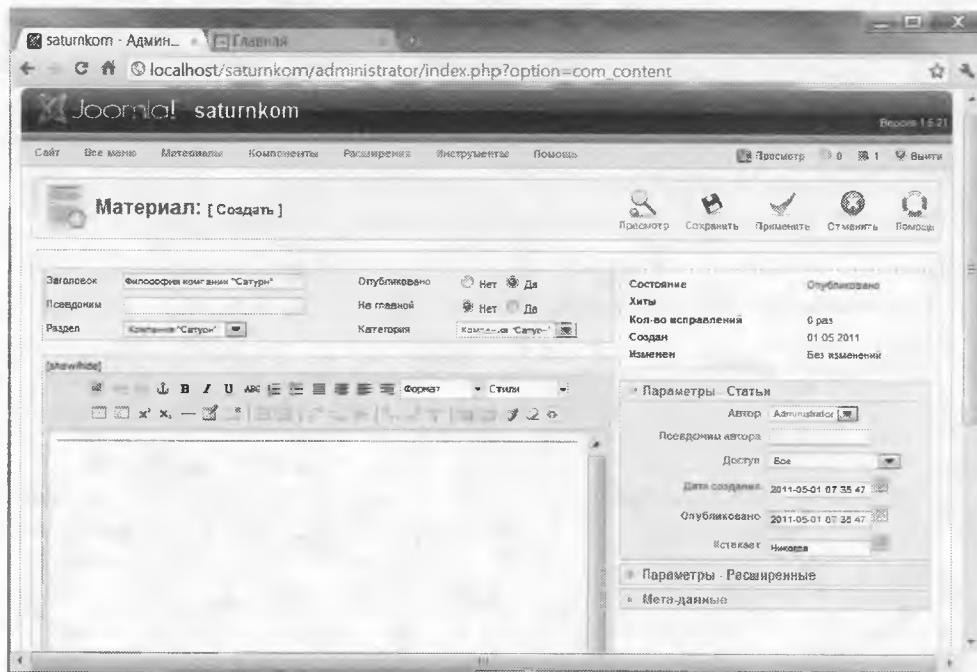


Рис. 9.29. Окно для создания статьи

Расширенные параметры для статьи можно заполнить следующим образом.

- Заголовок — Да.
- Заголовок как ссылка — Да.
- Вступительный текст — Показать.
- Название раздела — Показать.
- Название раздела как ссылка — Да.
- Название категории — Показать.
- Название категории как ссылка — Да.
- Рейтинг — Скрыть. Для данной статьи понятие рейтинга просто бессмысленно.
- Имя автора — Показать.
- Дата и время создания — Скрыть.
- Дата и время последнего изменения — Скрыть.
- Иконка PDF — Скрыть.
- Иконка печати — Скрыть.

- Иконка E-mail — Скрыть.
- Язык материала — русский.
- Ключевая сноска — философия.
- Альтернативный текст — пусто.

Поля окна **Мета-данные** заполним следующим образом.

- **Описание** — статья содержит материал, описывающий мировоззрение сотрудников компании.
- **Ключевые слова** — философия, космос, Сатурн, простор, звезды, пространство, мечта и пр.

В статье рассказывается о развитии компании, и она будет достаточно объемной, поэтому разделим ее на вступительный текст и основное содержание. Обычно для этого достаточно нажать кнопку **Разрыв страницы**, расположенную внизу окна, и заполнить требуемые поля в открывшемся окне (рис. 9.30). Но в разных версиях CMS Joomla! возможны различные алгоритмы, поэтому не теряйтесь, а просто посмотрите, где бы это могло быть и как это можно реализовать.

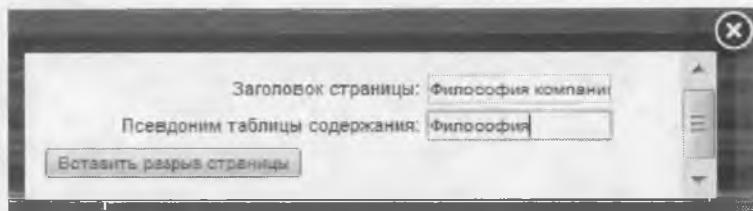


Рис. 9.30. Окно, которое открывается при разбиении статьи на вступительный текст и основную часть

Теперь можно начать вводить текст в поле статьи. В область вступительного текста введем нашу ключевую фразу:

Компания "Сатурн" Открытый космос у Ваших ног

В области **Подробнее** разместим такой текст:

Философия компании "Сатурн" лежит в плоскости мечты любого жителя Земли. Просторы открытого космоса, безбрежное пространство, открытое только нашему внутреннему взору, поскольку охватить Вселенную взглядом невозможно. Холодные мерцающие звезды, бисером устилающие черный бархат вечного ночного неба, и золотые кольца Сатурна, как вечный подарок-загадка...

При вводе текста его можно форматировать с использованием инструментов встроенного текстового редактора. В результате окно создания статьи будет выглядеть, как на рис. 9.31.

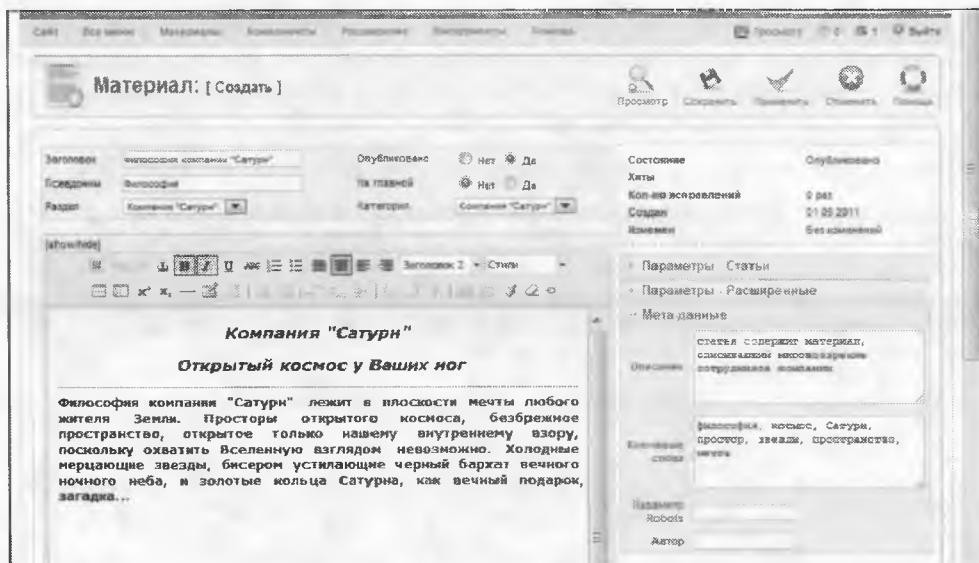


Рис. 9.31. Окно создания статьи с текстом

Теперь можно нажать кнопку **Применить**, расположенную в правом верхнем углу экрана, перейти на вкладку с видом веб-страниц и нажать кнопку **Обновить**, а затем перейти по появившейся ссылке **Философия компании "Сатурн"**. Как видно на рис. 9.32, созданная статья прекрасно располагается на выделенной для нее веб-странице, а гиперссылки позволяют переходить между статьями созданного раздела.

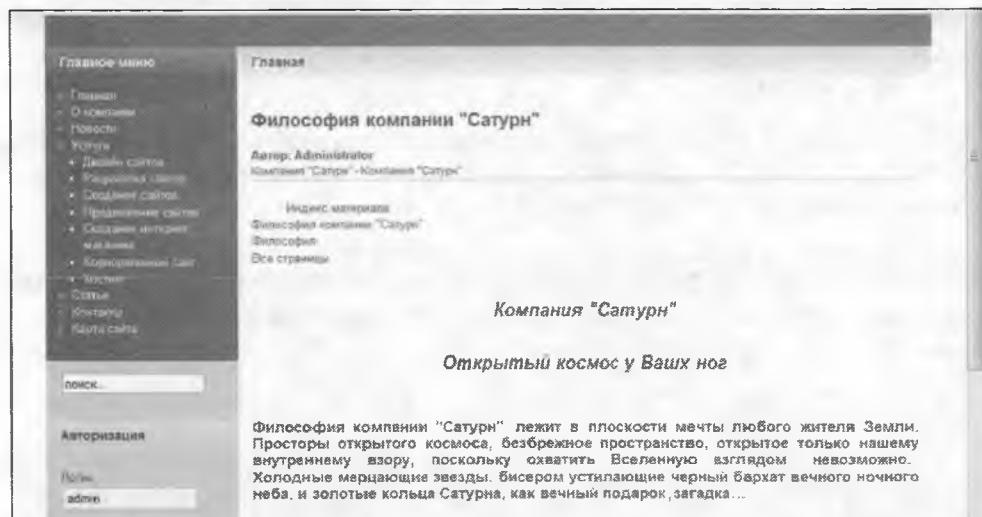


Рис. 9.32. Опубликованная статья «Философия компании „Сатурн“»

Поскольку нас устраивает правильное отображение созданной статьи, можно вернуться в панель административного управления CMS Joomla! и щелкнуть на значке **Сохранить**, закрывая статью для редактирования.

При сохранении статьи будет выполнен переход в окно менеджера материалов, где можно увидеть краткую информацию о статье (рис. 9.33).

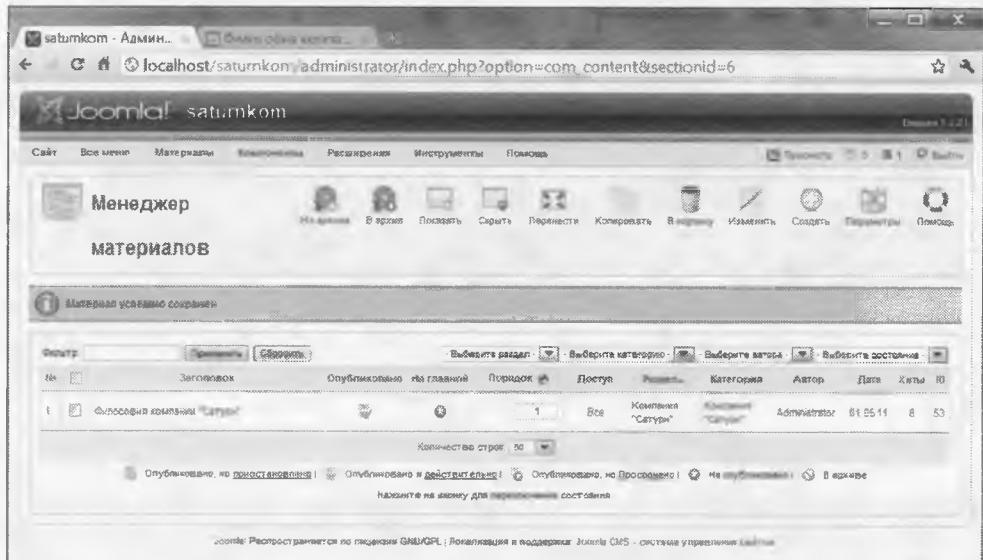


Рис. 9.33. Окно менеджера материалов с указанной статьей «Философия компании „Сатурн“»

Создание и публикация некатегоризированных статей в CMS Joomla!

Для создания некатегоризированных, или статических, страниц сайта в CMS Joomla! необходимо *не* указывать раздел и категорию при создании статьи в окне менеджера создания материалов.

Создадим новую статическую статью, щелкнув на значке **Создать** в правом верхнем углу (см. рис. 9.33). Заполним статью, которая будет вступительной для нашего сайта, но не заполняя поля **Раздел** и **Категория** и обозначив их как **Не указано**. Текст статьи достаточно прост:

Компания "Сатурн" рада приветствовать на страницах своего сайта наших новых гостей и старых друзей

При этом следует указать, что статью нужно опубликовать на главной странице сайта, и в расширенных параметрах страницы закрыть и запретить отображение дополнительной информации. Таким образом, окно с вносимыми изменениями статической страницы может выглядеть, как на рис. 9.34.

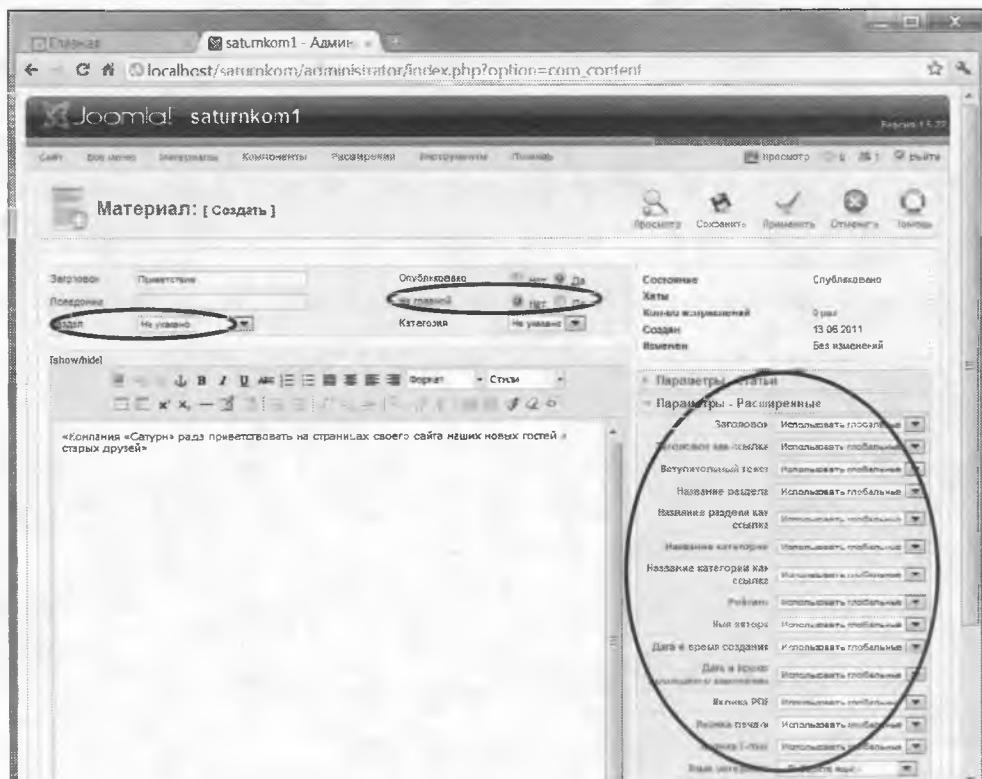


Рис. 9.34. Окно создания материала с областями, которые требуется изменить при создании статической страницы, опубликованной на главной странице сайта

Отредактировав и отформатировав текст статьи с помощью инструментов текстового редактора, щелкаем на значке **Применить**, переходим на вкладку браузера, содержащего веб-страницы сайта, и обновляем главную страницу. Как видно из рис. 9.35, наша некатегоризированная статья была опубликована на главной странице, чего мы и добивались.

Не правда ли, прекрасно получается? Только почему-то на главной странице очень много лишней информации, не относящейся к компании «Сатурн». Как ее изменить или удалить со страниц сайта? Об этом мы расскажем чуть ниже.

Пока же необходимо вернуться на вкладку административного управления сайтом и щелкнуть кнопкой мыши на значке **Сохранить**, тем самым завершив редактирование статьи. Теперь в окне менеджера материалов мы имеем уже две созданные статьи, а если выбрать пункт меню **Материалы** → **Менеджер Главной страницы** либо щелкнуть на значке **Главная страница** в панели управления, то в открывшемся окне менеджера (рис. 9.36) можно увидеть все статьи, размещенные на главной странице сайта. У нас пока имеется только одна статья.

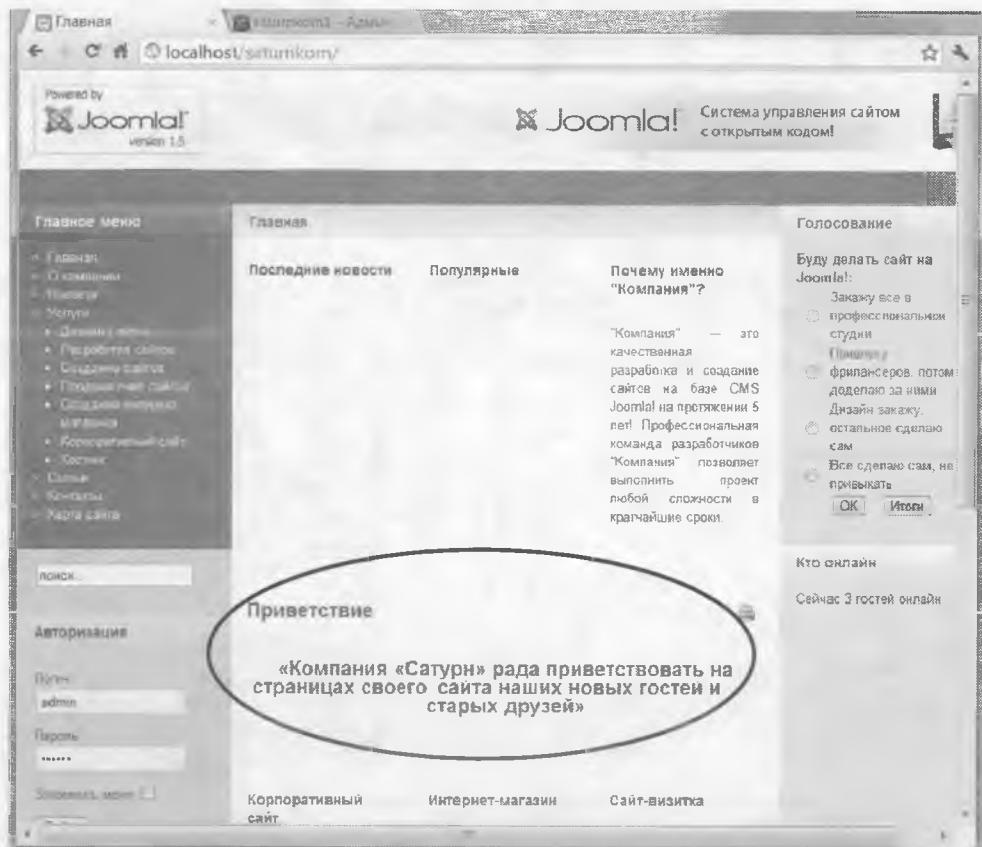


Рис. 9.35. Публикация некатегоризированной статьи на главной странице сайта

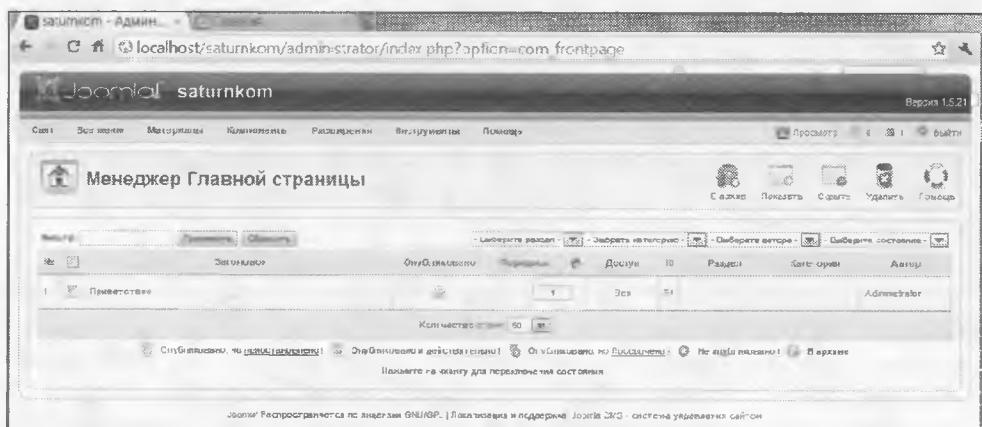


Рис. 9.36. Окно менеджера главной страницы с указанием на размещение статьи Приветствие

Неоднократно повторив указанные выше действия, создадим статьи, соответствующие двум другим страницам, разнеся их в специально созданные разделы. Таким образом, окно менеджера материалов будет отображать подготовленные для дальнейших действий статьи (рис. 9.37).

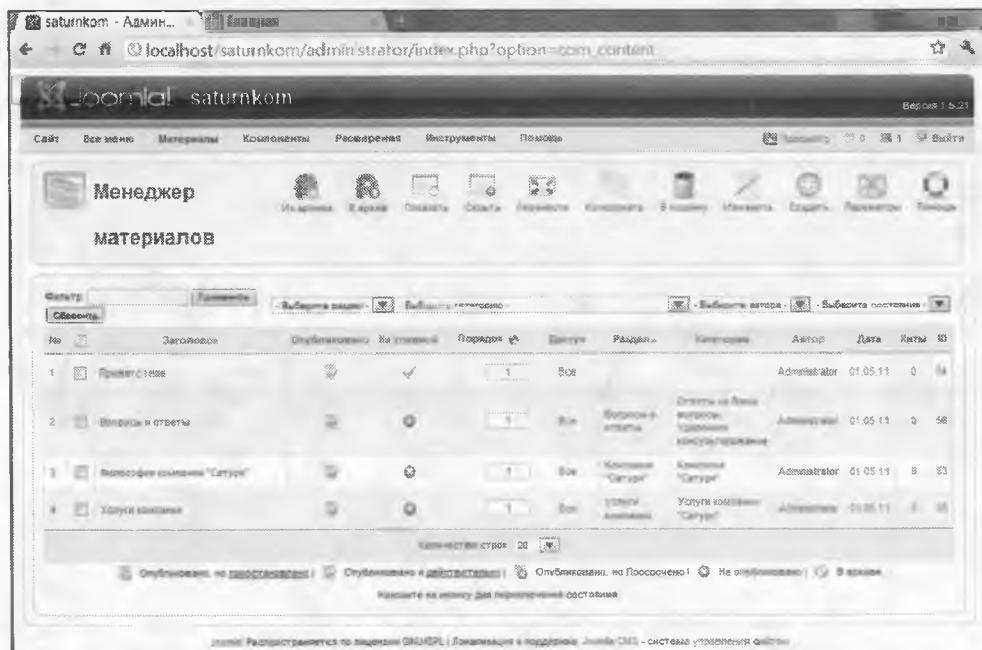


Рис. 9.37. Окно менеджера материалов с четырьмя созданными и готовыми к публикации материалами

Поскольку статьи уже готовы, пора перейти к следующему шагу построения сайта — организации меню.

Организация меню и определение соответствия статей пунктам меню в CMS Joomla!

Всего в CMS Joomla! по умолчанию существует четыре основных вида меню:

- верхнее, или горизонтальное, меню topmenu;
- главное, или вертикальное, меню mainmenu;
- прочее меню othermenu;
- пользовательское меню usermenu.

Очень редко шаблон CMS Joomla!, используемый для создания сайта, требует одновременного открытия нескольких меню. Обычно хватает одного горизонтального либо, что чаще, вертикального меню сайта.

Соответственно названию верхнее меню располагается горизонтально, в верхней части веб-страницы, как это было при создании сайта в главе 2. Главное — вертикальное — меню в обычных шаблонах располагается с левого, как сейчас, или с правого края веб-страницы. Прочее меню чаще всего находится в нижней части страницы и, как правило, содержит ссылки на дружественные сайты. Пользовательское меню будет отображаться только для зарегистрированного пользователя после его входа на сайт под личным логином, оно, как правило, содержит ссылки редактирования профиля и материала, ссылку для выхода с сайта и другие пункты, определяемые системой.

Для управления меню, разделами, категориями и статьями также служит **Менеджер меню**, активизировать который можно через пункт меню **Все меню** → **Менеджер меню** (рис. 9.38).

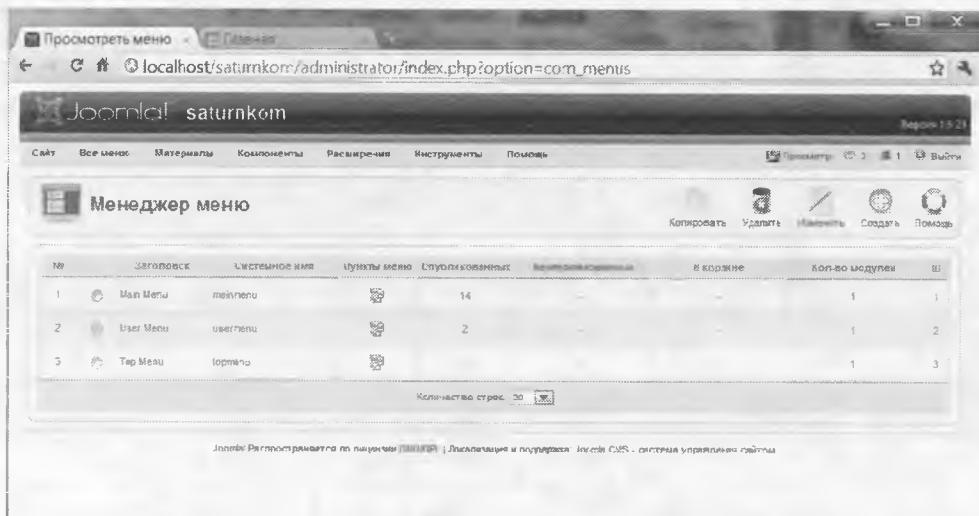


Рис. 9.38. Окно менеджера меню

Как видно на рис. 9.38, в наследство от автоматической установки демо-данных осталось три меню, и в главном опубликовано 14 пунктов. Для нас пока этого много, тем более что названия пунктов меню не соответствуют нашему сайту, следовательно, нужно отредактировать главное меню нашего сайта. Для этого установите переключатель в положение **Main Menu** и щелкните кнопкой мыши на значке **Изменить**, который находится в правом верхнем углу. В открывшемся окне **Детали меню** (рис. 9.39) изменим заголовок и описание меню, записав их на русском языке, чтобы в дальнейшем было удобнее работать. После этого щелкнем на значке **Сохранить**.

После внесения изменений окно менеджера меню будет выглядеть, как на рис. 9.40. Теперь пора приступить к редактированию пунктов меню, для этого следует выбрать значок **Пункты меню** для изменяемого Главного меню (см. рис. 9.40).

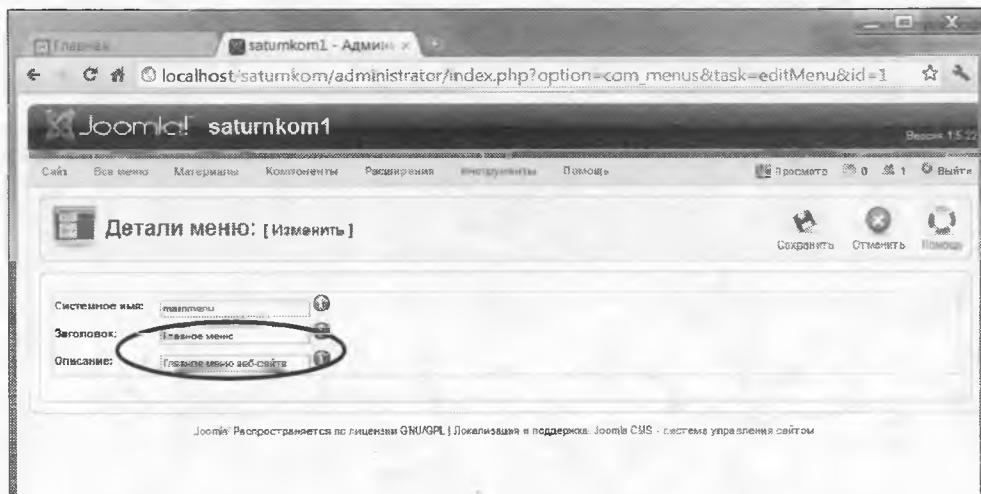


Рис. 9.39. Окно Детали меню

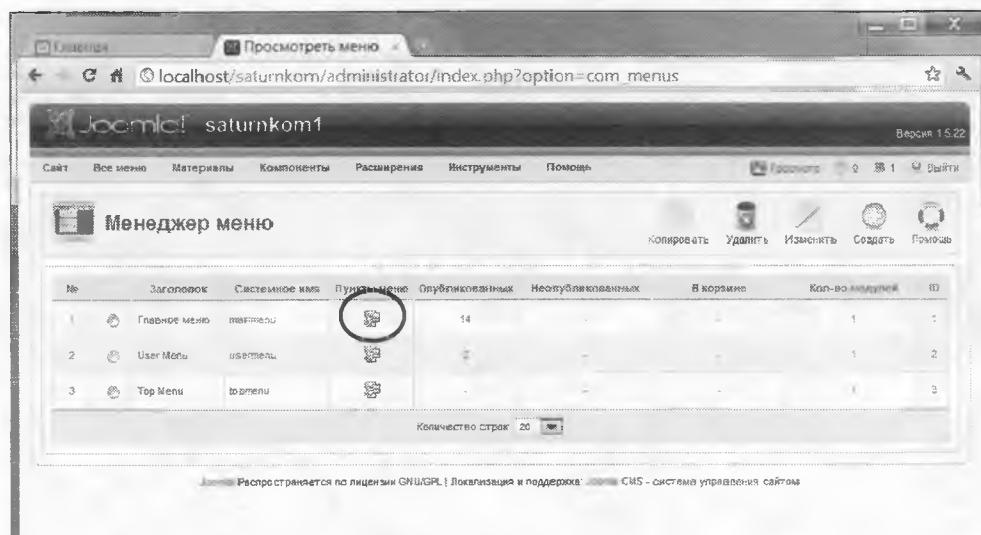


Рис. 9.40. Окно менеджера меню

На рис. 9.41 указаны все созданные при установке CMS Joomla! пункты меню, которые ведут к статьям, удаленным ранее. Имеет смысл либо скрыть, либо удалить пункты меню, которые не будут использоваться. Поскольку создание меню — очень ответственная задача, следует предварительно четко распланировать пункты, чтобы в дальнейшем не возникало трудностей при навигации по веб-страницам.

Почистив пункты меню, получим следующий результат (рис. 9.42).

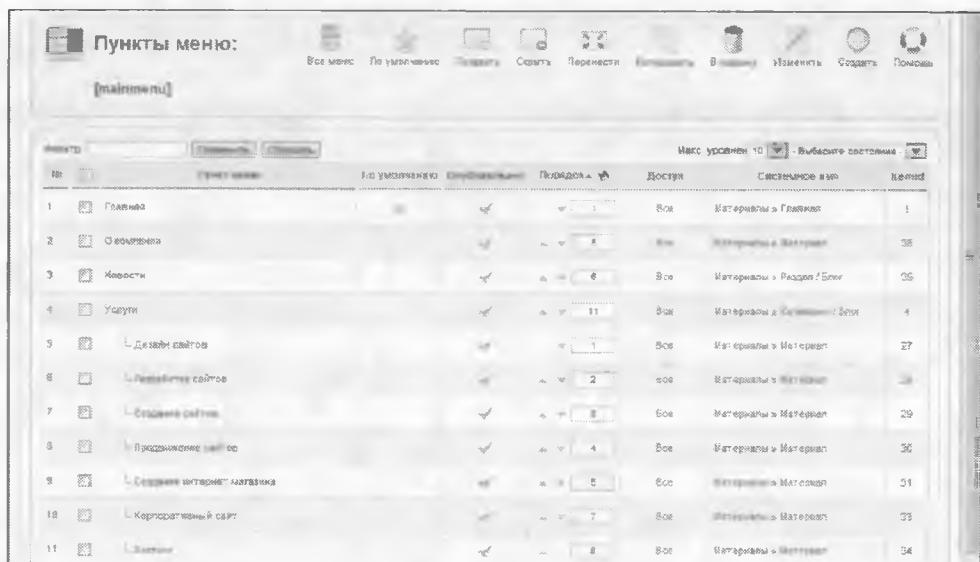


Рис. 9.41. Окно Пункты меню

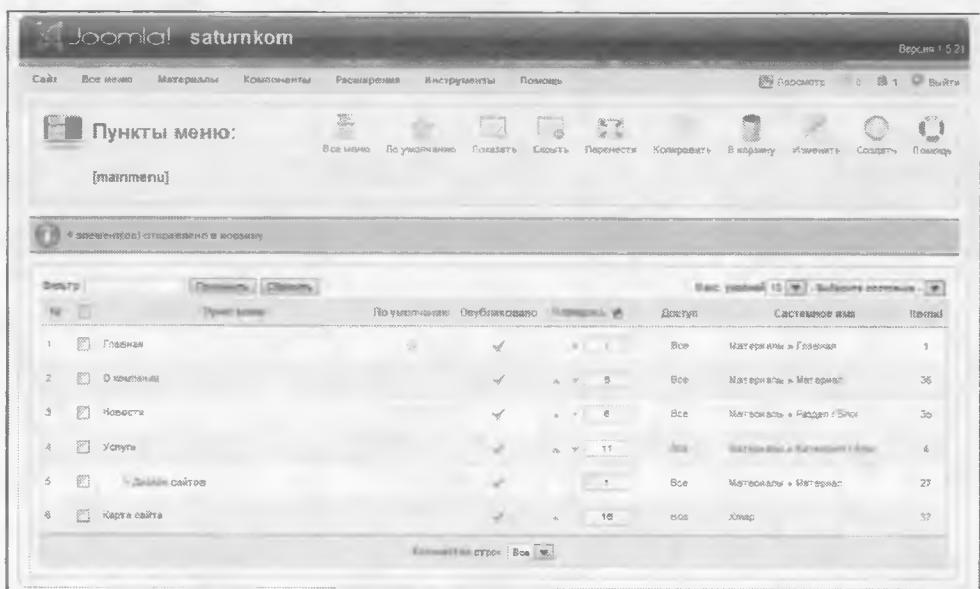


Рис. 9.42. Измененные пункты меню

Оставшиеся пункты меню необходимо переименовать в соответствии с нашими требованиями и закрепить за ними соответствующие статьи.

Для изменения пункта меню нужно либо щелкнуть на его названии кнопкой мыши, либо установить флажок напротив названия и щелкнуть

на значке Изменить, находящемся в правом верхнем углу окна. В результате будет открыто новое окно Пункт меню: Изменить (рис. 9.43).

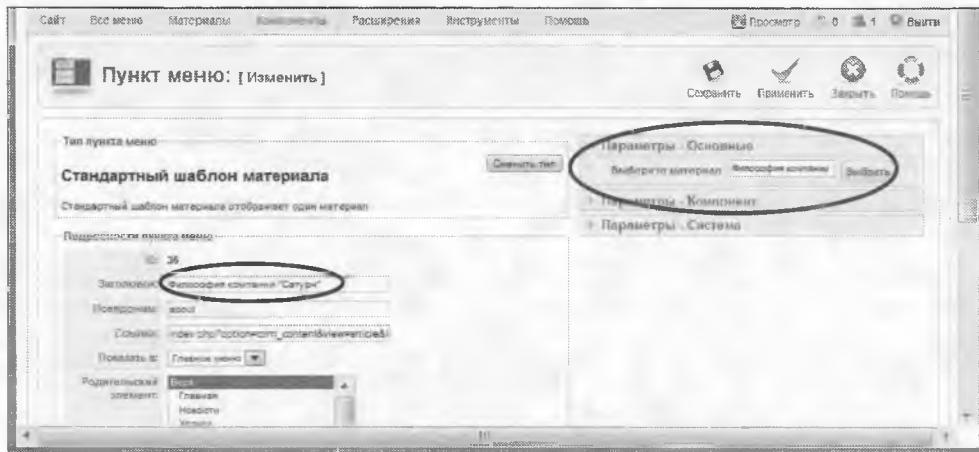


Рис. 9.43. Окно Пункт меню: Изменить

В области заголовка изменим надпись О компании на надпись Философия компании "Сатурн". Для заполнения окна Выберите материал нужно нажать кнопку Выбрать и в открывшемся окне (рис. 9.44) указать ту статью, которая и будет связана с этим пунктом меню. В нашем случае это статья Философия компании "Сатурн".

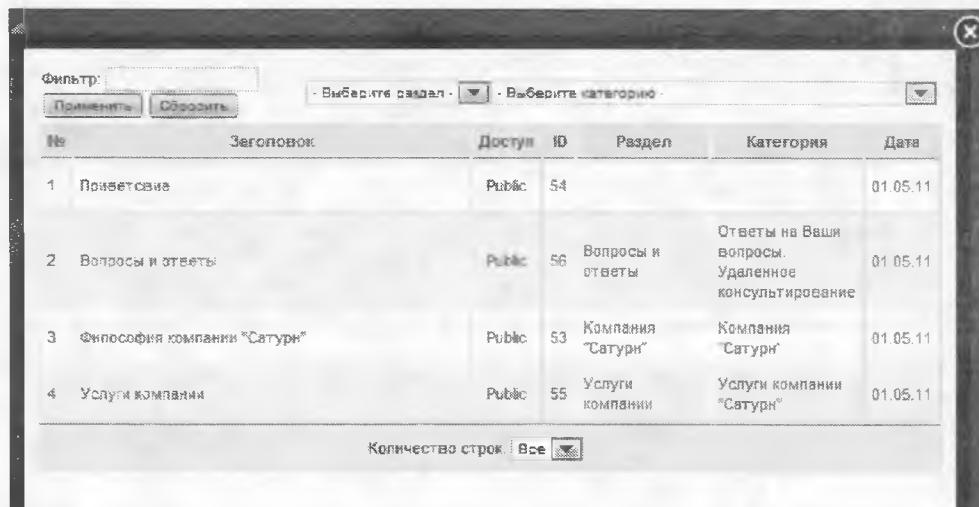


Рис. 9.44. Окно выбора статьи для связи с пунктом меню

После щелчка на значке Сохранить окно пунктов меню примет вид, как на рис. 9.45.

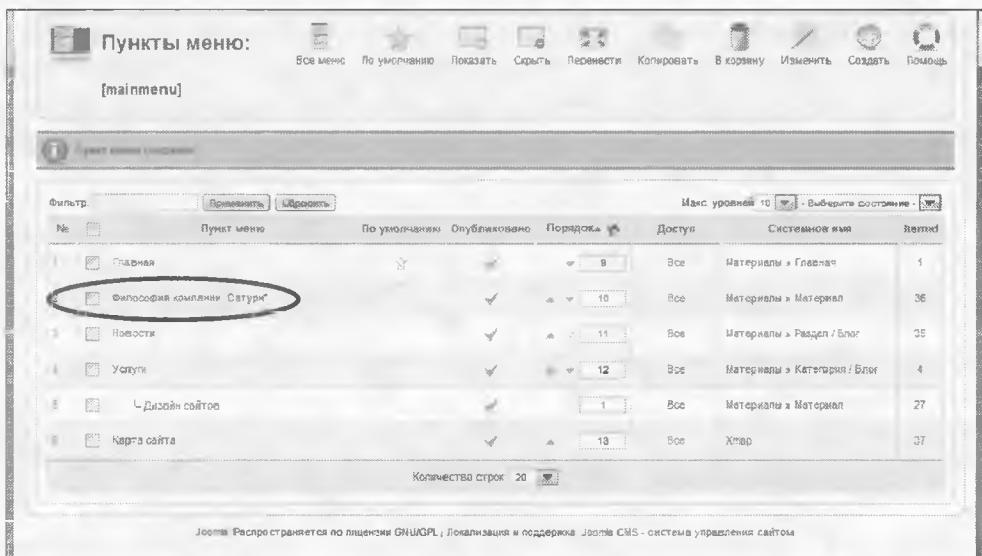


Рис. 9.45. Окно Пункты меню после внесения изменений

С изменением пункта меню О компании все прошло достаточно просто, поскольку он имеет тип стандартного шаблона материала. При изменении пункта меню Новости на пункт меню Услуги компании "Сатурн" все пройдет не так гладко, как в первый раз, так как он имеет тип Шаблон блога раздела (рис. 9.46).

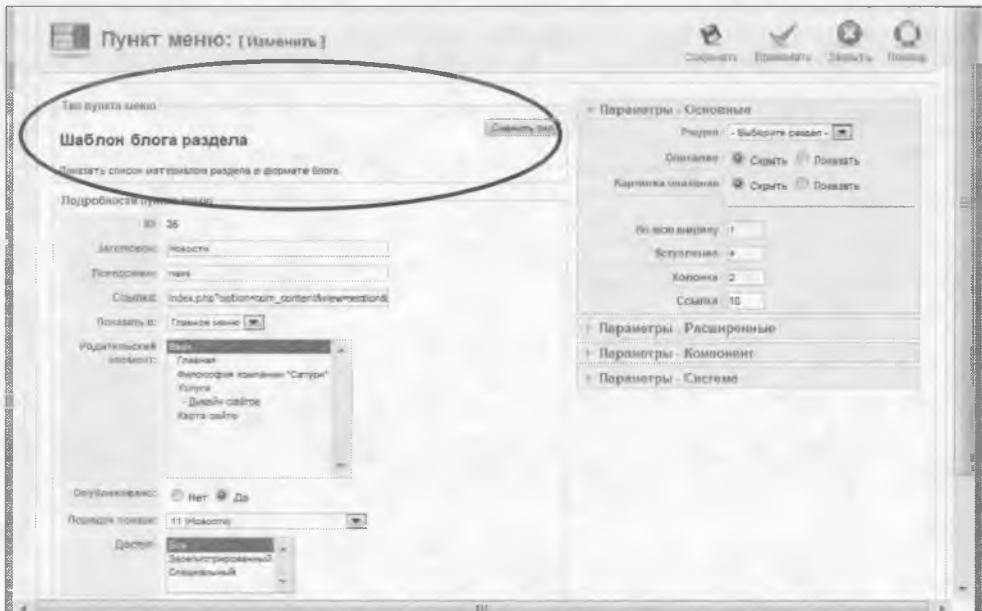


Рис. 9.46. Пункт меню Новости

При нажатии кнопки Сменить тип в открывшемся окне следует выбрать тип Стандартный шаблон материала (рис. 9.47).

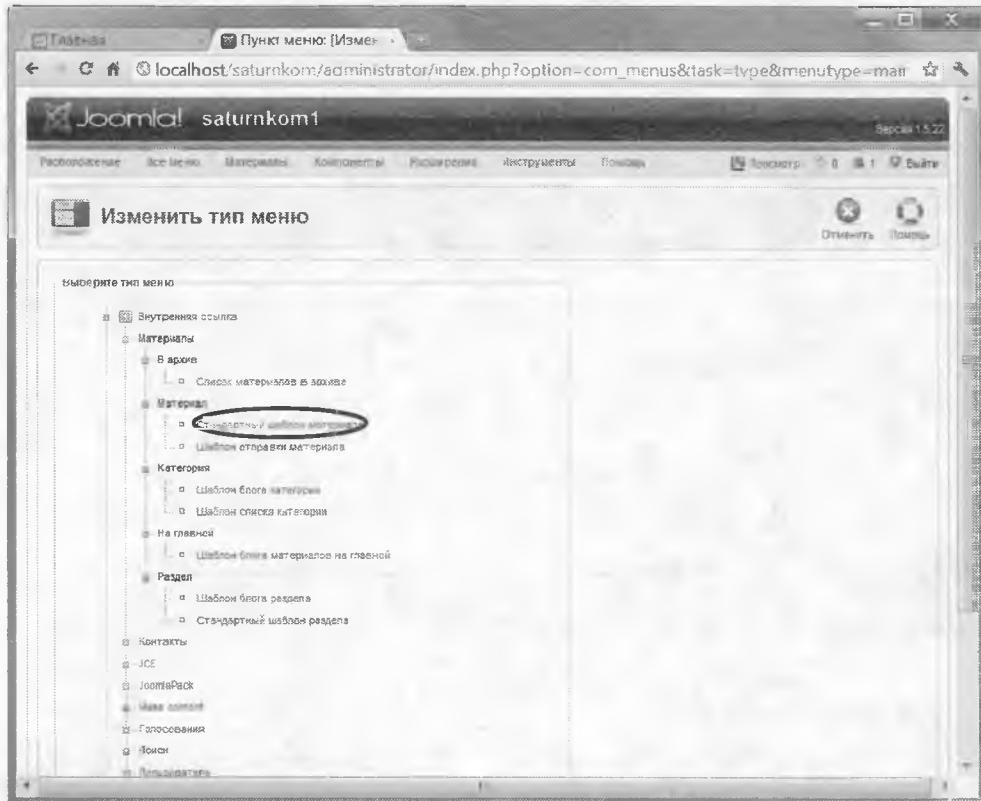


Рис. 9.47. Окно для задания типа пункта меню

Далее, переименовав пункт меню и определив его связь со статьей, сохраним его, обновим страницу сайта и посмотрим, как он работает (рис. 9.48).

Поскольку с прямым обращением пункта меню к статье более или менее все понятно, разберем случай, когда в одной категории у нас несколько статей.

Пока неизмененным у нас остался пункт меню Услуги, имеющий тип Материалы Категория/Блог, поэтому его и будем изменять. Переидем в окно изменения пункта меню (рис. 9.49).

Прежде всего изменим тип с Шаблона блога категорий на Шаблон списка категорий. Затем изменим заголовок на Вопросы и ответы. Далее определим, к какой категории будет относиться данный пункт меню. Щелкаем кнопкой мыши на значке Сохранить и в открывшемся окне наблюдаем следующую картину (рис. 9.50).

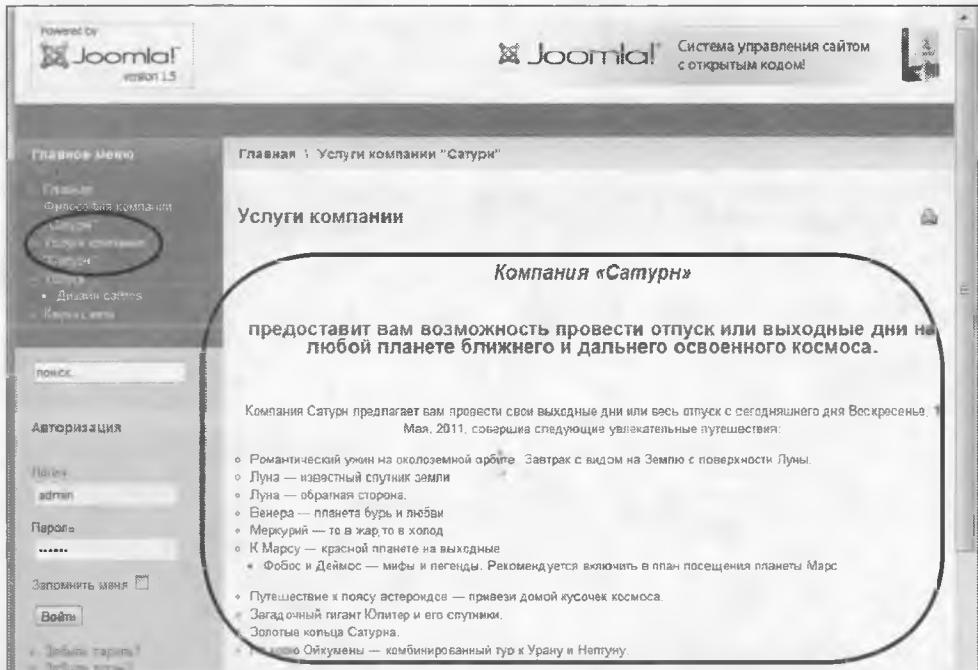


Рис. 9.48. Проверка работоспособности пункта меню Услуги компании "Сатурн"

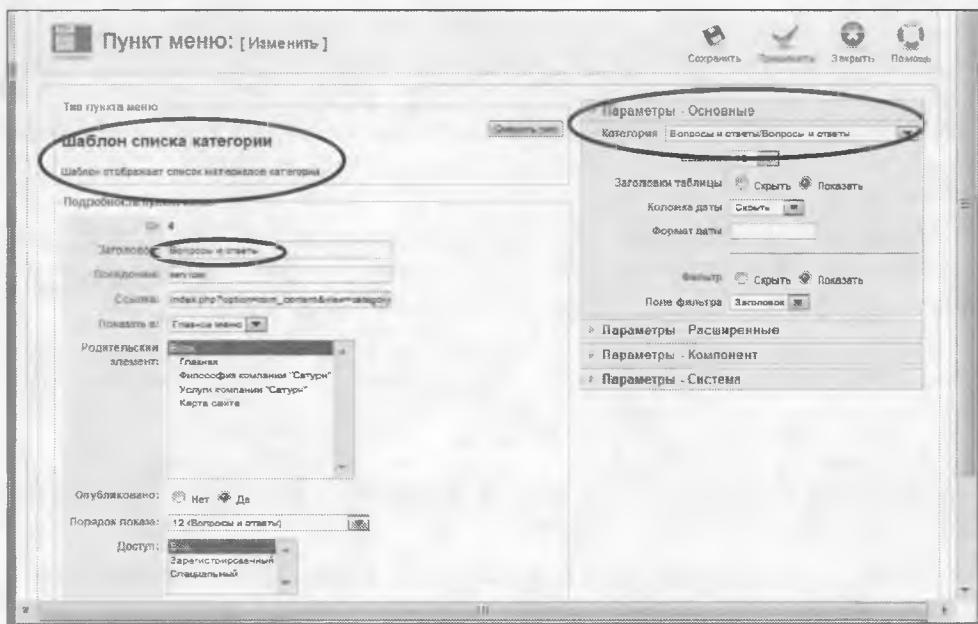


Рис. 9.49. Области, требующие изменений при редактировании пункта меню Вопросы и ответы

Рис. 9.50. Окно пунктов меню. В помеченной области последний неизмененный подпункт исходного меню

Для изменения подпункта меню **Дизайн сайтов** нужно выполнить стандартную процедуру редактирования пункта меню (рис. 9.51).

Рис. 9.51. Окно редактирования подпункта меню Вопросы и ответы

В открывшемся окне следует указать соответствующий заголовок и выбрать связанную с пунктом меню статью. Далее обязательно сохраняем внесенные изменения и обновляем страницу сайта (рис. 9.52).

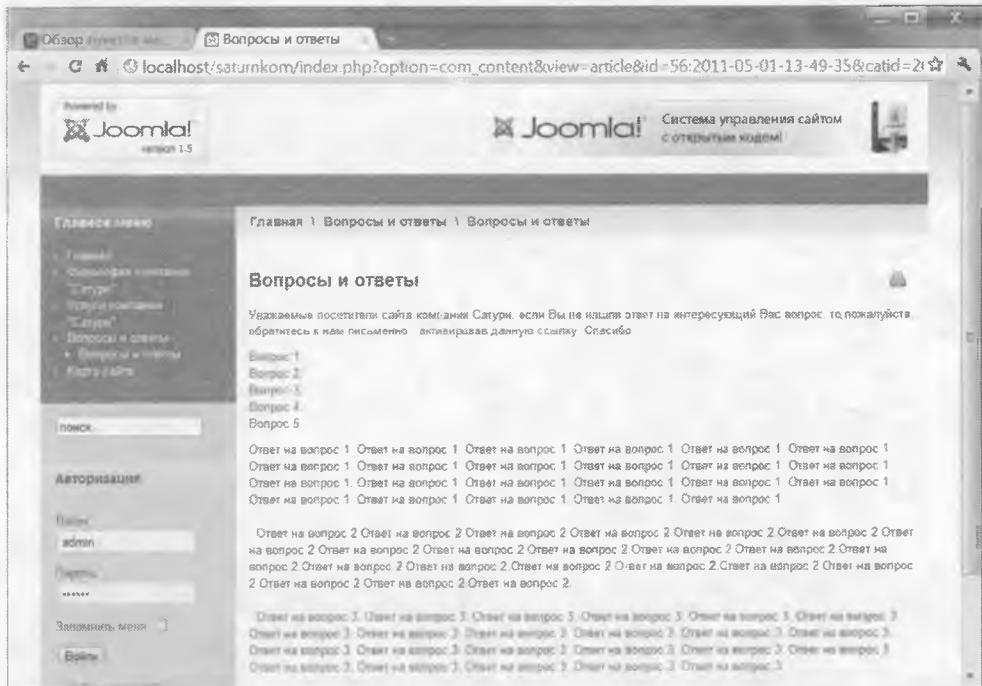


Рис. 9.52. Полностью измененное меню сайта с включенным меню категорий Вопросы и ответы

Как видно, создание пунктов меню хоть и хлопотное занятие, особенно при первоначальном планировании сайта, но совершенно необходимое. Доступ к любому материалу любой статьи возможен только через соответствующий пункт меню, и если нет указывающего на статью пункта, то ее невозможно открыть.

Создание новых пунктов меню ничем не отличается от редактирования уже имеющихся, только вместо значка Изменить необходимо выбирать значок Создать. В открывшемся окне необходимо указать тип пункта меню (рис. 9.53). Поскольку у нас неописанной осталась только одна некатегоризированная статья Приветствие, то создадим пункт меню для нее и сразу укажем тип Стандартный шаблон материала.

Далее продолжим заполнять окно Пункт меню: Создать, где и определим заголовок и ссылку на статью как Приветствие (рис. 9.54).

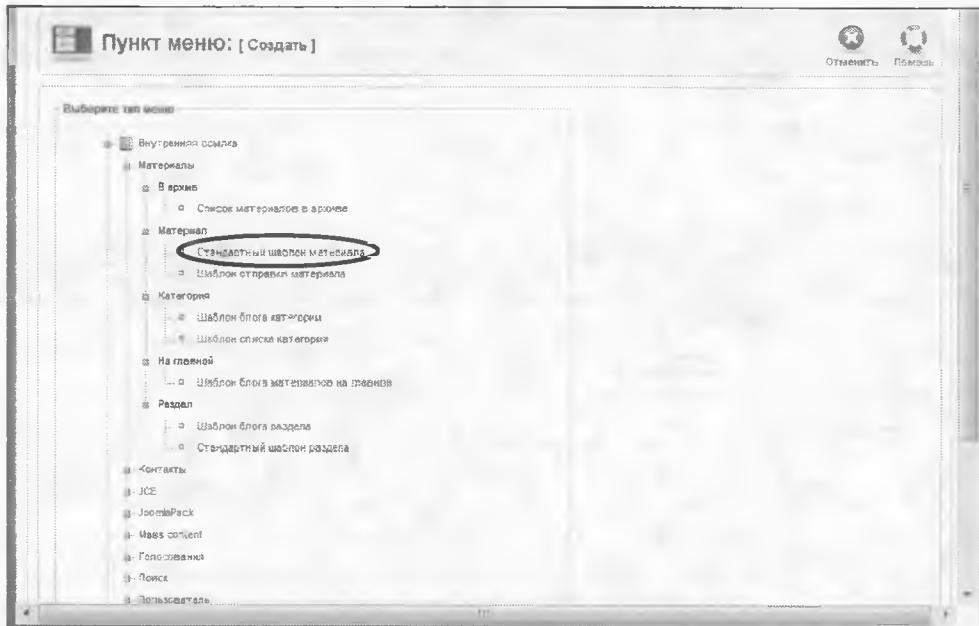


Рис. 9.53. Определение типа пункта меню
для некатегоризированной статьи

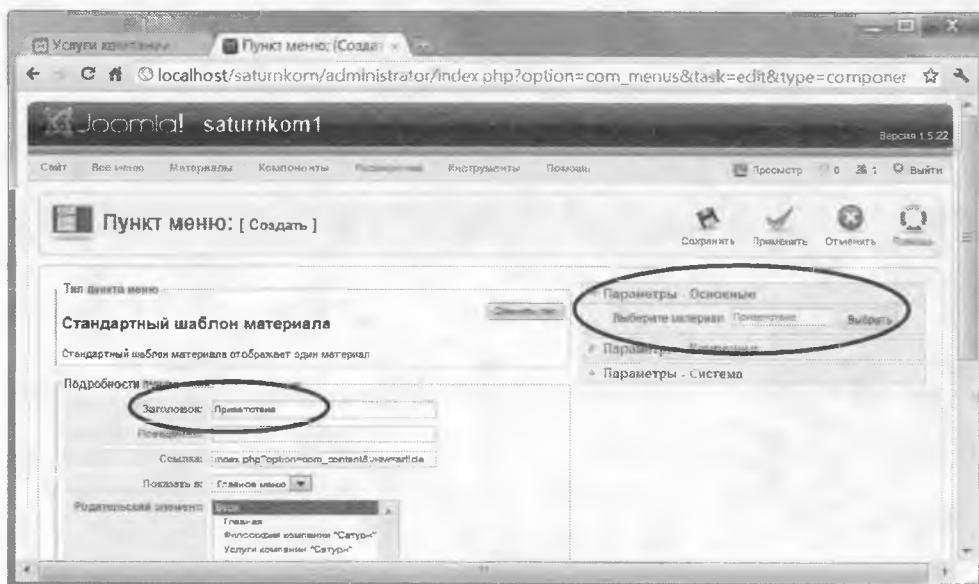


Рис. 9.54. Окно создания пункта меню
для некатегоризированной статьи Приветствие

Сохраним страницу и в открывшемся окне поднимем пункт Приветствие вверх на первую позицию, сразу под пункт меню Главная. Для этого необходимо щелкнуть на треугольных указателях в столбце Порядок (рис. 9.55).

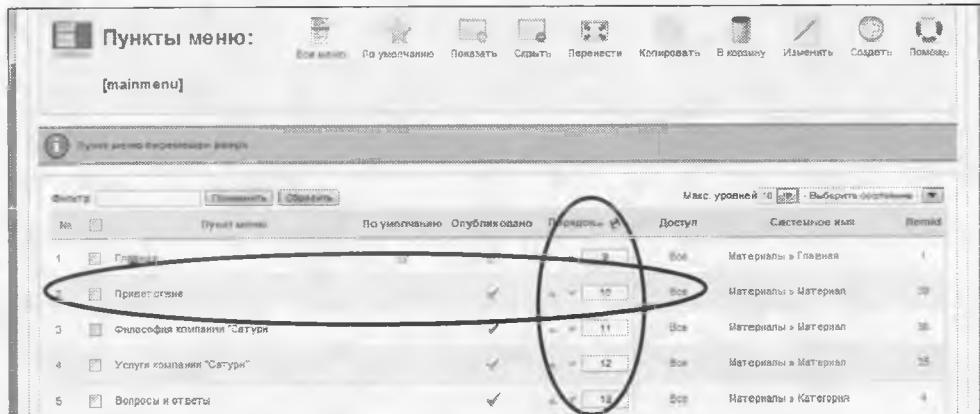


Рис. 9.55. Перемещение пунктов меню относительно друг друга для изображения на веб-страницах

После перемещения пункта меню Приветствие в обязательном порядке сохраним внесенные изменения, щелкнув на значке Сохранить в виде дискеты, расположенной в заголовке столбца Порядок.

Затем обновим веб-страницу на соседней вкладке и посмотрим на созданное меню сайта, связывающее нас с его статьями (рис. 9.56).

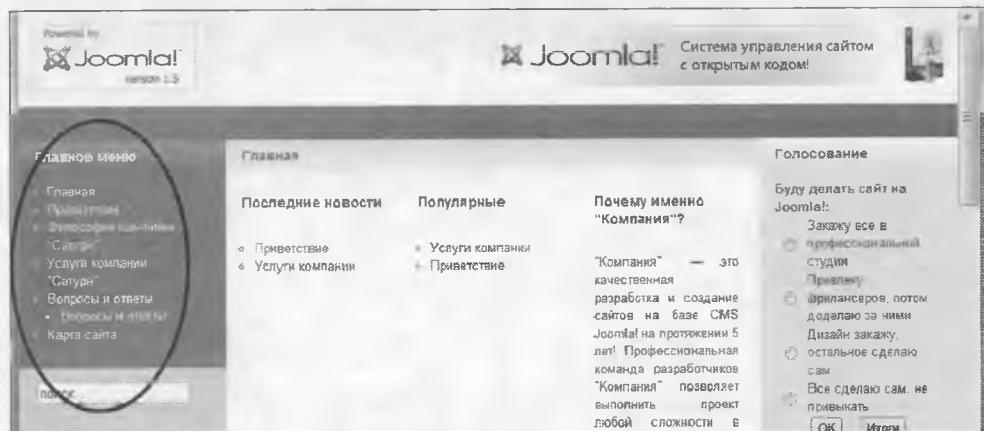


Рис. 9.56. Созданное меню сайта

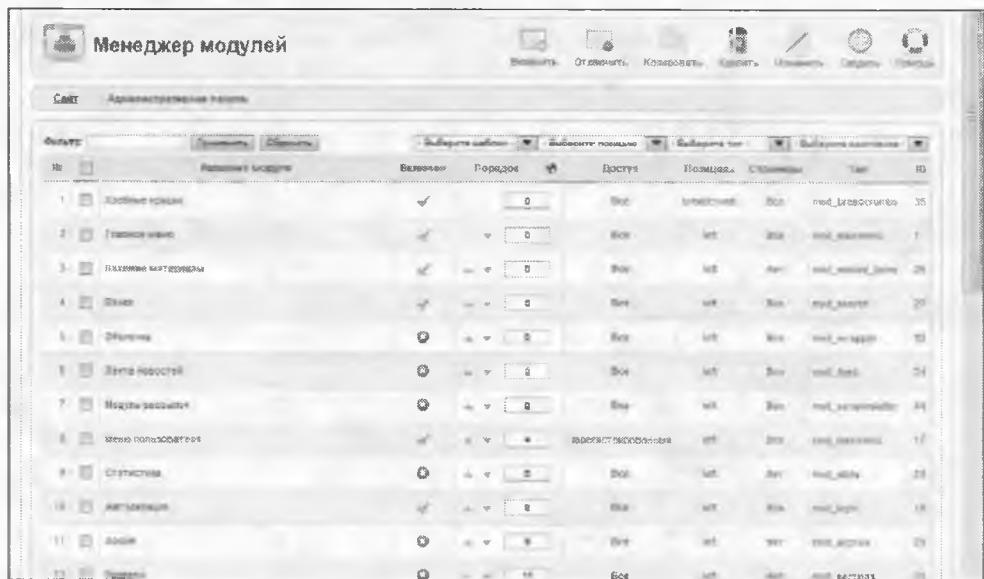
Таким образом, можно создавать или редактировать пункты меню и привязывать их к различным статьям, как категоризированным — относящимся к определенным разделам и категориям, так и к статическим.

9.5. Расширения

Управление модулями

Вот и пришла пора обратиться к той информации, что захламляет наши веб-страницы. Это не статьи, поскольку их мы удалили, а модули. **Модули** — это независимые области веб-страницы, в которых отображается контент, напрямую не относящийся к материалам сайта, например модуль голосования либо модуль меню. Обычно модули отражают желание пользователя получить все сразу. Это также может быть потребность не пользователя, а веб-мастера. Например, ему нечем заполнить веб-страницу, поэтому он и размещает здесь различные новостные ленты или продает рекламное место на веб-страницах. В любом случае подключение какого-либо модуля на небольшом сайте компании должно быть строго мотивировано. Согласитесь, вряд ли мне как пользователю нужна новостная лента от Mail.ru на сайте компании, производящей рыболовные сети. Кроме того, на сайте фирмы, монопольно выпускающей один вид товара, вызовет недоумение форма голосования с вопросом: «Нравится ли вам наша продукция?» Но так или иначе возможность использовать или не использовать дополнительные модули в CMS Joomla! есть, и это ваш выбор, а наша цель — узнать, как это сделать.

Управление модулями, как и всем остальным, что мы рассматривали выше, осуществляется с помощью менеджера модулей, вызвать который можно через меню **Расширения** ▶ **Менеджер модулей** (рис. 9.57).



The screenshot shows the Joomla Module Manager window. The title bar says 'Менеджер модулей'. The main area is a table with the following data:

№	Название	Видимость	Порядок	Быстрые	Позиция	Страницы	Тип	ID
1	Хлебные крошки	Видим	0	Быс.	Быс.	Все	mod_breadcrumbs	35
2	Главное меню	Видим	0	Быс.	Быс.	Все	mod_menu	1
3	Показать материалы	Видим	0	Быс.	Быс.	Все	mod_mainmenu	28
4	Лента	Видим	0	Быс.	Быс.	Все	mod_mainmenu	29
5	Оффсеты	Видим	0	Быс.	Быс.	Все	mod_mainmenu	31
6	Лента новостей	Видим	0	Быс.	Быс.	Все	mod_mainmenu	34
7	Модуль-реквизит	Видим	0	Быс.	Быс.	Все	mod_mainmenu	34
8	Меню пользователя	Видим	0	Быс.	Быс.	Все	mod_mainmenu	37
9	Статистика	Видим	0	Быс.	Быс.	Все	mod_mainmenu	38
10	Автоматика	Видим	0	Быс.	Быс.	Все	mod_mainmenu	39
11	Логин	Видим	0	Быс.	Быс.	Все	mod_mainmenu	40
12	Онлайн	Видим	0	Быс.	Быс.	Все	mod_mainmenu	41

Рис. 9.57. Окно менеджера модулей

Для отключения или включения определенного модуля необходимо установить или снять флажок в столбце **Включен** напротив имени модуля. Отключайте ненужные модули с осторожностью, каждый раз просматривая веб-страницу на предмет изменений. Дело в том, что локализация в CMS Joomla! не совсем совершенна, и что переводчик имел в виду под названием модуля **Хлебные крошки**, не совсем понятно. В результате отключения ненужных модулей главная страница нашего сайта выглядит, как на рис. 9.58.

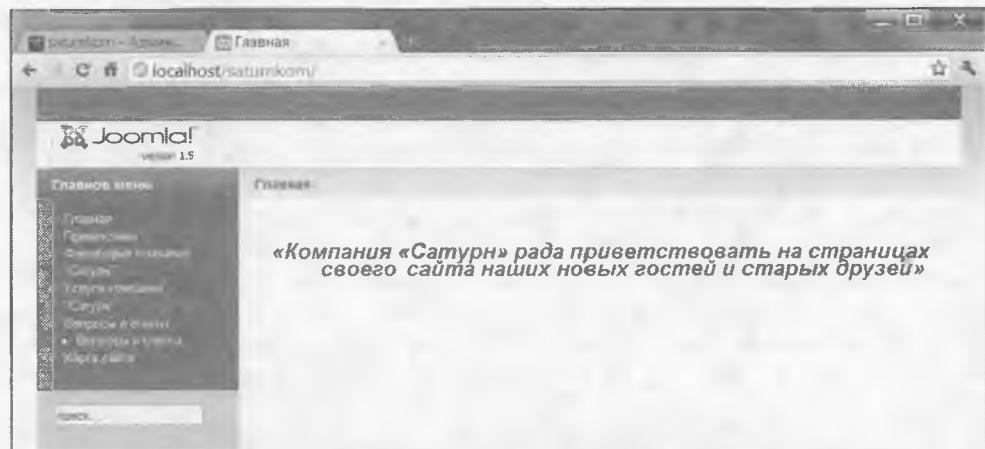


Рис. 9.58. Вид главной страницы сайта после отключения ненужных модулей

Разберемся со способами редактирования модулей, для чего щелкнем кнопкой мыши на имени модуля, который собираемся изменить. В нашем случае это модуль **Хлебные крошки**. В появившемся окне (рис. 9.59) найдем описание модуля и в соответствии с ним исправим содержимое поля заголовка. Кроме того, следует обратить внимание на текст главной страницы, ведь при необходимости его можно изменить согласно своим требованиям.

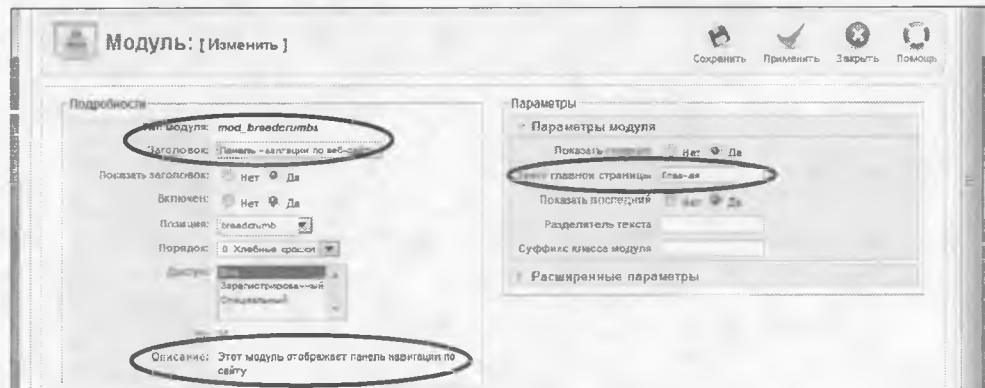


Рис. 9.59. Окно редактирования модуля

После редактирования модуля сохраняем внесенные изменения, щелкнув на соответствующем значке.

Поскольку модули — это просто часть веб-страницы, на которой отображается соответствующая информация, то первоначально эта информация должна быть подготовлена к размещению в области модуля. Подготовкой занимаются другие части CMS Joomla!.

Создание опроса-голосования на базе CMS Joomla!

Голосование — это довольно мощный способ узнать мнение посетителей сайта по различным вопросам при условии, что оно не является установленным навсегда, а регулярно изменяется. Поэтому система Joomla! имеет универсальный инструмент по созданию опросов.

Как отмечалось выше, в системе есть модуль, который как раз и отвечает за голосование посетителей. Маленькая проблема заключается только в том, что данный модуль сегодня имеет в ассортименте только одно созданное по умолчанию голосование *Буду делать сайт на Joomla!*, поэтому необходимо перед подключением модуля создать новое голосование и определить его модулю.

Для создания нового голосования придется обратиться к меню **Компоненты** и выбрать пункт **Голосование**, в результате откроется стандартное для системы окно менеджера голосований (рис. 9.60), где и будет показано единственное существующее голосование *Буду делать сайт на Joomla!*.

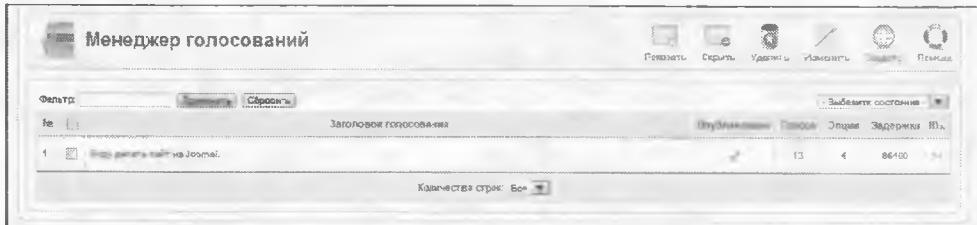


Рис. 9.60. Окно менеджера голосований

Необходимо щелкнуть кнопкой мыши на значке **Создать** и в открывшемся окне создания голосований заполнить необходимые поля (рис. 9.61).

В нашем случае заголовком голосования будет вопрос: *Путешествовали ли Вы по просторам космоса?* Ответы будут следующие.

- Нет.
- Да, но только в пределах лунной орбиты.
- Да, на внутренних планетах солнечной системы.
- Да, на внешних планетах солнечной системы.
- Да, за пределами солнечной системы.

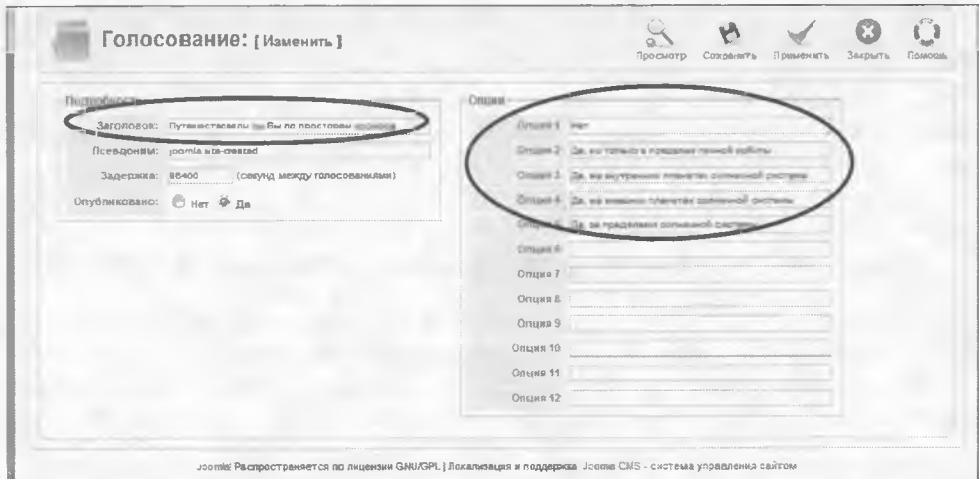


Рис. 9.61. Окно создания голосования средствами CMS Joomla!

Затем сохраняем, щелкнув на соответствующем значке, и в менеджере голосования видим новое голосование Путешествовали ли Вы по просторам космоса. Не забываем опубликовать данное голосование, установив соответствующий флажок в столбце Опубликовано и сняв его для ранее созданного голосования (рис. 9.62).

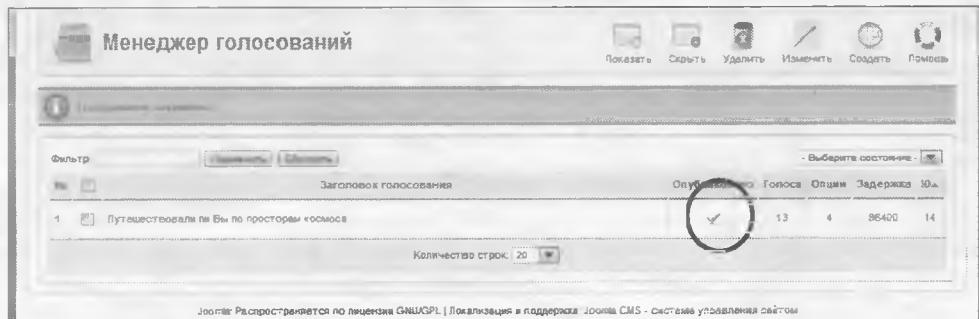


Рис. 9.62. Установка параметров публикации для вновь созданного голосования

Теперь пришло время вернуться к менеджеру модулей, для чего выполним команду Расширения ▶ Менеджер модулей. Здесь выберем окно редактирования модуля Голосование. В открывшемся окне в параметрах модуля из раскрывающегося списка укажем наше голосование Путешествовали ли Вы по просторам космоса (рис. 9.63).

Сохраним изменения, щелкнув кнопкой мыши на значке Сохранить. После этого нужно обязательно опубликовать голосование в окне менеджера модулей, а затем обновить веб-страницу на соседней вкладке (рис. 9.64).

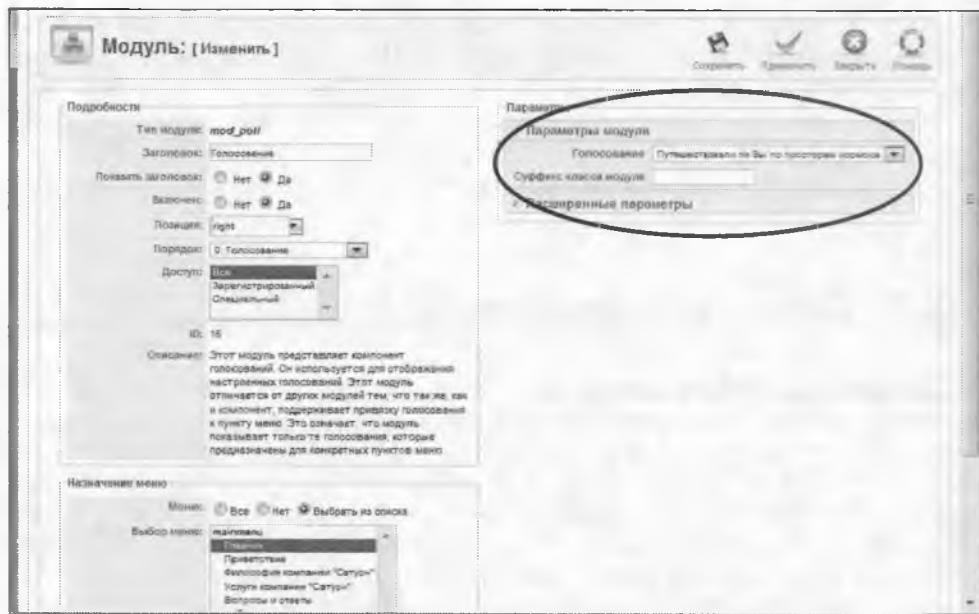


Рис. 9.63. Окно редактирования модуля Голосование

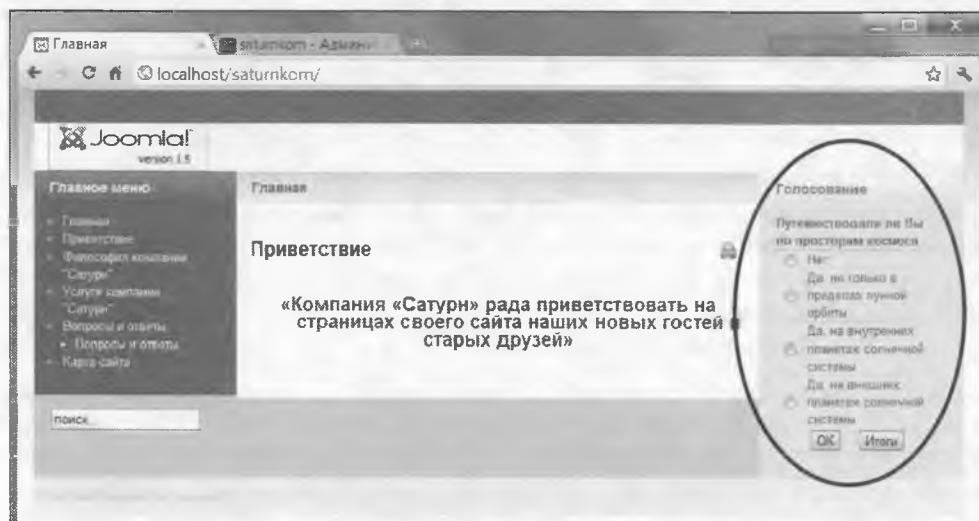


Рис. 9.64. Вновь созданный модуль голосования на главной странице сайта

При формировании и редактировании компонента и модуля Голосование обращайте внимание на расширенные параметры как компонента, так и модуля. В их характеристиках указывается временной интервал, допустимый между голосованиями, и срок кэширования результатов голосований.

Мы рассмотрели тот минимум сведений, которые нужны для работы с установленными по умолчанию модулями. При необходимости дополнительные модули и расширения можно скачать из Сети как в бесплатном, так и в коммерческом варианте. Если же вы всерьез увлечетесь созданием сайтов на базе CMS Joomla!, то изучите дополнительную документацию. Тогда вы сумеете самостоятельно писать как модули, так и шаблоны, о которых мы сейчас и поговорим.

9.6. Смена внешнего вида сайта

Шаблоны в CMS Joomla!

Удобство практически каждой системы управления содержимым заключается в том, что вне зависимости от наполнения сайта его можно представлять в различном виде. Для этого следует только сменить шаблон, который и формирует из контента базы данных SQL-сервера готовые к просмотру в браузере страницы, написанные на HTML. CMS Joomla! в этом случае не исключение. Для нее на просторах Интернета существуют сотни, если уже не тысячи готовых шаблонов, предоставляемых на безвозмездной или коммерческой основе любому пользователю. Если вы решите вплотную заниматься созданием и сопровождением сайтов, а в последние годы этот бизнес серьезно идет в гору, то после глубокого освоения языка гипертекстовой разметки и каскадных таблиц стилей самостоятельное написание рабочего шаблона для CMS Joomla! не составит особого труда. Тем более что документации на эту тему в Сети достаточно, и она находится в открытом доступе.

Пока же разберемся с готовыми шаблонами, находящимися в системе по умолчанию или самостоятельно устанавливаемыми пользователями. Управлять сменой установленного шаблона в CMS Joomla! можно с помощью менеджера шаблонов. Для его вызова (рис. 9.65) следует выполнить команду **Расширения** ▶ **Менеджер шаблонов**.

Как видно из окна, представленного на рис. 9.65, по умолчанию в базовом дистрибутиве CMS Joomla! есть несколько шаблонов, определяющих внешний вид страниц сайта. Шаблон, напротив имени которого в столбце **По умолчанию** стоит звездочки, является активным и формирует внешний вид веб-страницы. Если подвести указатель мыши к имени шаблона, то всплывет окно подсказки. В нем будет продемонстрирован внешний вид веб-страниц, которые сгенерируются при установке выбранного шаблона в режиме по умолчанию. Для смены шаблона достаточно установить переключатель в соответствующее положение и щелкнуть кнопкой мыши на значке **По умолчанию** в правом верхнем углу окна. Посмотрим на наш сайт, пропущенный через разные шаблоны (рис. 9.66–9.68).

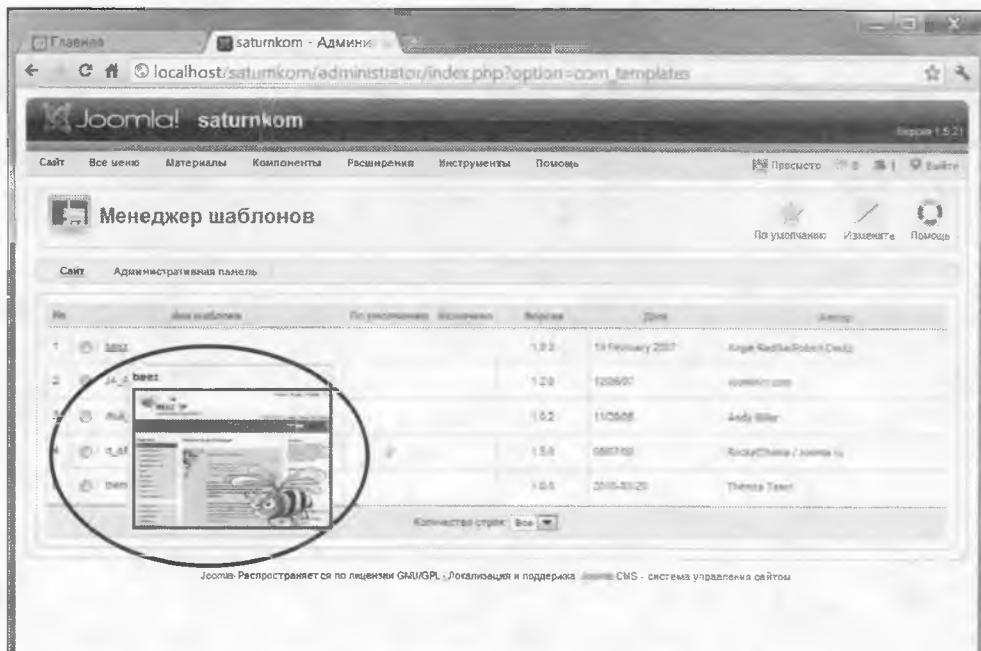


Рис. 9.65. Окно менеджера шаблонов



Рис. 9.66. Применение к оформлению веб-страниц шаблона beez

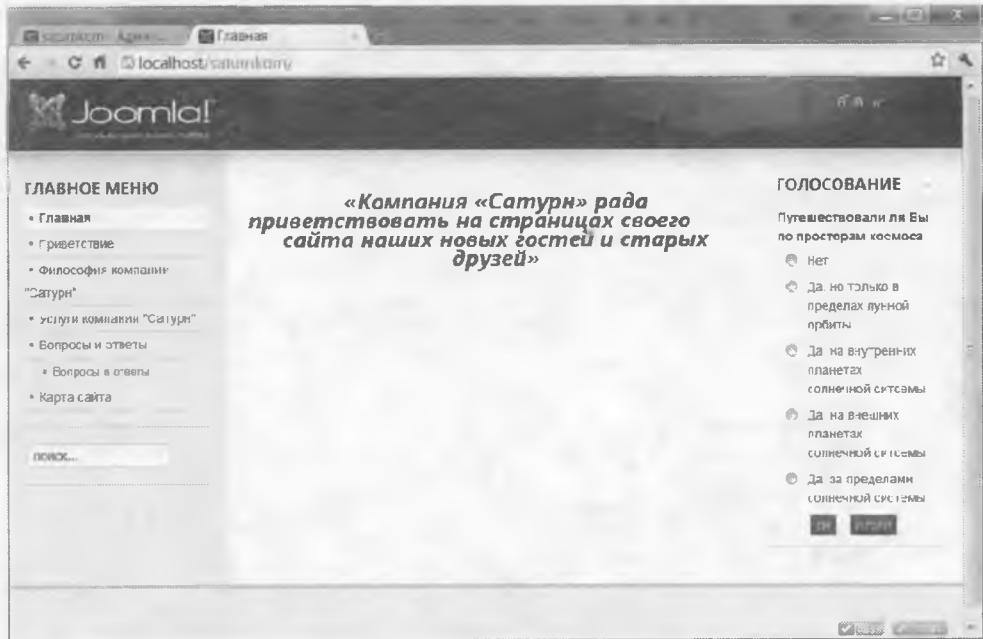


Рис. 9.67. Применение к оформлению веб-страниц шаблона JA_Purity



Рис. 9.68. Применение к оформлению веб-страниц шаблона rhuk_milkyway

Как видно из представленных выше рисунков, содержимое главной страницы и ее структура остаются неизменными и при задании другого шаблона изменяется только внешнее оформление веб-страницы. Таким образом, для изменения внешнего вида сайта достаточно скачать из Сети понравившийся шаблон и установить его для дальнейшего использования в системе управления содержимым. Как это сделать?

Шаблоны для CMS Joomla! обычно доступны для свободного или коммерческого скачивания в виде ZIP- или TAR-архивов. Первоначально вы должны получить архивный файл шаблона, и, поскольку CMS Joomla! умеет работать с архивными файлами, то его можно даже не распаковывать. Далее следует выполнить команду **Расширения** ▶ **Установить** или **Удалить**. В открывшемся окне менеджера расширений нужно указать имя и расположение загружаемого архивного файла, содержащего устанавливаемый шаблон, а затем нажать кнопку **Загрузить файл & Установить** (рис. 9.69).

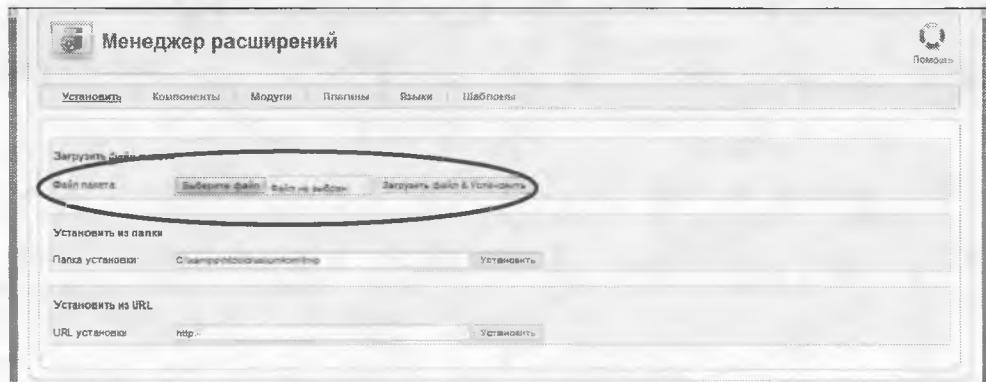


Рис. 9.69. Окно менеджера расширений

Установка шаблона обычно проходит без замечаний, и при завершении выводится сообщение об успешно выполненной операции. Теперь вернемся к окну менеджера шаблонов, где нужно определить новый шаблон установленным по умолчанию (рис. 9.70).

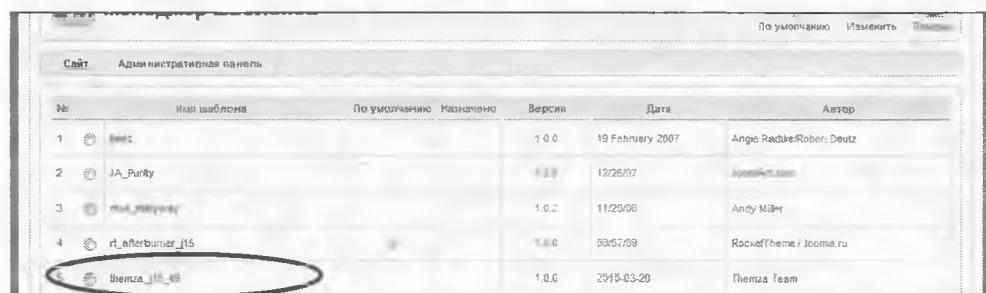


Рис. 9.70. Окно менеджера шаблонов с установленным по умолчанию новым шаблоном `themza_j15_49`

Далее перейдем на соседнюю вкладку с отображением страниц сайта и обновим ее содержимое. В результате наш сайт получит новое оформление (рис. 9.71).



Рис. 9.71. Новый шаблон сайта

При выборе и установке новых шаблонов следует иметь в виду, что не все шаблоны имеют в своем устройстве те модули, которые вы определили для своего сайта. Например, в установленном нами шаблоне отсутствует модуль Голосование, так как он не предусмотрен конструкцией шаблона (см. рис. 9.71). Как следствие, необходимо менять шаблон на другой либо повышать квалификацию в области веб-программирования и самостоятельно вносить изменения. Следует также учитывать, что навигация, предусмотренная в данном шаблоне, может отличаться от ваших дизайнерских задумок, поэтому внимательно выбирайте новый шаблон для своего сайта.

Теперь вернемся к ранее установленным классическим шаблонам CMS Joomla! И попробуем изменить логотип нашего сайта с установленного по умолчанию на заранее приготовленный в главе 5.

Изменение стандартного логотипа сайта в шаблоне CMS Joomla!

Логотип, обычно расположенный в графическом заголовке, — лицо сайта. Вне зависимости от применяемого шаблона нужно установить фирменный логотип компании. Особых сложностей это не представляет, поскольку теоретически всю процедуру можно описать следующим образом: графический файл с логотипом CMS Joomla! следует заменить графическим файлом с логотипом компании. Если оба файла имеют одинаковый тип и линейные размеры, то все сводится к переименованию файла с логотипом компании и его установке вместо одноименного файла с логотипом CMS Joomla!. Но описанная возможность смены логотипа — достаточно редкий

случай. Очень часто приходится вносить минимальные, но необходимые изменения в сам шаблон сайта, чем мы сейчас и займемся.

Итак, установим по умолчанию шаблон `rhuk_milkyway`, логотип на котором и будем изменять (рис. 9.72).

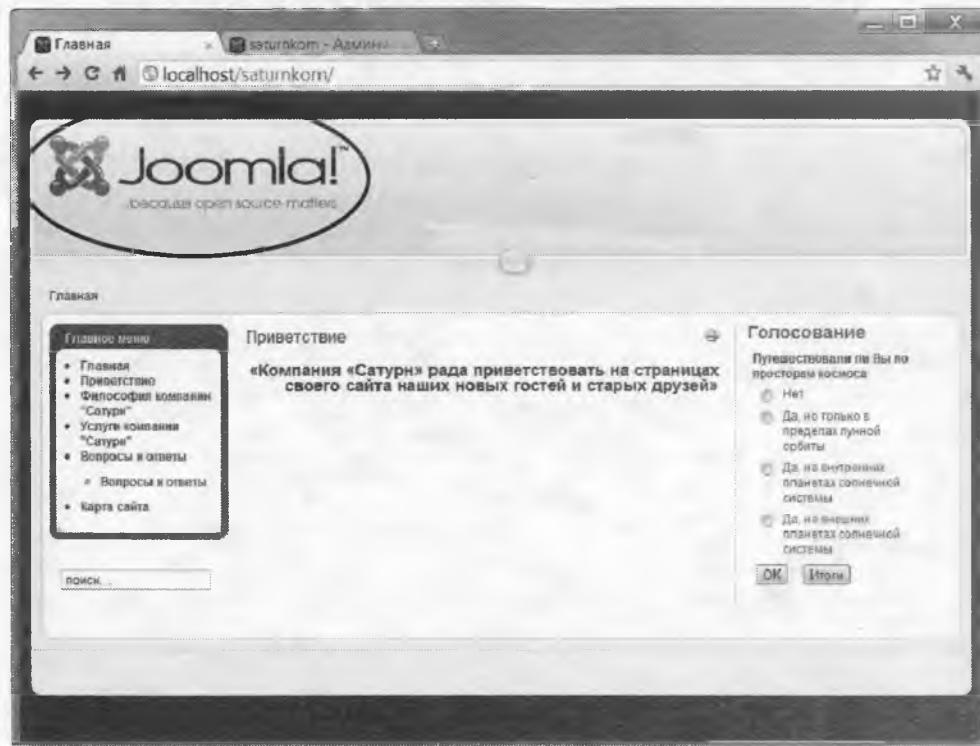


Рис. 9.72. Область расположения графического логотипа сайта, созданного с применением шаблона `rhuk_milkyway`

На следующем шаге расположим в каталоге текущего шаблона файл `nadpis.gif`, содержащий наш графический заголовок **Приветствие**, который мы создали в главе 5. Его надо разместить в подкаталоге `/images` папки `templates` для нашего шаблона `rhuk_milkyway`, для чего можно воспользоваться FTP-клиентом FileZilla, о котором уже упоминалось в главе 3. На этом самую легкую часть работы следует считать завершенной. Дело в том, что любой шаблон в CMS Joomla! представляет собой простой набор инструкций, в которых прописано, как и какой последовательности следует располагать те или иные элементы на веб-странице. Обычно эти инструкции помещаются в файлы каскадных таблиц стилей, поэтому необходимо знать язык гипертекстовой разметки и желательно разбираться в структуре каскадных таблиц стилей. К сожалению, и этого мало, так как в оформлении страниц сайта также участвуют графические элементы, которые, как вы помните, имеют строго заданные линейные размеры. Соответственно, при

добавлении какого-либо графического объекта в структуру шаблона CMS Joomla! придется в обязательном порядке изменять и графические элементы самого шаблона, связанные с добавляемым элементом. В самом изменении линейных размеров графических объектов нет абсолютно ничего сложного. Главное – понять логику создателя шаблона и найти те файлы, в которых будут содержаться необходимые для изменения графические элементы сайта. Рассмотрим это на примере.

Поскольку наш файл nadpis.gif уже находится в положенном ему каталоге, нужно указать шаблону, что он должен использовать в заголовке веб-страниц вместо стандартной заранее подготовленную надпись. Для этого перейдем в окно менеджера шаблонов, выберем редактируемый шаблон rhuk_milkyway для изменения и щелкнем кнопкой мыши на значке Изменить. В открывшемся окне Шаблон: Изменить щелкнем на значке Редактировать CSS (рис. 9.73).

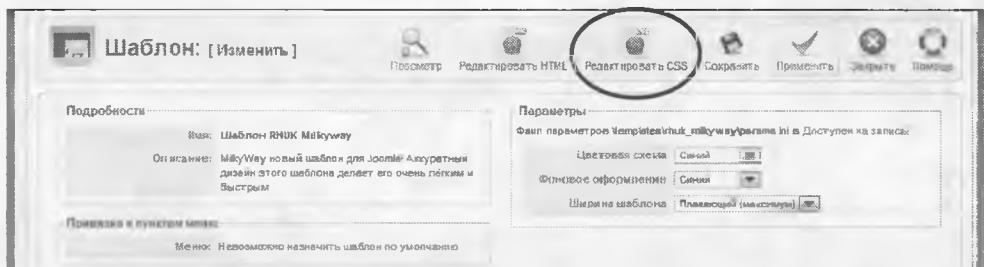


Рис. 9.73. Окно Шаблон: Изменить

После выполнения последовательности действий откроется окно Редактор CSS файла шаблона (рис. 9.74).



Рис. 9.74. Окно Редактор CSS файла шаблона

Из предложенного списка следует указать файл шаблона `template.css`, который мы и будем редактировать. Затем нужно выбрать данный файл для изменения и щелкнуть на значке **Изменить**. В появившемся окне редактора файлов необходимо найти следующий текст:

```
div#logo {  
    position: absolute;  
    left: 0;  
    top: 0;  
    float: left;  
    width: 298px;  
    height: 75px;  
    background: url(..../images/mw_joomla_logo.png) 0 0 no-repeat;  
    margin-left: 30px;  
    margin-top: 25px;  
}
```

Затем нужно заменить написанное в нем имя графического файла `mw_joomla_logo.png` для логотипа CMS Joomla!, установленного по умолчанию, на имя нашего файла `nadpis.gif`. Таким образом, измененный контейнер с надписью в теле файла примет вид:

```
div#logo {  
    position: absolute;  
    left: 0;  
    top: 0;  
    float: left;  
    width: 298px;  
    height: 75px;  
    background: url(..../images/ nadpis.gif) 0 0 no-repeat;  
    margin-left: 30px;  
    margin-top: 25px;  
}
```

Затем следует щелкнуть кнопкой мыши на значке **Применить**. Но не торопитесь щелкать на значке **Сохранить**, так как мы еще не выходим из окна редактирования. Перейдите на вкладку с веб-страницей и обновите ее (рис. 9.75).

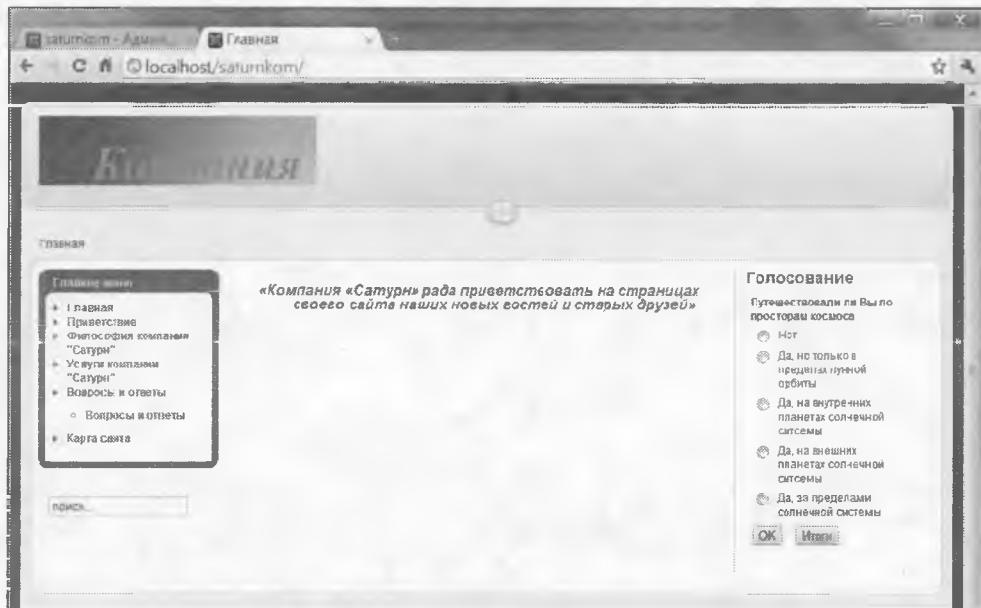


Рис. 9.75. В шаблон были внесены изменения

Как видно из рис. 9.75, наш графический файл успешно внедрен в шаблон, но почему-то показывается не полностью, а частично. Все дело в том, что установленный по умолчанию файл логотипа CMS Joomla! имеет меньшие размеры, соответственно и зарезервированное для него место в шаблоне также меньше. Для изменения предоставляемого графическому объекту места в шаблоне вернемся к надписи в теле файла `template.css` и зададим ширину графического объекта 600 пикселов, а высоту — 120 пикселов. Это соответствует фактическому размеру надписи. Таким образом, содержимое контейнера будет выглядеть так:

```
div#logo {
    position: absolute;
    left: 0;
    top: 0;
    float: left;
    width: 600px;
    height: 120px;
    background: url(..../images/nadpis.gif) 0 0 no-repeat;
    margin-left: 30px;
    margin-top: 25px;
}
```

Обновленная веб-страница примет вид, как на рис. 9.76.

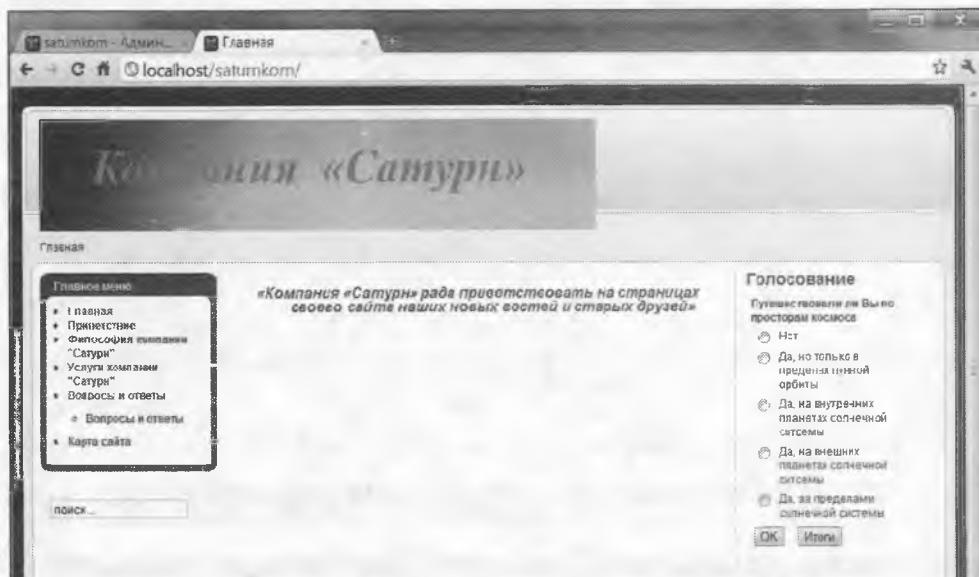


Рис. 9.76. Обновленная веб-страница с полностью видимым графическим объектом

Как видно из рис. 9.76, наш графический объект прекрасно помещается на веб-странице, и для невзыскательных пользователей можно все так и оставить. Но поскольку мы профессионалы, придется углубляться в изменение параметров графических элементов шаблона, иначе нас просто не поймут и, соответственно, денег не заплатят.

Рассматривая поле заголовка, можно понять следующее: графический объект располагается в окружении минимум трех графических элементов, размеры которых необходимо изменить, не меняя размеров самой графической области. Размеры области, отводимой под рисунок, задаются в самом файле `template.css` — в контейнере, что расположен чуть выше ранее отредактированных записей:

```
div#header_r {
  height: 90px;
  padding-left: 370px;
  padding-right: 30px;
  padding-top: 25px;
  overflow: hidden;
  text-align: left;
}
```

Здесь изменим высоту области размещения `height` с 90 на 120 пикселов, то есть получится следующее:

```
div#header_r {
    height: 120px;
    padding-left: 370px;
    padding-right: 30px;
    padding-top: 25px;
    overflow: hidden;
    text-align: left;
}
```

Обновленная веб-страница будет выглядеть так же, как на рис. 9.77.

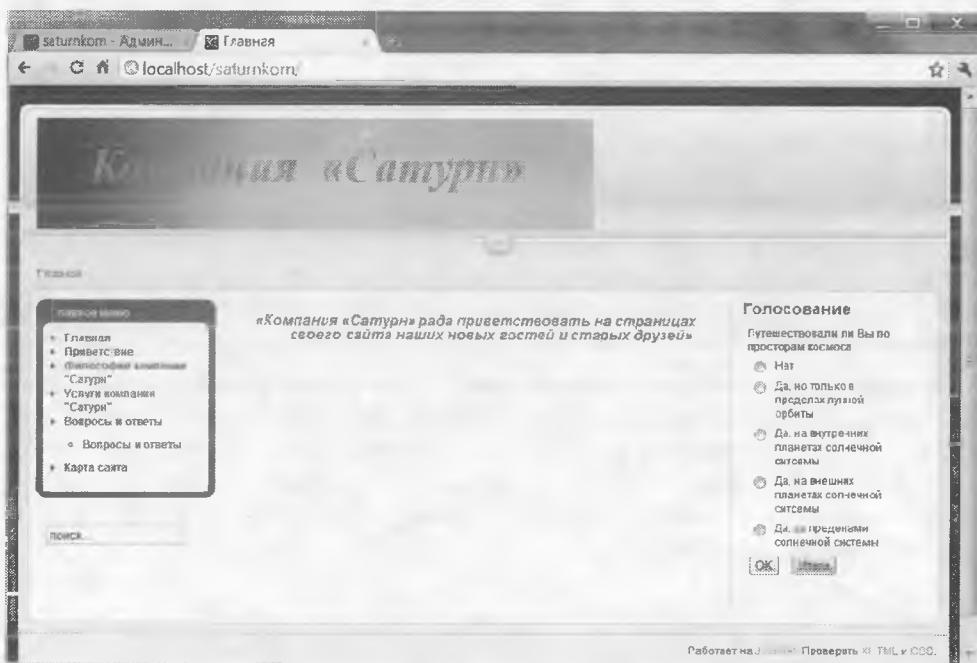


Рис. 9.77. Веб-страница с изменениями в области размещения графики

Как видите, осталось задать другие размеры для графических объектов шаблона, и он примет почти завершенный вид.

Необходимые для изменения файлы находятся в каталоге `rhuk_milkyway\images\blue`. Это файлы `mw_header_t.png`, `mw_header_t_l.png` и `mw_header_t_r.png` (рис. 9.78).

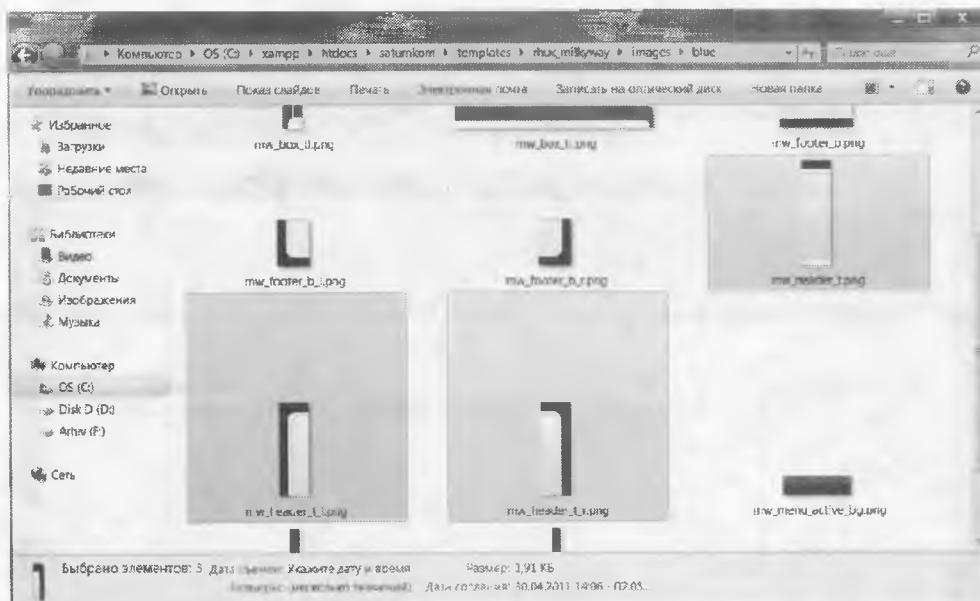


Рис. 9.78. Графические файлы, определяющие необходимую нам область шаблона

Будем по одному открывать эти файлы в программе Adobe Photoshop и указывать новую высоту изображения, выбирая пункт меню **Изображение** ▶ **Размер изображения** (рис. 9.79). Зададим 200 пикселов для файла **mw_header_t_.png** и по 180 пикселов для файлов **mw_header_t_l_.png** и **mw_header_t_r_.png** соответственно. При этом не забывайте сохранять изменяемые документы.

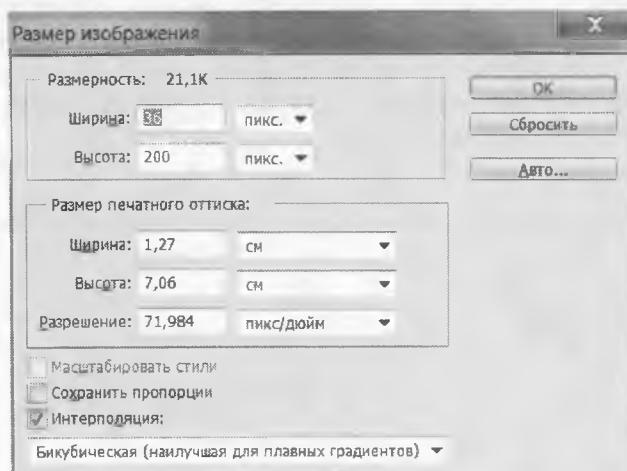


Рис. 9.79. Окно установки размеров изображения программы Adobe Photoshop

Затем, как всегда, обновляем страницу сайта на соседней вкладке и любуемся результатами нашего труда (рис. 9.80).

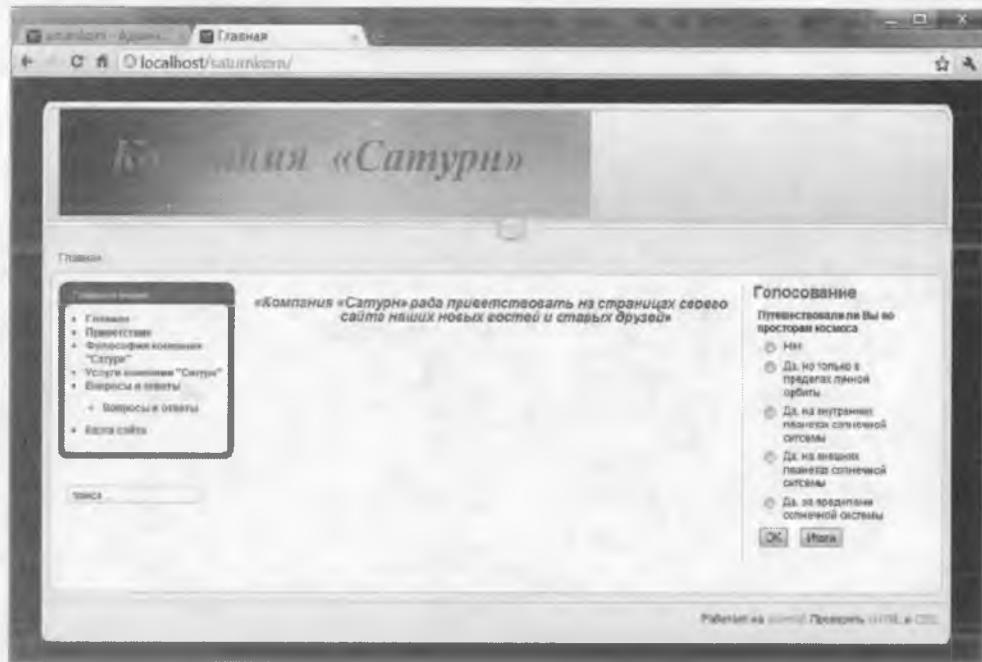


Рис. 9.80. Веб-страница после внесения изменений в графические файлы шаблона

Теперь нужно отлепить верхнюю границу нашего рисунка от границы графической области. Для этого опять обратимся к редакции нашего файла `template.css` и внесем некоторые изменения в контейнер, определяющий размещение надписи внутри шаблона. Изменим записи `margin-left: 30px` и `margin-top: 25px`, устанавливающие положение левого верхнего угла нашего графического объекта. Таким образом, исходная надпись с содержимым контейнера имеет вид:

```
div#logo {
    position: absolute;
    left: 0;
    top: 0;
    float: left;
    width: 600px;
    height: 120px;
    background: url(..../images/nadpis.gif) 0 0 no-repeat;
```

```
margin-left: 30px;  
margin-top: 25px;  
}  
}
```

Результат изменения содержимого контейнера выглядит, как:

```
div#logo {  
position: absolute;  
left: 0;  
top: 0;  
float: left;  
width: 600px;  
height: 120px;  
background: url(..../images/nadpis.gif) 0 0 no-repeat;  
margin-left: 25px;  
margin-top: 27px;  
}
```

После этого можно щелкнуть на значке Сохранить в окне редактирования шаблона и после обновления страницы увидеть результат изменения (рис. 9.81).

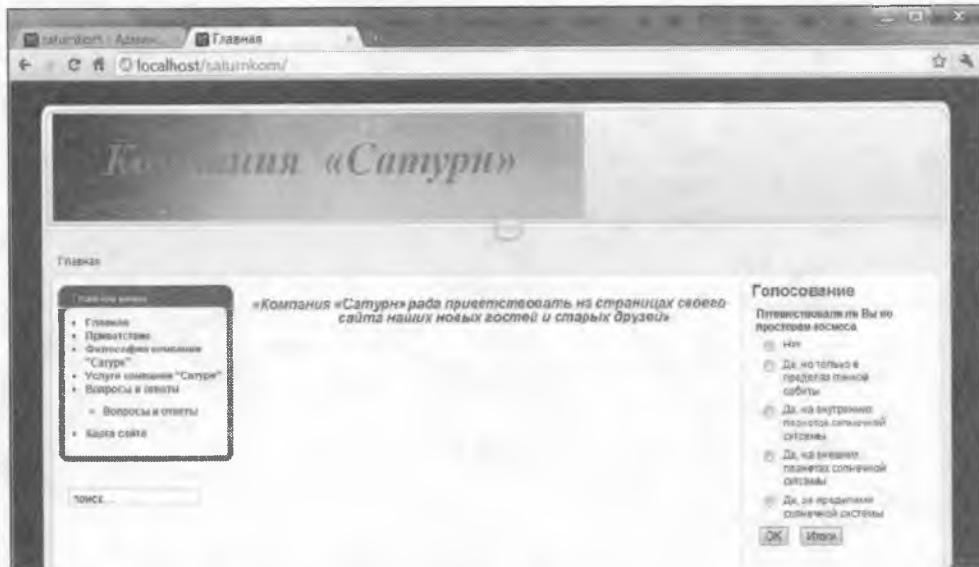


Рис. 9.81. Окончательный вид веб-страницы после внесения изменений в шаблон

Продолжим редактирование нашего шаблона и уберем с сайта надпись Работает на Joomla!. Проверить XHTML и CSS, которая обычно располагается в правом нижнем углу или по центру веб-страницы (рис. 9.82). Она указывает на то, что сайт создан на основе Joomla!, и является гиперссылкой на официальные зеркала CMS. Это не хорошо и не плохо, но излишняя информация, не относящаяся к тематике сайта, обычно отвлекает внимание, попросту отягощая созданный ресурс.

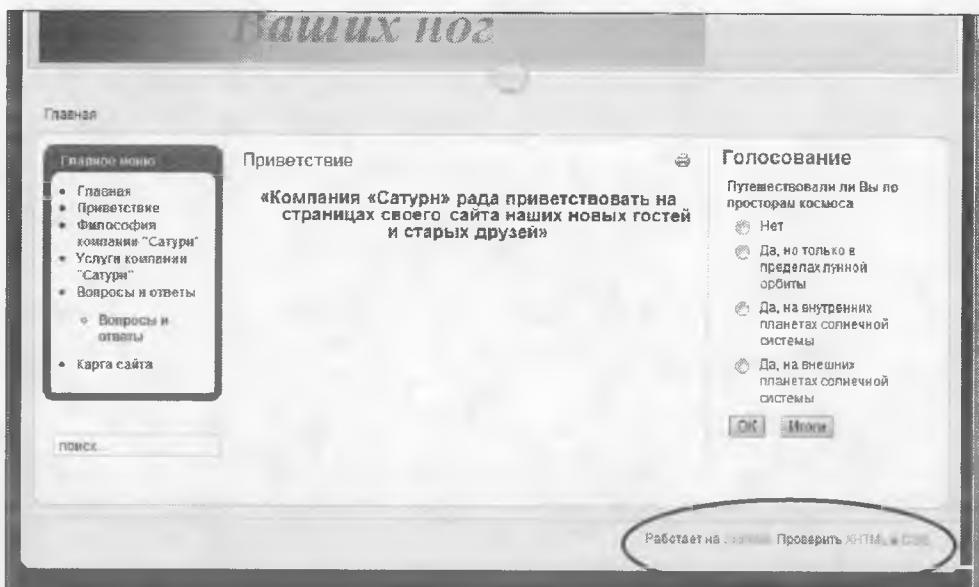


Рис. 9.82. Надпись Работает на Joomla!. Проверить XHTML и CSS

Для удаления обозначенного текста в шаблоне нам опять придется вернуться к окну редактирования шаблонов, но придется выбрать значок Редактировать HTML вместо значка Редактировать CSS. В открывшемся окне Редактор HTML файла шаблона в нижней части файла нужно найти текст следующего содержания (рис. 9.83):

```

<p id="power_by">

<?php echo JText::_('Powered by') ?>

<a href="http://www.joomla.org">Joomla!</a>.

<?php echo JText::_('Valid') ?>

<a href="http://validator.w3.org/check/referer">XHTML</a>

<?php echo JText::_('and') ?>

<a href="http://jigsaw.w3.org/css-validator/check/referer">CSS</a>.

</p>
  
```

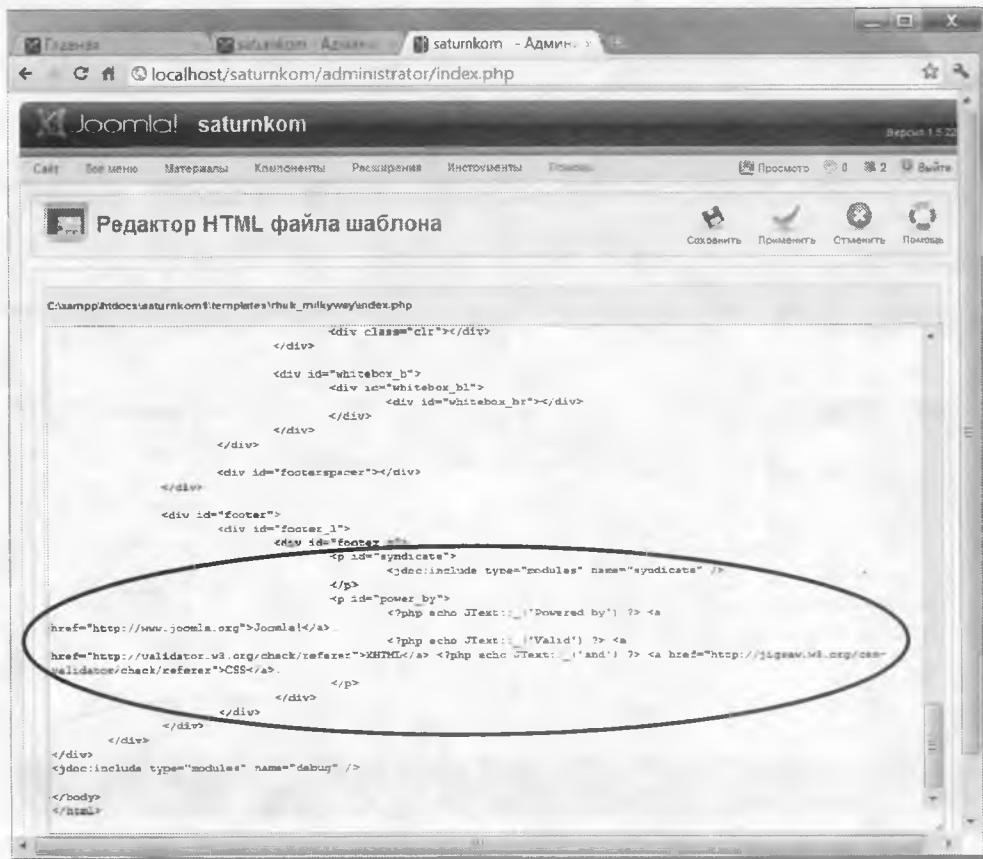


Рис. 9.83. Окно Редактор HTML файла шаблона

Данный текст представляет собой именно то, что нам нужно удалить со страницы шаблона. При необходимости его можно модифицировать согласно нашим целям, но для этого следует владеть языком программирования PHP и основательно разбираться в основных принципах создания шаблонов для CMS Joomla!. Пока же в наших силах только удалить данный текст и сохранить изменения на странице, щелкнув на значке Сохранить. Перейдем на вкладку с изображением веб-страницы сайта, обновим содержимое и увидим результат, которого добивались (рис. 9.84).

Как видно, действия, которые выполнялись при модификации установленного по умолчанию в CMS Joomla! шаблона, не являются чем-то очень сложным и чрезесчур заумным. Изменять готовые шаблоны под себя можно, конечно, и простым методом «тыка», вопрос только в том, сколько времени это займет. Если не обладать основательными знаниями в области построения каскадных таблиц стилей или хотя бы не держать на столе справочник по CSS, то гораздо проще найти требуемый или очень подходящий шаблон в

Интернете, чем серьезно модифицировать не совсем подходящий, но очень понравившийся жене начальника шаблон сайта на основе CMS Joomla!.

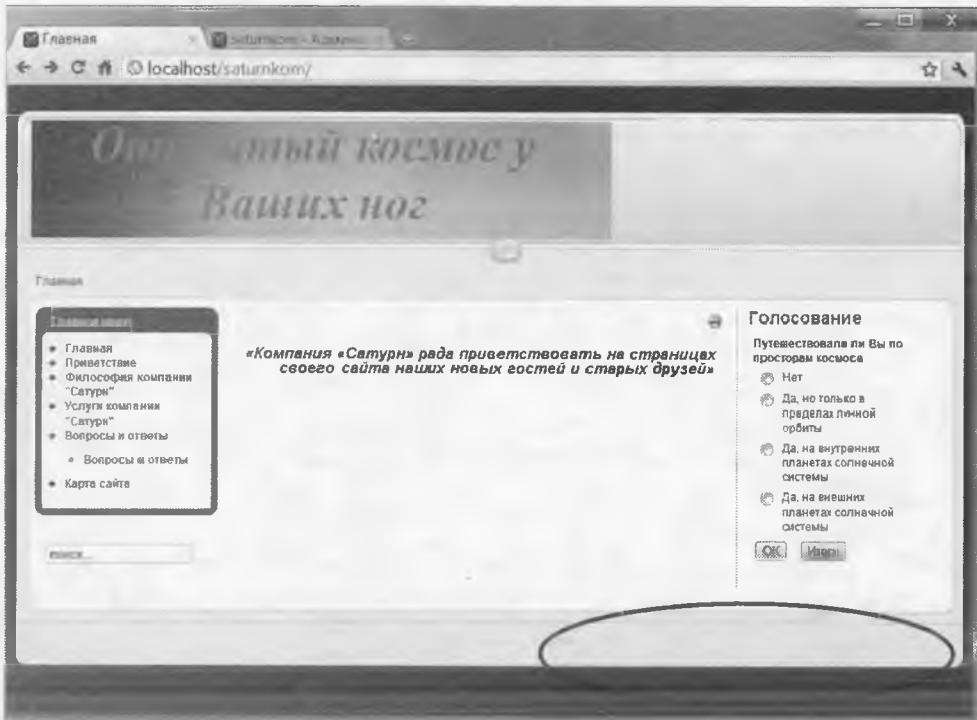


Рис. 9.84. Окончательный вид главной веб-страницы сайта после модификации шаблона `rhuk_milkyway`

9.7. Добавление нетекстовых объектов на веб-страницу

Расположение мультимедийных объектов в CMS Joomla!. Поскольку все основные, кроме одной, операции по наполнению сайта и изменению его внешнего вида рассмотрены выше, пришло время изучить способ размещения на веб-страницах графических или мультимедийных объектов.

Из подготовленных заранее мультимедийных объектов у нас остался неиспользованный Flash-ролик, находящийся в файле `saturn.swf`. Именно его мы поместим на веб-страницу.

Как любой графический объект, файл `saturn.swf` сначала нужно загрузить на веб-сервер. Для этого можно в очередной раз воспользоваться FTP-клиентом, например FileZilla, а также задействовать мощности самой системы управления содержимым Joomla!, открыв ее медиаменеджер.

Для запуска медиаменеджера следует щелкнуть кнопкой мыши на значке **Медиа менеджер**, расположеннном на панели управления (рис. 9.85), либо выполнить пункт меню **Сайт** ▶ **Медиа менеджер**.

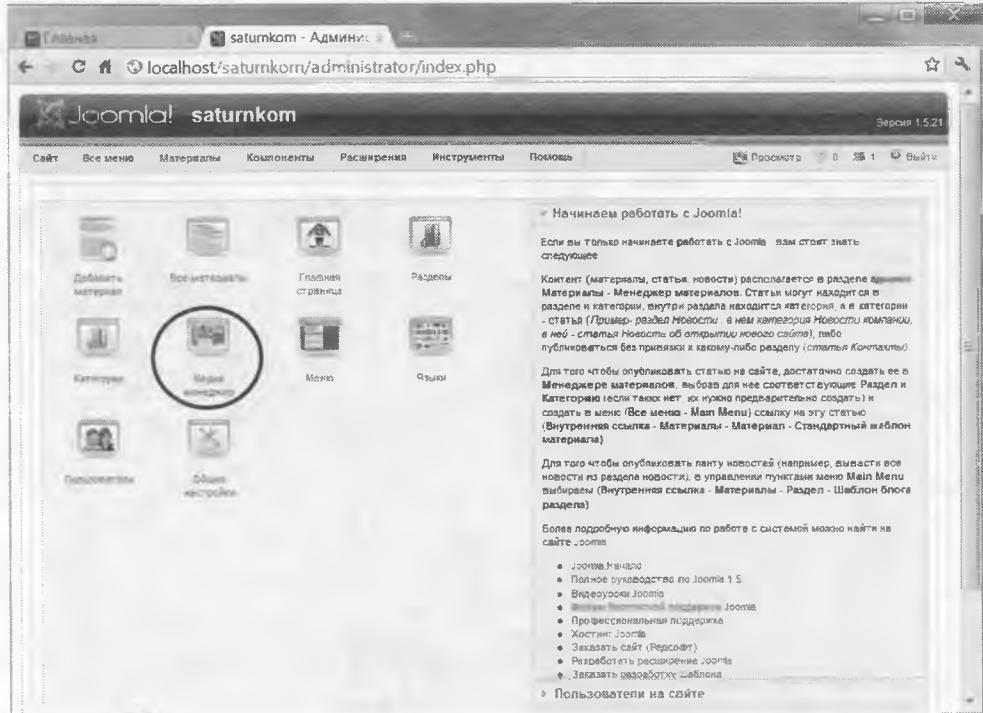


Рис. 9.85. Панель управления CMS Joomla!

В открывшемся окне медиаменеджера (рис. 9.86) в нижней его части расположено поле **Выберите файл**, в котором следует указать имя нашего файла. Затем нужно нажать кнопку **Загрузить** и дождаться сообщения, подтверждающего загрузку файла на веб-сервер.

Убедившись, что в списке имеющихся файлов появился и загруженный нами документ **saturn.swf**, можно перейти к следующему шагу.

Откроем окно менеджера материалов (рис. 9.87), для чего выполним команду **Материалы** ▶ **Менеджер материалов**.

Далее следует выбрать статью **Приветствие** и щелкнуть на значке **Изменить** либо просто дважды щелкнуть на имени самой статьи, чтобы открылось окно ее редактирования.

В окне редактирования статьи, как на панели редактора, так и под самим окном, есть значки, позволяющие вставить графический объект. Но все не так просто — согласно настройкам CMS Joomla! графические объекты — это только файлы именно зарегистрированных и именно графических

форматов, а SWF-файл, содержащий Flash-ролик, к ним никак не относится. Иначе говоря, если у вас простой графический файл, то достаточно щелкнуть на значке **Изображение**, как тут же откроется окно, позволяющее добавить на страницу рисунок из ранее загруженного файла (рис. 9.88).

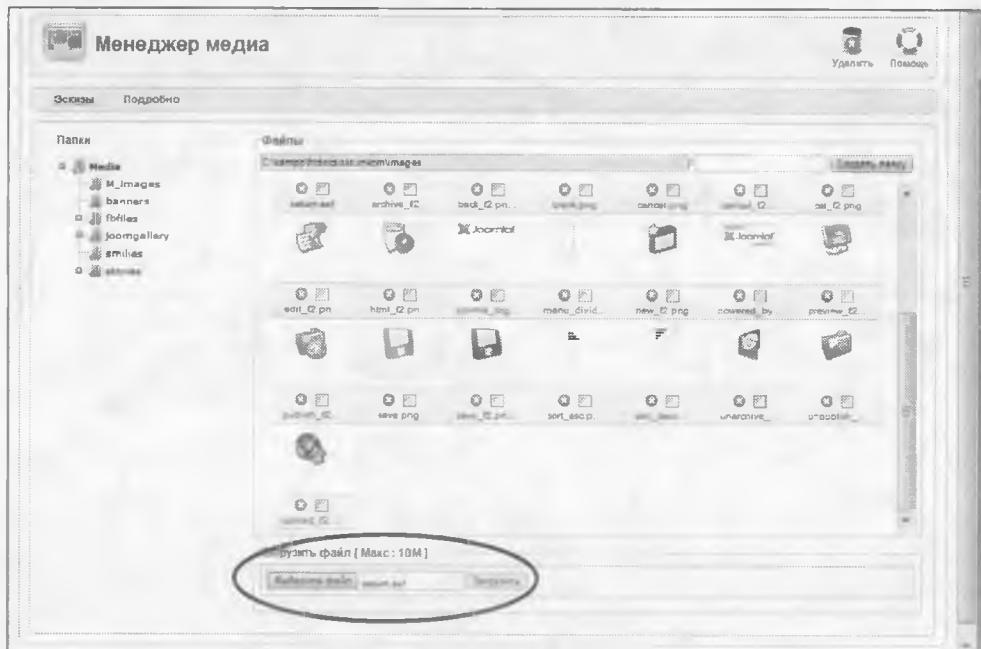


Рис. 9.86. Окно медиаменеджера

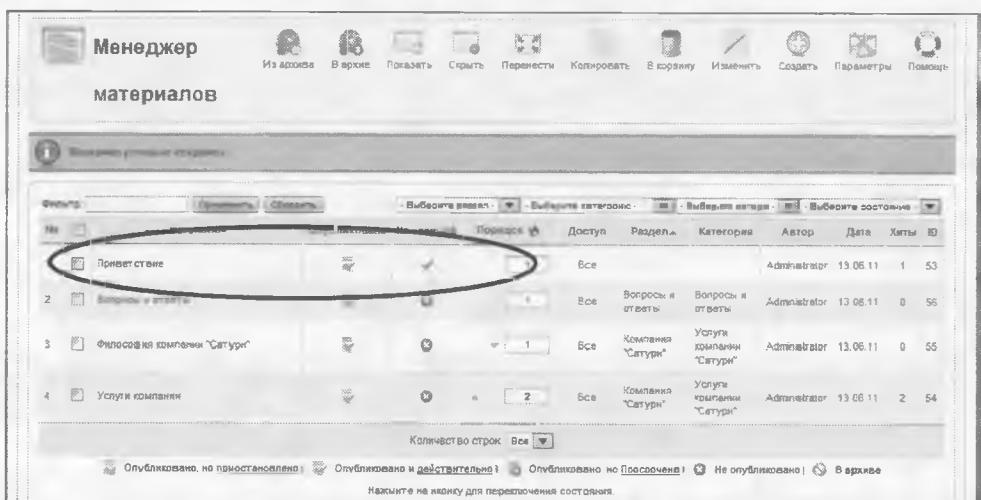


Рис. 9.87. Окно менеджера материалов

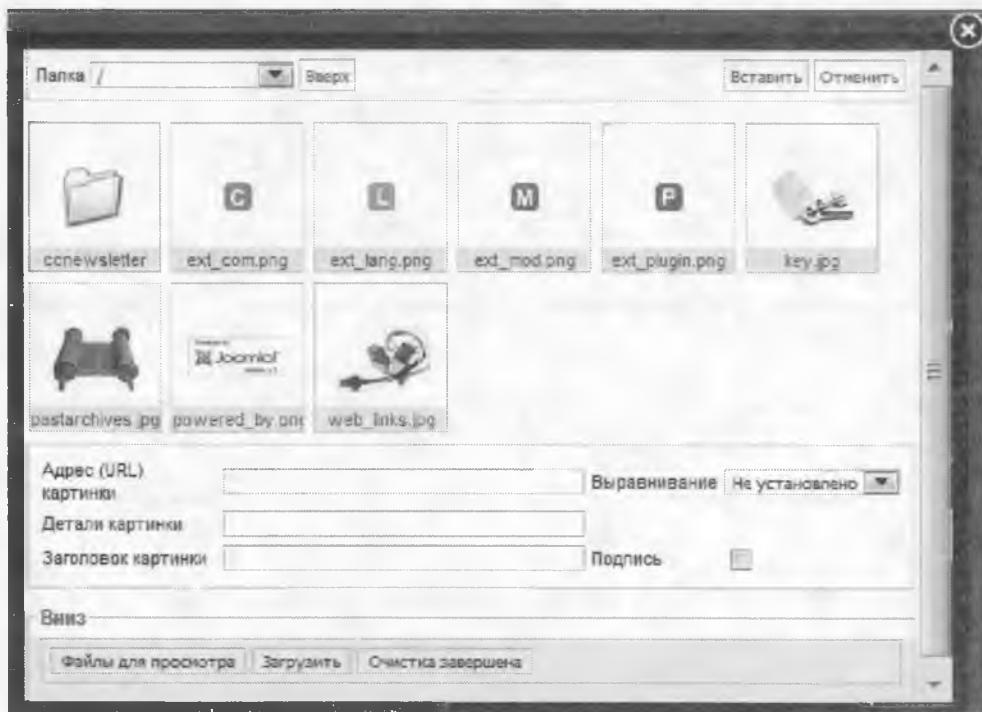


Рис. 9.88 Окно вставки изображений на веб-страницу

Здесь достаточно указать имя файла и заполнить соответствующие поля, почти так же, как мы делали в главе 2, когда вставляли на страницу сайта графические объекты, обозначающие заголовок картинки, всплывающее описание, выравнивание и прочие параметры. Затем нужно нажать кнопку Вставить, и изображение появится на веб-странице.

Поскольку наш SWF-файл для CMS Joomla! не является графическим, то так просто вставить его нельзя. Существует два метода вставки SWF-файлов на веб-страницу. Первый метод достаточно долгий — нужно найти в Сети модуль, который позволяет размещать Flash-ролики на веб-страницах, затем подключить его и только потом с его помощью устанавливать на веб-страницу ролик. Все это долго и сложно. Поэтому будем использовать второй метод — добавим на веб-страницу наш Flash-ролик с помощью HTML-кода. Теперь рассмотрим все более подробно.

Сначала вставим на страницу обычную таблицу, необходимую для раздельного размещения текста и картинки. Для этого на панели редактора необходимо щелкнуть на значке Вставка новой таблицы и в появившемся окне Вставить/Изменить таблицу задать такие значения: Столбцы — 2, Строки — 1, Ширина — 99 % (рис. 9.89).

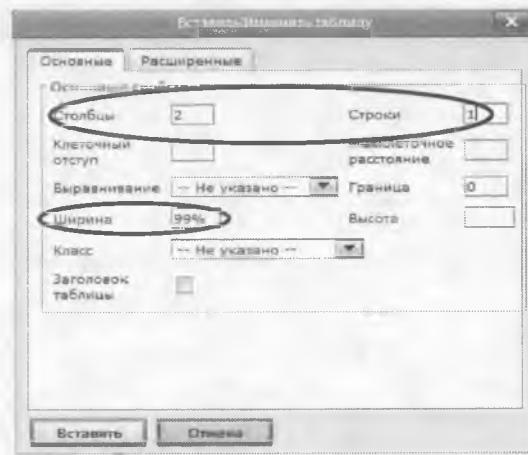


Рис. 9.89. Окно Вставить/Изменить таблицу с новыми параметрами

После вставки таблицы необходимо перенести в ее левую ячейку наш текст, а затем щелкнуть на значке Редактировать код (рис. 9.90).

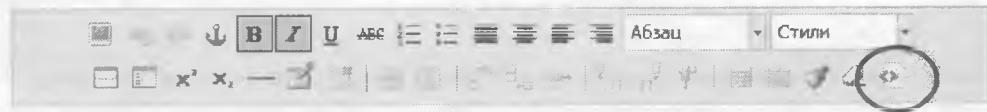


Рис. 9.90. Панель инструментов редактора

В появившемся окне Расширенный редактор кода нужно перейти в сам HTML-код, вернее, в ту его часть, где теги `<td>` и `</td>` показывают начало и конец правой ячейки таблицы (рис. 9.91), и вставить следующий фрагмент HTML-кода, описывающий наш графический объект:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=6,0,40,0"
width="100" height="100"><param name="width" value="100" /><param
name="height" value="100" />
<param name="src" value="images/saturn.swf" /><embed type="application/x-
shockwave-flash" width="100"
height="100" src="images/saturn.swf"></embed></object>
```

После этого нажмем кнопку Обновить, и можно лицезреть измененную страницу в окне редактора статьи (рис. 9.92).

Затем, как всегда, переходим на вкладку с самой веб-страницей и обновляем ее. Как видно на рис. 9.93, наш Flash-ролик внедрен и при этом неплохо смотрится.

Расширенный редактор кода

Подсветка № Номера строк № Перенос строк № Размер шрифта №

Форматирование

```

1 <h2 style="text-align: center;">  
<i><br><br><br></i></h2>
2 <table style="width: 59%;" border="0">
3   <tbody>
4     <tr>
5       <td><span style="color: #333333; font-family: Helvetica, Arial, sans-serif; font-size: 12px;">
6         <h2 style="text-align: center;"><em><strong>«Компания «Сатурн» рада
7         приветствовать на страницах своего сайта наших новых гостей и старых
8         друзей!</strong></em></h2>
9       </span></td>
10
11       <object classid="clsid:D27CDB6E-AE6D-11cf-96B3-444553510000"
12         codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash2.cab#version=6,0,0,0"
13         width="100" height="100"><param name="width" value="100" /><param name="height" value="100" />
14         <param name="src" value="images/saturn.swf" /><embed type="application/x-shockwave-flash"
15         width="100"
16         height="100" src="images/saturn.swf"></embed></object>
17     </tr>
18   </tbody>
19 </table>
20 <p><em><strong><br /></strong></em></p>

```

Рис. 9.91. Расширенный редактор кода

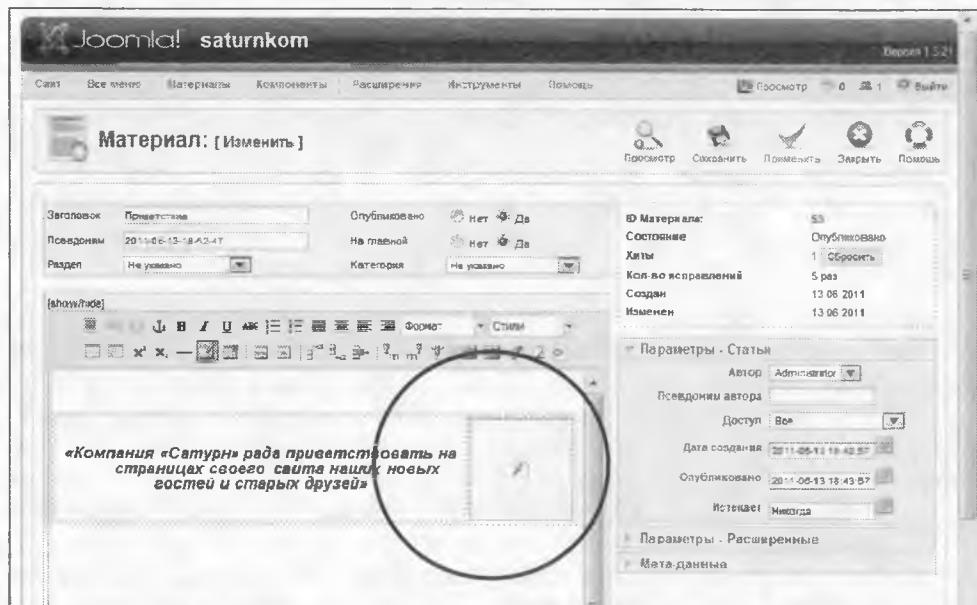


Рис. 9.92. Область вставки на веб-страницу Flash-ролика

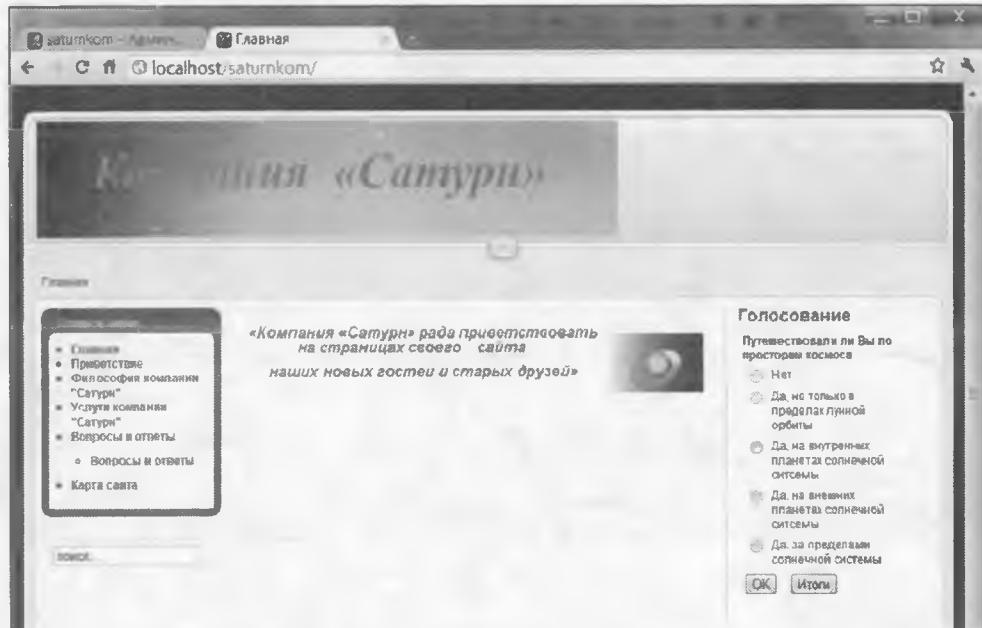


Рис. 9.93. Результат внедрения Flash-объекта на веб-страницу

9.8. Подключение дополнительных расширений к сайту

Подключение форума в системе Joomla!

Если вы не собираетесь организовывать, а затем и модерировать форум на своем сайте, то смело пропускайте этот раздел. Если же вы задержали свое внимание именно здесь, то можете узнать, каким образом подключить компонент форума в системе управления содержимым Joomla!.

Сам по себе модуль форума занимает немного места, довольно быстро устанавливается и легок в обслуживании. Но насколько он необходим на вашем сайте? Сколько вы готовы уделять ему времени и вкладывать в него сил? Ни для кого не секрет, что наибольшее количество форумов, сопровождающих различные сайты, зачастую являются мертвыми, то есть не содержат практически никаких сообщений. На многих форумах без должного управления происходят целые баталии с применением ненормативной лексики в качестве тяжелого вооружения, группы сообщений, а иногда и целые ветки форумов вдруг становятся перегруженными рекламными сообщениями абсолютно не ваших услуг. В общем, поддержание форума на плаву требует гораздо больше временных затрат, чем поддержание сайта. Создав сайт, нужно его регулярно обновлять, а если ассортимент

услуг неизменен, то и про обновление можно забыть на определенный промежуток времени. С форумом этот номер не пройдет — немодерируемый форум рано или поздно скатится в такие низины, что уважающий себя пользователь там не покажется. Таким образом, первое, что вы должны для себя определить, где взять ежедневно несколько десятков минут для управления форумом.

Если вы все-таки решились на создание и поддержание форума для своего сайта и готовы вкладывать в него время и средства, то рано или поздно удача улыбнется вам. Ведь правильно управляемый форум — это ни много ни мало контентное наполнение вашего сайта и основа вашего веб-сообщества единомышленников. Как правило, грамотно управляемые форумы по объему своего содержимого, а зачастую и по качеству, значительно превышают те сайты, при которых они созданы.

Прежде чем устанавливать компонент форума CMS Joomla!, необходимо его получить. Для этого можно воспользоваться либо самим сайтом проекта Joomla! <http://www.joomla.org/>, либо одним из многочисленных русскоязычных зеркал проекта, в нашем случае <http://www.joomla.ru/>.

Для получения файлового архива с компонентом форума для CMS Joomla! нужно перейти на страницу загрузки расширений <http://www.joomla.ru/download/viewcategory/33.html> (рис. 9.94).

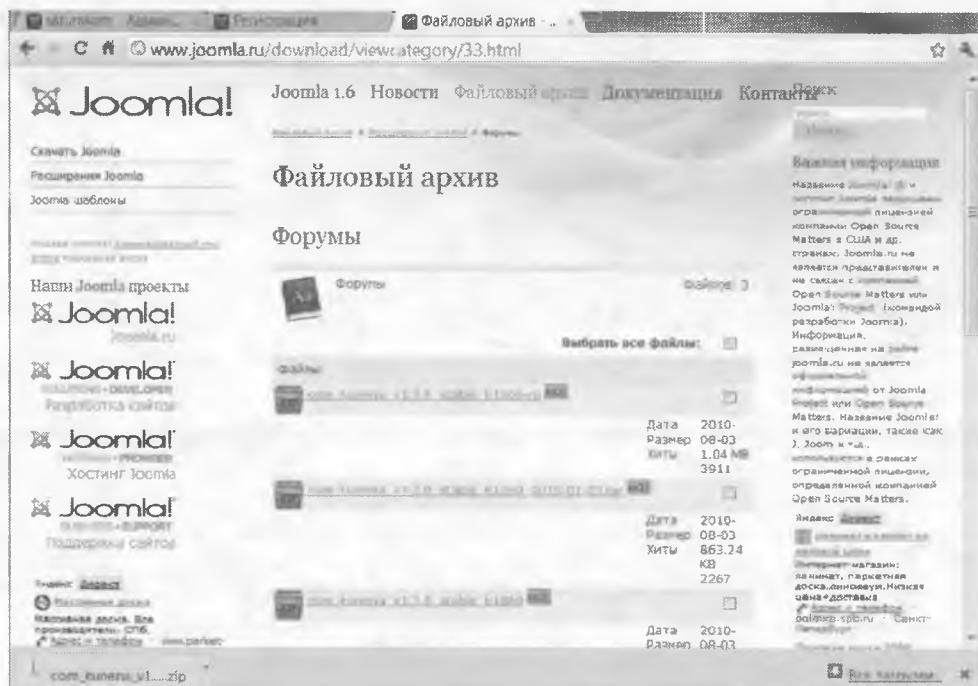


Рис. 9.94. Веб-страница загрузки файловых архивов, содержащих компонент форума для CMS Joomla!

Как видно из рисунка, на время написания книги доступно три файла с русифицированными компонентами форума: один в формате TAR для UNIX-совместимых операционных систем и две версии ZIP-архивов, которые могут быть открыты как и в UNIX-совместимых операционных системах, так и из-под Windows. Поскольку компоненты написаны на языке программирования PHP, то абсолютно неважно, какая операционная система стоит на вашем персональном компьютере, ведь программа все равно будет выполняться не на нем, а на удаленном веб-сервере. Однако наш эксперимент пока проводится на локальном компьютере под управлением Windows, и XAMPP установлен также в Windows, поэтому, чтобы не искушать судьбу, скачаем ZIP-архив.

Внимание! Обязательно загляните в скачанный файловый архив с компонентом форума и изучите *все* находящиеся там текстовые файлы на предмет дополнительных инструкций по установке данного расширения. Это позволит избежать дальнейших проблем при установке или обслуживании форума.

Далее перейдем к панели управления нашим сайтом и выберем пункт меню **Расширения** ▶ **Установить** или **Удалить** (рис. 9.95).

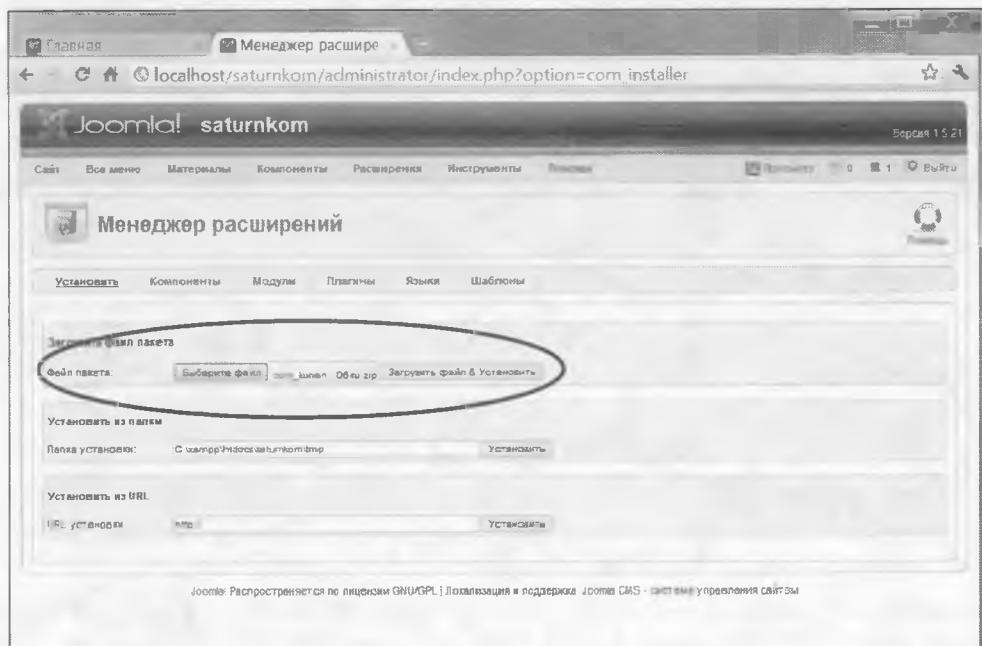


Рис. 9.95. Окно менеджера расширений

Нажав кнопку **Выберите файл**, укажем имя и расположение скачанного файлового архива с компонентом форума, а затем нажмем кнопку **Загрузить**

файл & Установить. После правильно выполненной загрузки окно менеджера расширений должно выдать сообщение о том, что загрузка состоялась (рис. 9.96).

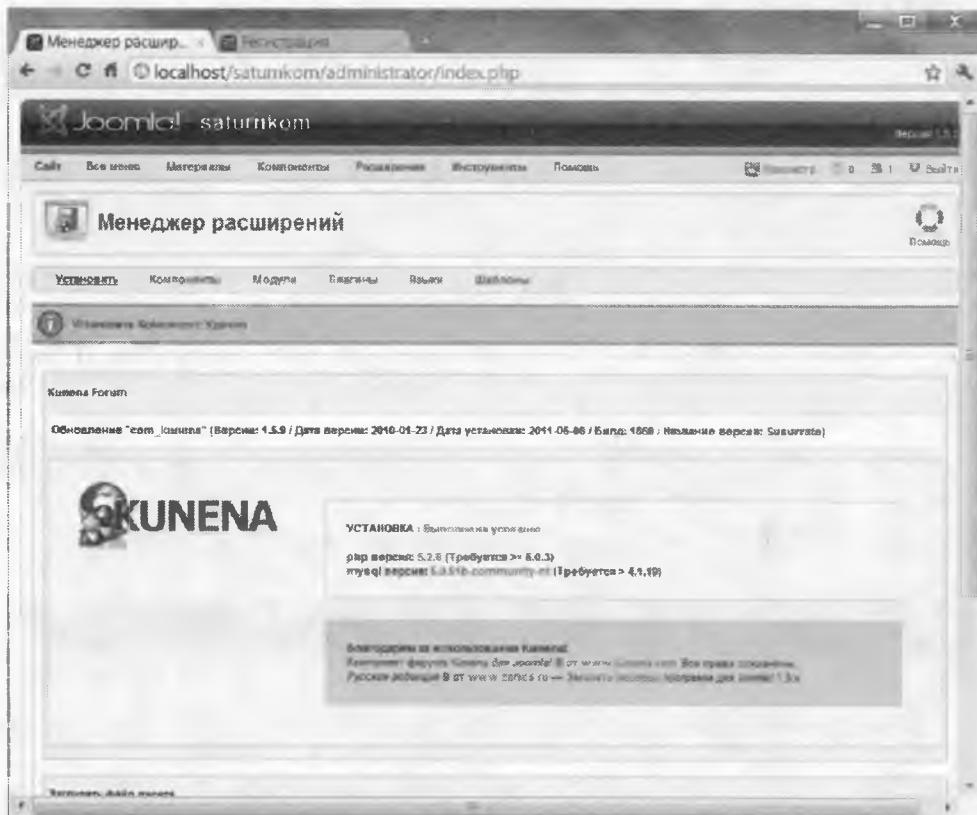


Рис. 9.96. Сообщение менеджера расширений об успешной установке компонента форума KUNENA

Поскольку компонент успешно установлен, необходимо создать пункт меню, который и будет вызывать форум на выполнение со страниц нашего сайта. Для этого выполним пункт меню **Все меню > Главное меню**. В открывшемся окне Пункты меню щелкнем на значке **Создать**. В следующем окне необходимо выбрать тип **Kunena Forum** (рис. 9.97).

В последнем окне меню следует указать заголовок и псевдоним создаваемого пункта меню (рис. 9.98).

Теперь, как всегда, щелкнем на значке **Сохранить** и, перейдя на соседнюю вкладку браузера, обновим ее. Как видно на рис. 9.99, в главном меню сайта появился пункт меню **Форум**, который работает, если на нем щелкнуть кнопкой мыши.



Рис. 9.97. Окно привязки компонента форума Kunena Forum к пункту меню

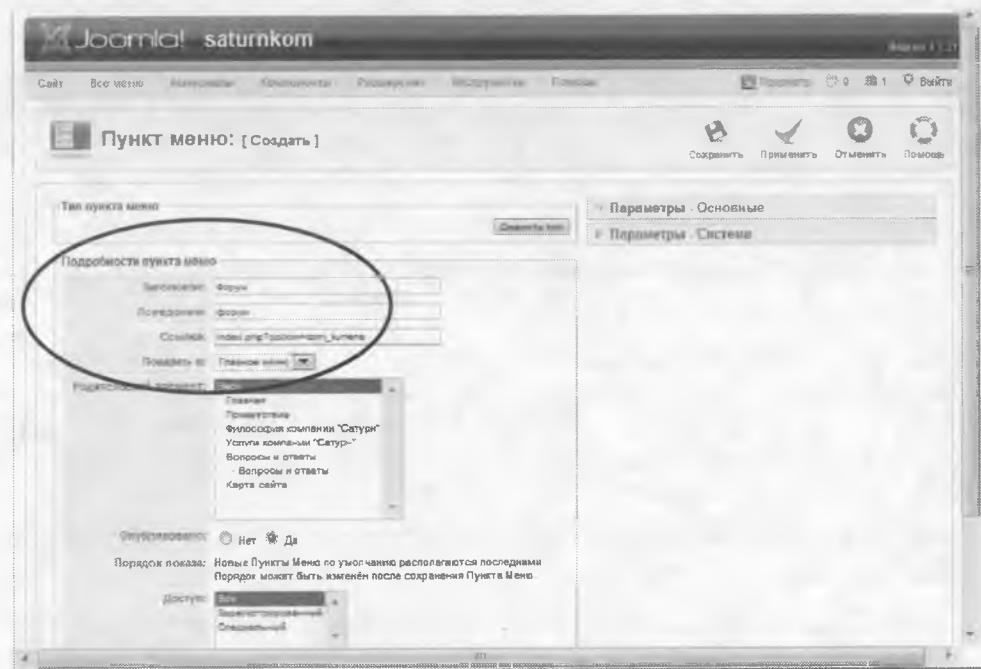


Рис. 9.98. Завершающее окно создания пункта меню, где указывается заголовок и псевдоним пункта меню

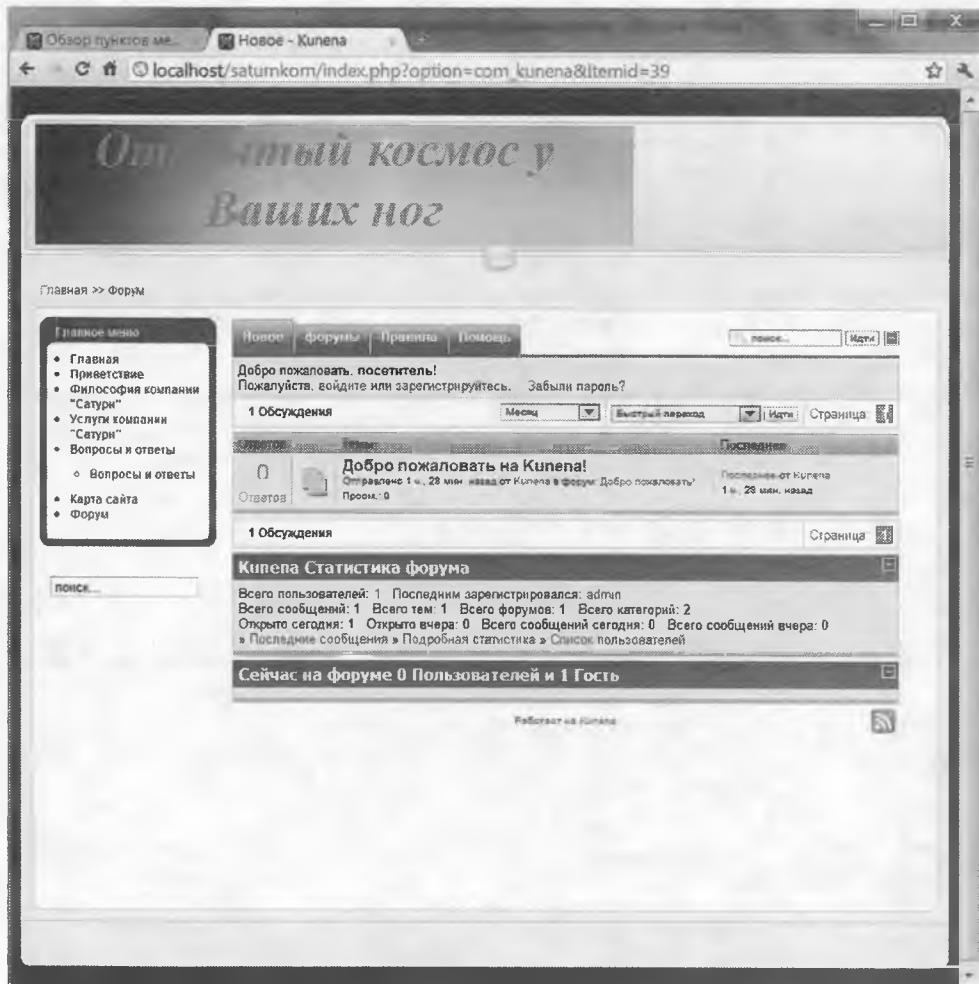


Рис. 9.99. Форум Kunena, установленный и запущенный на сайте на основе CMS Joomla!

Для модерирования форума, его настройки и полного управления необходимо вызвать оснастку Панель управления, поэтому выберите следующие пункты меню: Компоненты ▶ Kunena Forum (рис. 9.100).

В результате выполненных действий загрузится окно Панели управления компонентом форума Kunena (рис. 9.101).

Как видно на рис. 9.101, установленный на нашем сайте компонент с форумом интегрирован в систему Joomla! и имеет интуитивно понятный интерфейс. Разобраться с настройкой и управлением форума не составляет особого труда, так как принцип управления любым компонентом CMS Joomla! аналогичен управлению самой CMS.

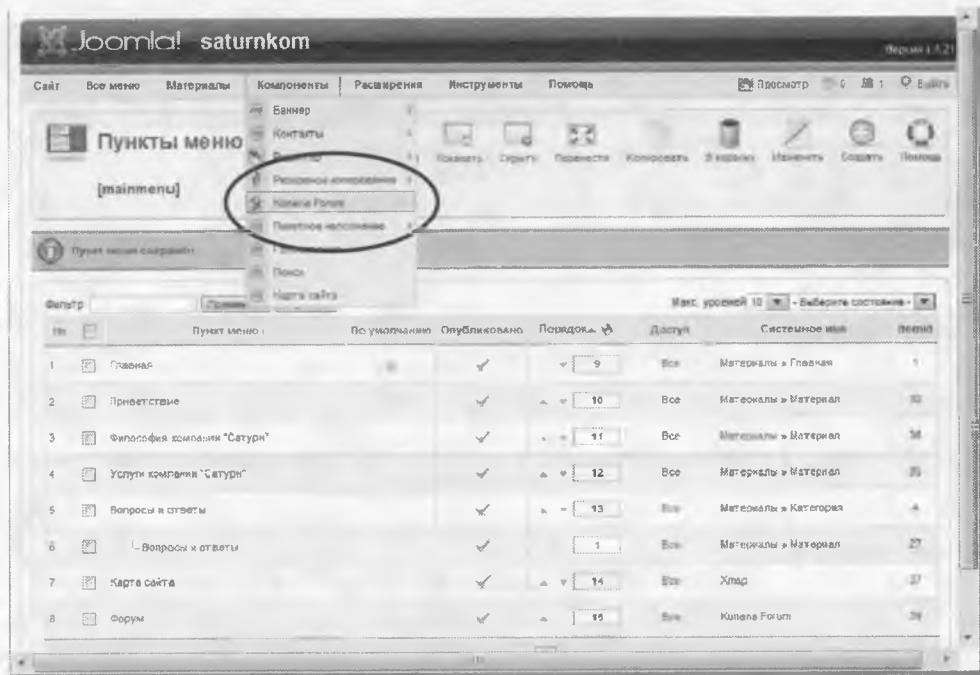


Рис. 9.100. Меню для вызова Панели управления установленным форумом Кипена

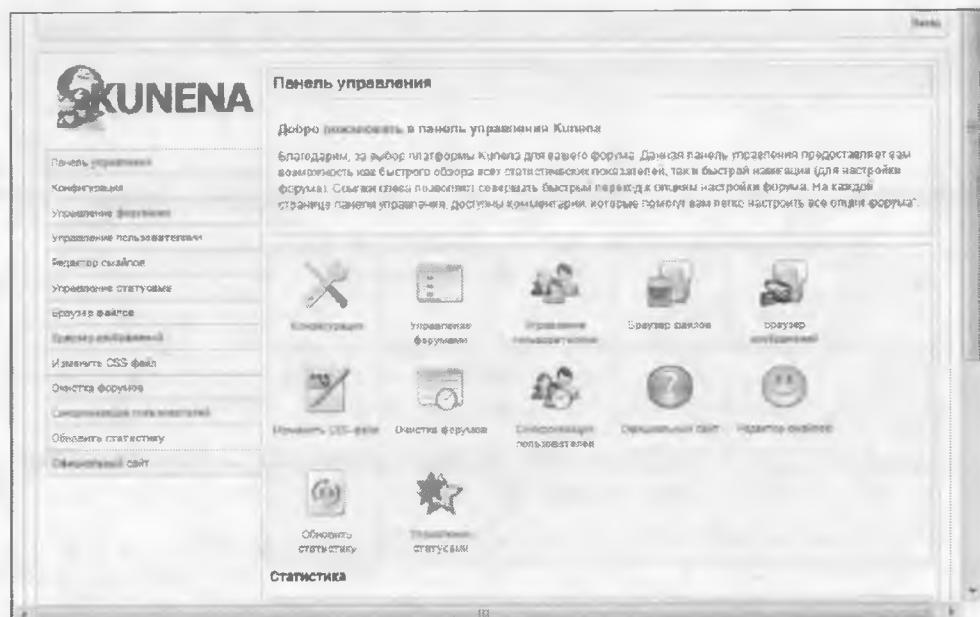


Рис. 9.101. Окно Панели управления форума Кипена

Возможность добавления комментариев к статье

Продолжим знакомство с расширениями, доступными для CMS Joomla!. Один из показателей востребованности статей посетителями вашего сайта — обратная реакция. Конечно, обсуждать материал статьи можно и на форуме, но вполне возможно, что представленная информация просто не заинтересует посетителей. Если обсуждение статьи принимает интересный и, самое главное, полезный для сайта характер, то его достаточно легко перенести в отдельную, специально созданную ветку форума.

Согласно модульному построению CMS Joomla! все дополнительные возможности, помимо базовых, установленных в шаблоне по умолчанию, вводятся администратором сайта по необходимости. Поэтому компонент, организующий обсуждение статьи, также необходимо скачать, установить и настроить в системе Joomla!. Процедура скачивания компонента была описана выше, поэтому не будем на ней останавливаться. С сайта www.joomla.ru нужно скачать файл расширений для создания комментариев к статье, на момент написания книги я выбрал `com_jcomments_v2.1.0.0.zip`. Далее, выполнив команду **Расширения ▶ Установить** или **Удалить**, перейдем в окно менеджера расширений, где выберем ранее скачанный архив, содержащий расширение, и нажмем кнопку **Загрузить файл & Установить**. По окончании установки компонента нужно нажать кнопку **Далее** (рис. 9.102).

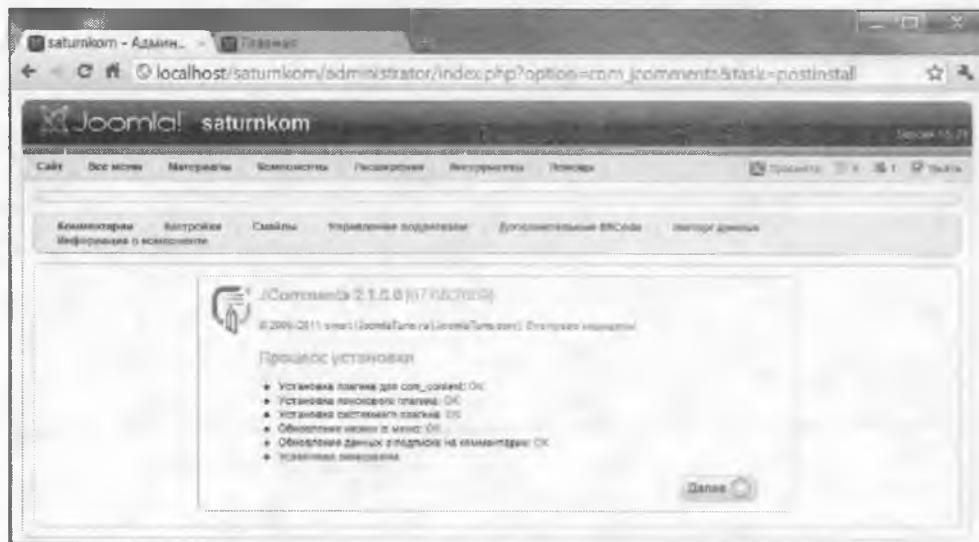


Рис. 9.102. Установка компонента JComments 2.1.0.0

В открывшемся окне настройки компонента необходимо выделить те категории, статьи которых вы собираетесь разрешать комментировать. Для выделения группы категорий следует удерживать нажатой клавишу **Ctrl** и щелкать кнопкой мыши на заголовках категорий (рис. 9.103).

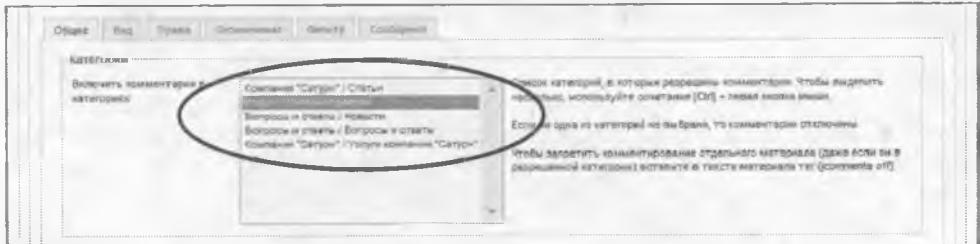


Рис. 9.103. Выделение имен категорий, для статей которых разрешены комментарии

Совершив описанные действия, необходимо перейти на вкладку **Вид**, что расположена рядом с вкладкой **Общие**, и в появившемся окне в самом низу указать, как выводить на веб-страницу форму комментария: сразу в виде формы или сначала в виде ссылки (рис. 9.104).

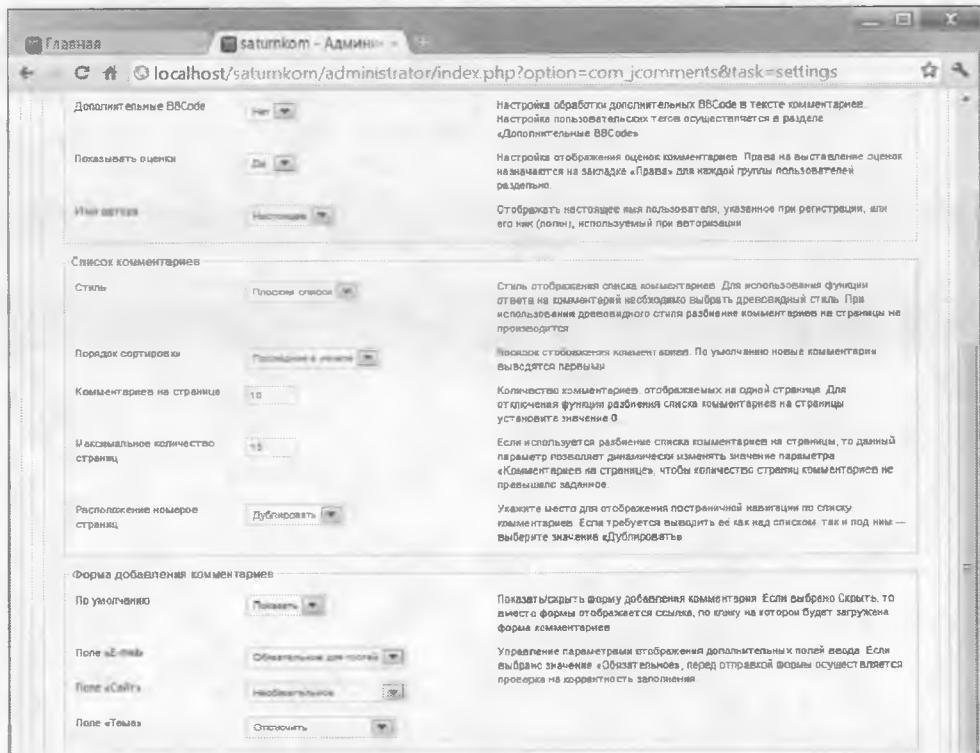


Рис. 9.104. Окно вкладки Вид

Перейдя на вкладку **Права** (рис. 9.105), определим для различных групп пользователей права, которые они будут иметь при создании комментария к статье, размещенной на веб-странице.

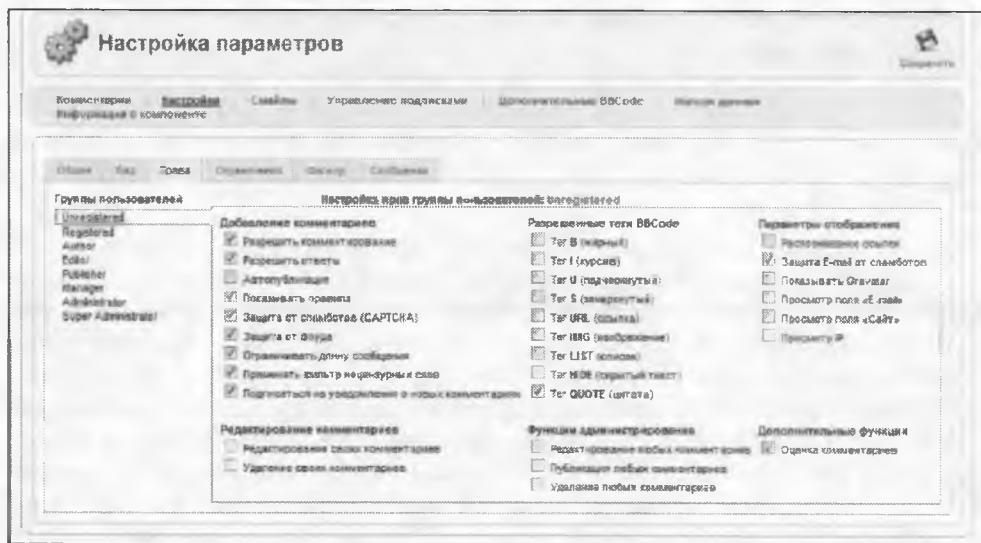


Рис. 9.105. Вкладка Права окна настройки комментариев

Далее, используя тот же алгоритм, выполним необходимую настройку дополнительных вкладок, после чего перейдем на вкладку с веб-страницей, обновим ее и получим результат, как на рис. 9.106, где изображена статья с открытой формой добавления комментария к ней.

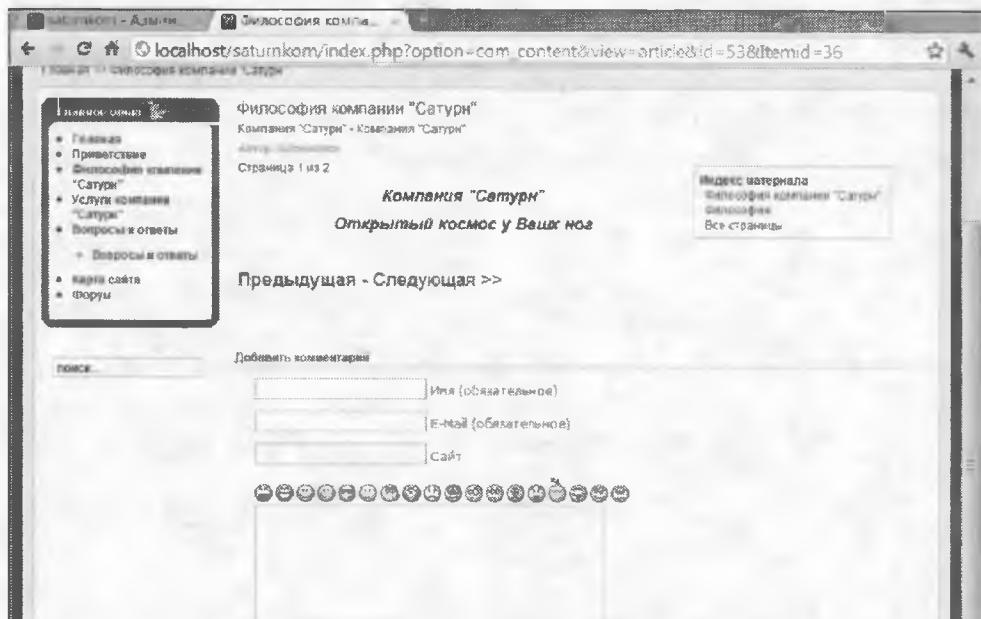


Рис. 9.106. Форма добавления комментария к статье

Фотогалерея на сайте под управлением CMS Joomla!

Рассмотрев принцип работы системы управления контентом Joomla! и разобравшись со способами расширения ее базовых возможностей, было бы непростительной ошибкой не вспомнить еще одно дополнительное расширение — фотогалерею.

Компонентов, выполняющих функцию фотогалереи, на момент написания книги уже несколько десятков, какую выбрать — решать вам самим. В принципе, каждый компонент фотогалереи обладает как базовыми, так и расширенными функциями. Все они поддерживают как одиночную загрузку контента, так и блочную — несколько фотографий в виде одного ZIP-архива. Поэтому говорить о различных особенностях того или иного компонента CMS Joomla!, а тем более сравнивать их, не имеет смысла ввиду малого объема данной книги. Гораздо проще найти в Сети сравнительный обзор, составленный специалистами. Я же ограничился файловым архивом *joomgallery_1_5_6_1_stable_build_20101205.zip*, а также языковым расширением для данного пакета в виде архива *ru-ru.com_joom*, скачанного все с того же сайта www.joomla.ru.

Фотогалерея устанавливается так же, как и все ранее устанавливаемые компоненты, — через менеджер расширений. При этом обязательно нужно помнить, что сначала устанавливается сама фотогалерея — файл *joomgallery_1_5_6_1_stable_build_20101205.zip*, затем сразу же документ русификации *ru-ru.com_joom*, и только после этого можно переходить к настройке фотогалереи и загрузке изображений.

Вся настройка фотогалереи выполняется с помощью меню Компоненты ► JoomGallery (рис. 9.107).

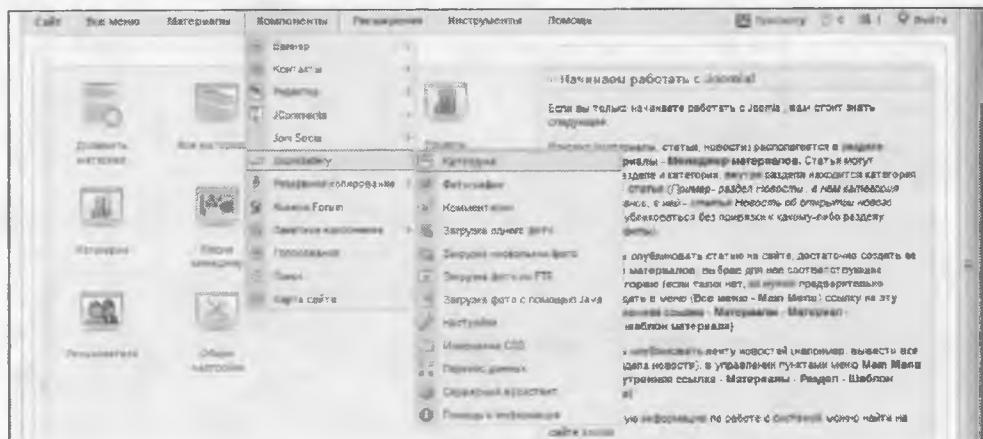


Рис. 9.107. Вызов панели управления фотогалереей через меню Компоненты ► JoomGallery

В результате будет первоначально открыто окно Ассистента категорий панели управления фотогалереей, где для дальнейшего размещения фотографий нужно создать хотя бы одну категорию.

Следующим нашим шагом будет загрузка фотографий на сервер. В этом случае компонент фотогалереи предлагает два способа. Первый заключается в компоновке всех загружаемых графических файлов в один ZIP-архив, который затем загружается на веб-сервер через вкладку **Загрузка нескольких фото**. Там программа фотогалереи самостоятельно распакует его и разместит фотографии. При этом следует помнить, что программа работает только с ZIP-архивами. Второй — более простой, но в то же время более привычный способ — это загрузка фотографий по одной. Для этого необходимо воспользоваться вкладкой **Загрузка одного фото**.

В области **Выберите фотографию** нажимаем кнопку **Выберите файл**, указываем файл, содержащий фотографию, затем ниже определяем категорию, к которой будет относиться данное изображение, и в обязательном порядке указываем ее название. После этого просто нажимаем кнопку **Загрузить**.

Поскольку фотогалерея успешно создана и в нее даже помещено несколько изображений, ее пора предоставлять для ознакомления сетевой общественности. А для этого, как помните, необходимо создать пункт в главном меню сайта и связать его с фотогалереей.

Итак, выполнив команду **Все меню** ▶ **Главное меню**, щелкнем на значке **Создать** и окажемся в окне создания пункта меню. Щелкнув кнопкой мыши на объекте **JoomGallery**, перейдем в расширенное окно выбора типа меню.

Здесь нам следует задать тип представления фотогалереи. Не торопитесь при выборе — тип меню всегда можно поменять. После этого перейдем к заключительному окну настройки. В этом окне требуется заполнить поля, определяющие заголовок и псевдоним заголовка для создаваемого пункта меню. После щелчка на значке **Сохранить** и обновления веб-страницы, находящейся на соседней вкладке браузера, получим прописанный пункт меню **Фотогалерея**. С его помощью можно получить доступ к файлам изображений, размещенным на веб-сервере (рис. 9.108).

Используя описанный алгоритм работы с расширениями для CMS Joomla!, вы сможете легко расширять возможности вашего сайта в необходимом направлении, вне зависимости от того, чем вы занимаетесь — социологическими исследованиями или розничной торговлей подгурниками. Самое главное при работе с расширениями — не теряться, искать наиболее оптимальные и приемлемые для своих нужд, причем искать достаточно регулярно, если вы не удовлетворены имеющимися расширениями, поскольку Joomla!-сообщество постоянно растет и, как следствие, появляются все новые и новые расширения для различных сайтов.



Рис. 9.108. Пункт меню Фотогалерея, открывающий доступ к фотографиям на веб-сервере

9.9. Что необходимо знать веб-мастеру при управлении сайтом на основе CMS

Поисковая оптимизация сайтов на основе CMS Joomla!

Пришло время сказать несколько слов о поисковой оптимизации сайтов, созданных в системе Joomla!. В этой CMS есть несколько параметров для оптимизации сайтов, созданных в ней для удобства перемещения поискового паука по веб-страницам. Поисковая оптимизация содержимого (SEO, Search Engine Optimization), расположенного на ваших веб-страницах, — это процесс подготовки веб-страниц таким образом, чтобы они занимали наиболее приближенные к верхнему уровню места в рейтингах поисковых машин, когда выполняется поиск информации по определенным запросам.

CMS Joomla!, как помните, динамически генерирует страницы сайта, что является огромным плюсом, так как изменения, вносимые при модификации, сразу же применяются для всего сайта целиком. Однако этот плюс имеет еще и минусы. Например, жесткая двухуровневая иерархия сайта не позволяет веб-мастеру изменить структуру сайта, как это можно сделать на классических статических ресурсах. И главная проблема всех систем управления содержимым — генерация URL веб-страниц.

Рассмотрим, что происходит. В классических статических сайтах каждая веб-страница представляет собой отдельный файл, расположенный в строго определенном месте. На файл указывают четкие статические гиперссылки, по которым поисковому пауку двигаться — сплошное удовольствие, а уж с индексированием статических страниц в базе данных и слов нет, как все славно происходит. В то же время сгенерированные CMS Joomla! веб-страницы не только не имеют четкого расположения на жестком диске, поскольку в виде файлов вообще не существуют, но и свои URL представляют в неудобочитаемом виде и для человека, и для поискового робота. По этим причинам, не взирая на ценность контента динамического сайта, рейтинг веб-страниц, имеющих статический URL, будет всегда выше рейтинга страниц с генерируемым URL.

Что же делать?

На самом деле разработчики предусмотрели способ превращения генерируемого URL в статический путем включения дополнительных параметров. Единственное но: данные параметры не включены по умолчанию, потому что при их включении CMS Joomla! динамически перезаписывает URL на веб-сервере, а это не всегда поддерживается веб-хостерами.

Для активизации SEF-параметров (Search Engine Friendly URL) — генерации дружественного URL для поисковых роботов — необходимо выбрать пункт меню Сайт ▶ Общие настройки. При этом откроется окно Общие настройки (рис. 9.109).

В области Установки Search Engine Optimization нужно задать параметры, которые вы выберете из трех предлагаемых.

- Включить SEO — система будет генерировать URL-адреса, сходные с URL статических веб-страниц.
- Использовать mod_rewrite — система будет генерировать URL-адреса чуть иначе. Использование данного параметра предусмотрено, если веб-сервер не находит запрашиваемую страницу и выдает ошибку 404 (файл не найден).
- Добавить суффикс к URL — к URL веб-страницы добавляется суффикс в виде номера, что, в свою очередь, позволяет не включать предыдущий параметр. Однако к этому достоинству добавляется и недостаток — не все веб-серверы провайдеров поддерживают данную возможность.

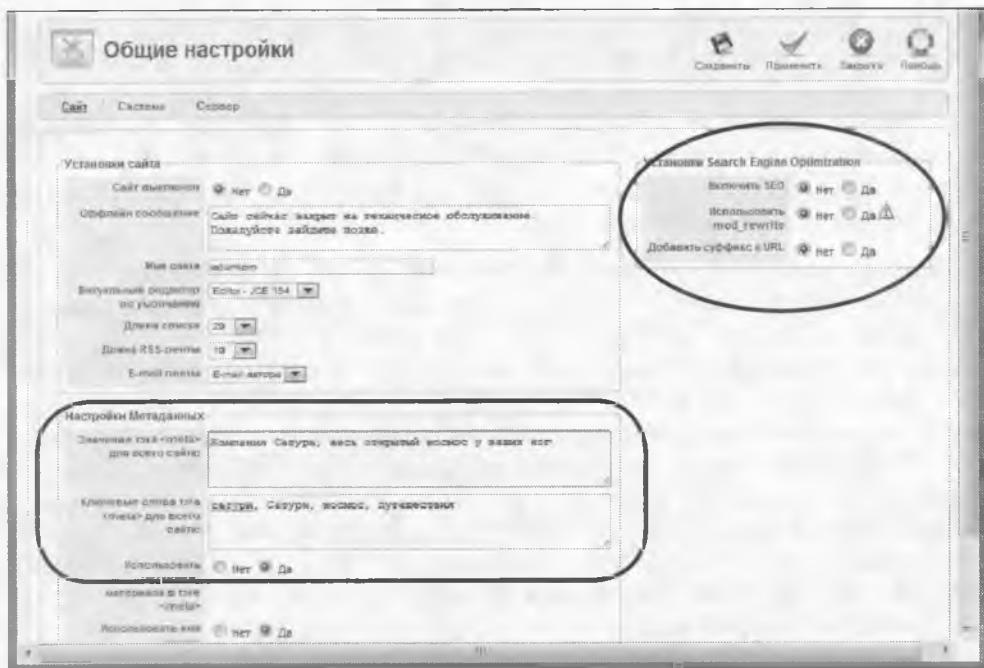


Рис. 9.109. Окно Общие настройки с областями, информация в которых влияет на поисковую оптимизацию сайта

Для увеличения релевантности ваших веб-страниц при формировании запросов не стоит пренебрегать заголовками страниц, наборами ключевых слов и областью метатегов веб-страницы. Поля, содержащие эти данные, можно и даже нужно заполнять, так как количество ключевых слов, входящих в область метатега и область описания, способны существенно повысить рейтинг страницы у поисковика. При этом следует учитывать, что создатели поисковых роботов предвидели ситуацию злоупотребления ключевыми словами и создали механизмы, понижающие, а в некоторых случаях и вовсе блокирующие статьи, если набор ключевых слов содержит слова, не входящие в текст статьи. Область параметров, определяющих состояние метаобласти редактируемой статьи, показана на рис. 9.110.

Кроме того, для SEO-оптимизации сайта можно использовать дополнительные расширения, устанавливаемые специально для этого, выполнять настройку веб-сервера, являясь, конечно, его владельцем, создавать карты сайта. В нашем случае карта создана автоматически при загрузке и установке демоданных. Можно создавать дополнительные SEF-шаблоны для CMS Joomla!, но об этом лучше всего узнать из документации к самой системе управления содержимым.

В конце главы, посвященной CMS Joomla!, мы затронем, пожалуй, одну из важнейших тем — резервное копирование созданного сайта.

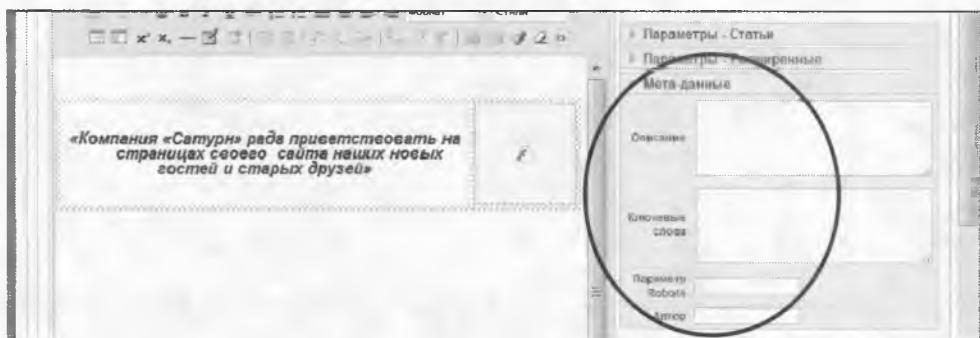


Рис. 9.110. Область описания метаданных в окне редактируемой статьи

Резервное копирование

При резервном копировании сайта, созданного в любой CMS, следует сохранять как копии файлов, находящиеся на веб-сервере, так и сам контент сайта, расположенный не в файлах на веб-сервере, а в таблицах базы данных SQL-сервера. Файлы сохраняются с помощью таких же алгоритмов сохранения, как и при резервном копировании классического статического сайта.

Для создания резервной копии вашего сайта, расположенного на удаленном хостинге, необходимо подключиться к нему через FTP-клиент, например уже знакомый нам FileZilla, и скачать файлы сайта в специально отведенную для этого папку на жестком диске локального компьютера. Если же вы сами являетесь владельцем веб-сервера, то эта процедура еще более упрощается — создавайте архивные файлы каталога, в котором расположен сайт, доступным для вашей операционной системы способом. Например, для операционной системы Windows это оснастка *Архивация и восстановление файлов* в Панели управления (рис. 9.111).

Для сохранения контента, находящегося в таблицах SQL-сервера, придется обращаться непосредственно к самому SQL-серверу и с помощью дополнительных инструментов создавать резервные копии таблиц. Обычно на веб-серверах провайдеров для организации доступа веб-мастеров к таблицам SQL-сервера устанавливают приложение phpMyAdmin. С его помощью можно быстро создать резервную копию или восстановить контент сайта. Для этого необходимо перейти по указанному в договоре с провайдером адресу — к веб-странице, содержащей окно управления phpMyAdmin. В нашем случае, поскольку мы учимся на локальном компьютере, это будет адрес localhost/phpmyadmin (рис. 9.112).

Щелкнув на имени базы данных (в нашем случае saturnkom), перейдем в окно управления ею (рис. 9.113).

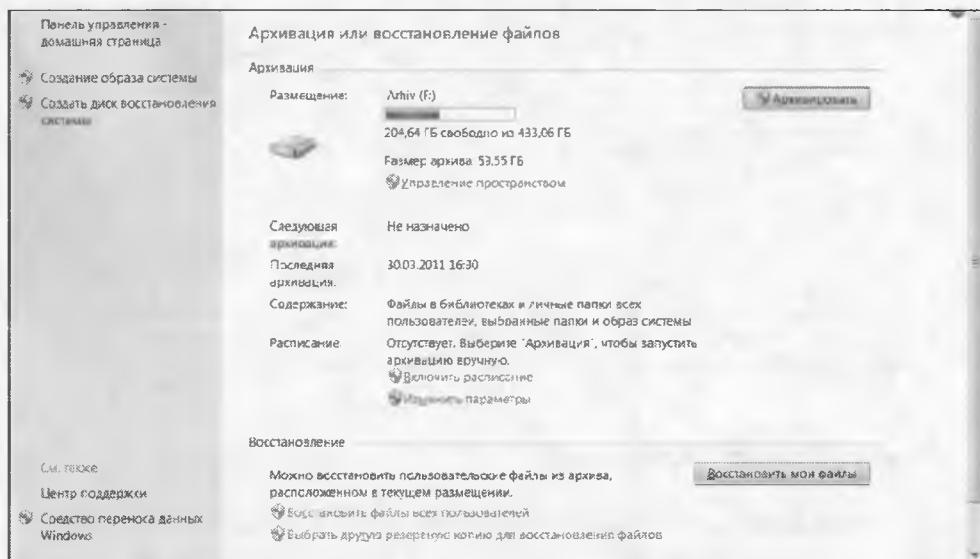


Рис. 9.111. Оснастка Архивация и восстановление файлов, расположенная на Панели управления Windows

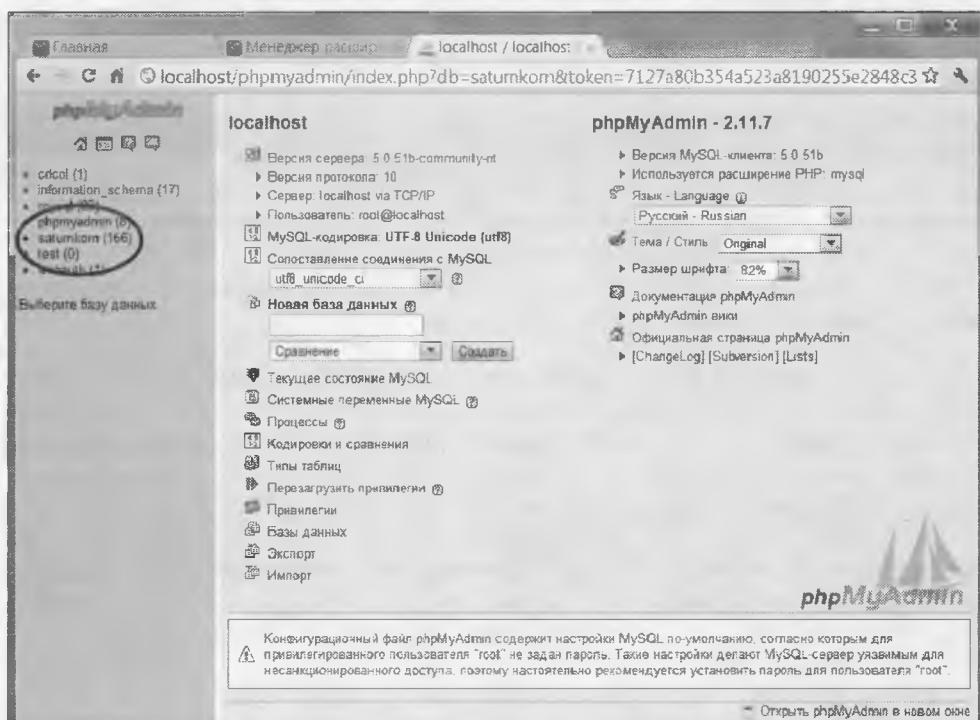


Рис. 9.112. Окно phpMyAdmin

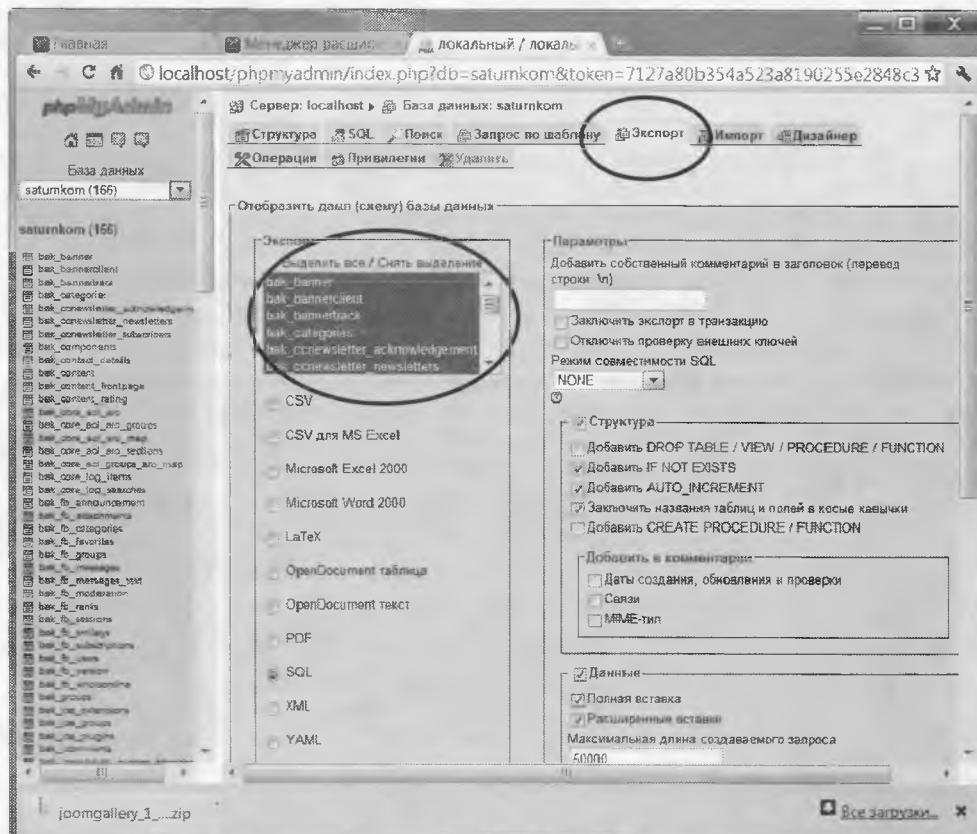


Рис. 9.113. Окно управления базой данных saturnkom, вкладка Экспорт

После этого нужно перейти на вкладку Экспорт и щелкнуть кнопкой мыши на Выделить все. Теперь спустимся вниз и установим флажок Сохранить как файл (рис. 9.114). Если база данных очень большая или плохой канал связи, то можно отметить файл как архив.

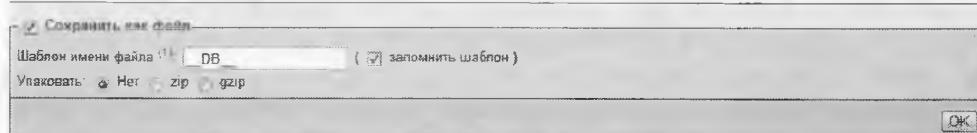


Рис. 9.114. Область окна управления базой данных saturnkom

Подтвердим свое намерение нажатием кнопки OK и через некоторое время получим в папке Загрузки браузера файл с контентом сайта, хранящимся в базе данных SQL-сервера.

Восстановление контента из файла резервного копирования также несложная процедура. Нужно только использовать вкладку Импорт окна управления базой данных, где и указать имя импортируемого файла (рис. 9.115).

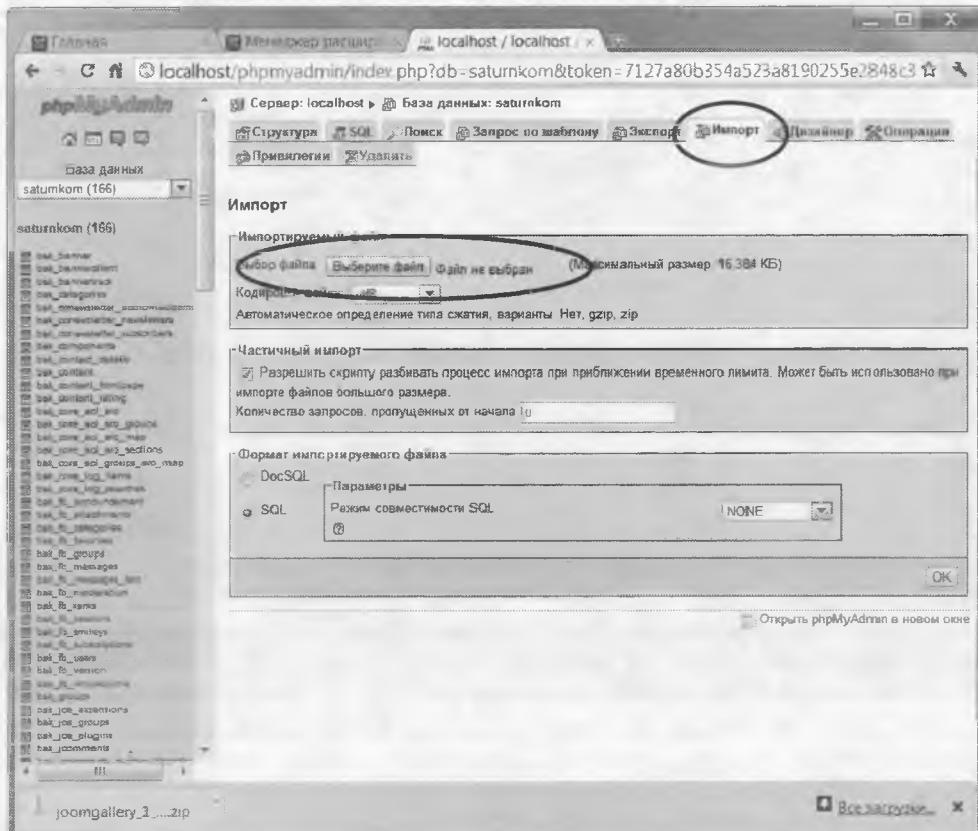


Рис. 9.115. Вкладка Импорт окна управления базой данных saturnkom

На этом, наверное, можно завершить краткое описание системы управления содержимым Joomla!. Как видно из приведенного в этой главе материала, создание сайтов на основе любых CMS, и в частности Joomla!, не представляет особого труда и не требует глобальных знаний в области веб-программирования. Достаточно понимать общие принципы построения сайта и изучить техническую документацию на приложения, с которыми вы собираетесь работать. Главное в сайтомостроении, как и в любой отрасли, связанной с созиданием, — четко знать, что ты хочешь создать, и очень хотеть это сделать.

Глава 10

О ПРОДВИЖЕНИИ САЙТОВ И ИХ РЕГИСТРАЦИИ В ПОИСКОВЫХ СИСТЕМАХ

10.1. Раскрутка и продвижение сайта

Для начала определимся в терминах. Что подразумевают под раскруткой и продвижением сайта?

Сейчас появилось очень много слов, кажущихся интуитивно ясными, но почему-то все их понимают по-разному. Продвинутый бренд, раскрученный сайт — что подразумевается под этими словами? На самом деле у всех понятий только одна общая характеристика — хорошо продаваемый по каким-либо причинам. Это может быть много раз показанная и поэтому въевшаяся реклама навязываемого дорогого бренда, характеризующего какие-нибудь (обычно несуществующие) качества его обладателя, или, что для нас сейчас важно, сайт, на который посетители приходят неоднократно. Итак, раскрученный сайт — это ресурс, который посещается большим количеством сетевых пользователей и обеспечивает неоднократное повторное возвращение старых пользователей, а также привлечение ими новых. Под продвижением сайта можно понимать комплекс разнообразных мер, направленных на привлечение наибольшего количества посетителей данного информационного ресурса.

Как видите, перед нами стоят две разноплановые задачи: сделать контент сайта востребованным и регулярно сообщать как можно более широкому кругу интернет-сообщества, что появился и существует сетевой ресурс, посетив который, вы больше ни на какие сайты не будете обращать внимания.

Решение первой задачи находится вне компетенции данной книги, но все же некоторые направления вы можете почерпнуть из своего сетевого опыта. Для этого ответьте на некоторые вопросы: на какие сайты вы заходите ежедневно, что вас туда привлекает, какую информацию помимо необходимой вам предоставляет посещаемый ресурс, по каким ссылкам или баннерам вы переходите и как часто. Таким образом, вы очертите сформированный круг своих интересов и увидите, какая необходимая информация привлекает вас на популярные сайты, а следовательно, потенциально сможете привлечь и других пользователей к вашему сетевому произведению. Кроме того, не стоит забывать, что содержимое должно регулярно обновляться и быть

читабельным на любом устройстве, материал необходимо излагать грамотным языком, нецензурных выражений быть не должно, а сленг желательно минимизировать или приводить пояснения в сносках, так как обычно он не всем понятен. Несмотря на сложность и обилие графики, ваш сайт должен легко загружаться и иметь простой дружественный интерфейс.

Решение второй задачи рассмотрим более подробно.

Вспомним, как мы ищем в Сети информацию по интересующему нас вопросу. Правильно — мы набираем ключевые слова в строке поиска. Следовательно, страница нашего сайта, содержащая информацию по какому-то вопросу, должна иметь как можно больше таких ключевых слов, которые будет прописывать потенциальный посетитель нашего сайта в запросе к поисковой машине. И тут возникает проблема. Конечно, можно заполнить всю страницу ключевыми словами, но из них очень трудно построить привлекательное содержимое. Как быть?

Решение данного вопроса достаточно тривиально. Кроме отображаемой части сайта есть еще и служебная — та часть, которая не отображается на экране, но учитывается поисковыми машинами при проведении поиска. Эта область страницы описывается тегами `<title>` и `<meta>`. Они предназначены для сохранении служебной информации и располагаются в начале кода страницы после тега `<head>`, при этом содержимое парного тега `<title> </title>` будет выведено в строке заголовка браузера и может содержать текст до 256 символов. Однако не увлекайтесь — не каждый экран может уместить строку такой длины, а читать и догадываться, что автор имел в виду под не поместившимися словами, — неблагодарное дело. Следует помнить, что ключевое слово, вставленное в заголовок страницы, автоматически повышает его рейтинг в поисковой системе. Содержимое же тега `<meta>` не будет отображено на экране. Рассмотрим пример использования описанных тегов.

```
<html>
  <head>
    <title>Пропишите сюда имя вашей страницы, сформулировав ее название с использованием ключевых слов</title>
    <meta http-equiv="KEYWORDS" content="Сюда внесите список ключевых слов и словосочетаний, записанных через запятую">
    <meta http-equiv="DESCRIPTION" content="Здесь вкратце опишите содержимое страницы">
    <meta http-equiv="AUTOR" content="Подтвердите свое авторство, написав имя">
  </head>
  <body>
    </body>
</html>
```

Таким образом, правильно записав содержимое метатегов для каждой страницы, вы официально увеличите количество необходимых слов на странице, что, в свою очередь, должно повысить ее релевантность в процессе обработки поисковым пауком. Не стоит забывать и то, что некоторые поисковые системы выполняют поиск с учетом регистра, то есть «хлеб» и «ХЛЕБ» – для поисковой системы разные слова. При анализе текста необходимо держать в памяти, что поиск может вестись по словосочетаниям и целым фразам, соответственно, их тоже имеет смысл вычленить и разместить здесь же. Некоторые поисковые системы игнорируют большой объем ключевых слов в области метатегов, но при ранжировании результатов поиска делают упор на слова, выделенные более крупным шрифтом, с полужирным выделением, оформленные как заголовки и подзаголовки. Следует также помнить, что ключевые слова в первых пяти-шести абзацах веб-страницы также положительно влияют на ее рейтинг, поэтому имеет смысл избегать в начале таблиц и графических объектов, не описанных ключевыми словами.

Теперь поговорим о графике. Поисковые машины работают только с текстовыми данными, поэтому если вы нарисуете замечательную картину, на которой «аршинными» буквами изобразите ключевые слова, характеризующие содержимое страницы, то поисковая машина все равно ее проигнорирует. Как этого избежать? Используйте все возможности тега ``, предназначенного для вставки графических объектов в текст веб-страницы, явно задавая его атрибуты: `alt` – отвечающий за размещение альтернативного текста и `title` – выводящий всплывающую подсказку. Рассмотрим это на примере:

```
<html>
  <head>
    <title>Заголовок страницы</title>
  </head>
  <body>
    
  </body>
</html>
```

Высокий показатель цитирования страницы определяется большим количеством ссылок, ведущих на нее, поэтому старайтесь максимально использовать особенности гипертекстовой разметки при создании перекрестных ссылок даже в пределах своего сайта. Если же есть возможность организовать ссылки на свои страницы и с других сайтов, обязательно этим воспользуйтесь.

Таким образом, используя все возможности языка HTML, можно существенно повысить количество ключевых слов для поиска необходимой посетителю информации именно на вашей странице. Но не следует злоупотреблять ключевыми словами как в области метатегов, так и в самой веб-странице. В недалеком прошлом недобросовестными сайтостроителями использовался один из алгоритмов поиска — соотношение кода страницы к размещенному на ней тексту, когда они помещали на своих страницах текст с минимальным размером символов либо с цветом букв, сливающихся с фоном. Это делало текст нечитаемым для пользователей и сбивало с толку поисковых пауков. На данный момент проблема решена и поисковые машины на этот трюк не ведутся, а просто снижают рейтинг страницы, если тематика ключевых слов не совпадает с тематикой страницы.

Еще одна проблема, решаемая на стороне клиента и отчасти хостера, — это сложность генерируемых URL-адресов для динамических сайтов, созданных с помощью различных CMS. Дело в том, что при генерации URL страницы сайта CMS, в частности Joomla!, генерирует не тот адрес, на котором фактически находится страница, а сообщение системе управления о том, какой контент необходимо получить из базы данных для формирования страницы. Ни для пользователя, ни для поискового паука это не проблема, но рейтинг сайта со статическим URL всегда будет выше, чем сайта с динамическим URL. Решение простое и встроено в саму систему Joomla! — нужно выполнить команду **Общие настройки** ▶ **Установки Search Engine Optimization** ▶ **Включить SEO**. Здесь может возникнуть только одна проблема: не все узлы, особенно бесплатного хостинга, поддерживают этот параметр. Таким образом, вы рискуете вместо генерируемого сайта получить ошибку 404 (файл не найден).

Для облегчения индексации всех страниц сайта поисковым роботом имеет смысл использовать так называемую карту сайта. Она генерируется специальной программой, которую можно скачать с сайта поисковой системы вместе с инструкцией по использованию либо с сайта поддержки вашей CMS.

Итак, ресурс создан, максимально оптимизирован, соответствует всем предъявляемым требованиям и даже размещен на хостинге с удобно воспроизводимым сетевым адресом. Пора приглашать гостей.

Чтобы попасть на тот или иной сайт в Сети, необходимо знать его адрес. Как много людей знают адрес вашего сайта? Что необходимо сделать для того, чтобы как можно больше людей посетили ваше детище?

Решение первое — рассыпать адрес созданного сайта знакомым и незнакомым людям с просьбой или требованием его посетить. Энергозатраты огромны, а эффект минимален. Кроме того, можно попасть под штрафные санкции за рассылку спама.

Решение второе — обменяться баннерами и гиперссылками с сайтами, сходными по тематике, обеспечив себе перекрестный переход целевого посетителя, ориентированного на информацию именно на вашем сайте. Но и здесь существует маленькая проблема — неизвестный и непосещаемый сайт никому не нужен, и если свой баннер вам разрешат разместить на странице, то за размещение вашего на раскрученном сайте, скорее всего, придется платить. Как итог — медленное формирование целевой аудитории, но рано или поздно какой-либо поисковый паук забредет на ваш сайт по перекрестной ссылке и проиндексирует его содержимое, внеся в базу данных поисковой машины. Хорошо использовать это решение, когда вы твердо уверены в незыблемой информационной ценности представленного материала и не торопитесь расширить аудиторию своего сайта.

Решение третье — самому зарегистрироваться в поисковых системах и самостоятельно начать активные наступательные действия. Об этом мы поговорим чуть ниже. Поскольку реклама — двигатель торговли, то вам и придется выступать в роли этого двигателя, продвигая информацию о своем сайте самостоятельно. Начните участвовать в максимальном количестве форумов и конференций, близких по тематике к вашему сайту. В своих никах и публичной информации не стесняйтесь в качестве подписи оставлять адрес своего сайта, можно даже добавлять приписку, что, дескать, дополнительная информация и консультации вот на этом сайте. Начните писать статьи по сходной тематике и публикуйте их в открытых информационных ресурсах, не забывая делать ссылки на материалы своего сайта. Ну и самое главное — всегда старайтесь грамотно и быстро отвечать посетителям вашего ресурса. Люди любят внимание, особенно в наше сетевое время, поэтому оперативные и качественные ответы на вопросы принесут вам не только почет и уважение, но и огромную целевую аудиторию, ради признания которой вы и создаете свой сетевой шедевр, стараясь открыть миру то, что доступно только вам.

10.2. Поисковые системы и регистрация в них

Использование поисковых систем (машин) — наиболее эффективный способ донести информацию о существовании вашего сайта до целевой аудитории. В современном мире поисковые системы обрабатывают несколько миллионов запросов в сутки, при этом в пиковые дни — перед праздниками, каникулами или студенческими сессиями — можно наблюдать рост целевых запросов, посвященных определенным темам. Если вы подготовили сайт к приему посетителей, то пора обратиться к этим помощникам,

предложив свой ресурс в качестве дополнения к их уже существующей базе данных мировых сетевых информационных ресурсов.

Поисковые системы состоят из двух частей: программы поиска информации — так называемого поискового робота, или поискового паука, и базы данных — индекса, где хранится информация, собранная пауком за время его скитания по просторам Сети. Алгоритмы работы поисковых пауков и построение рейтингов представленных ссылок на информационные ресурсы основаны на индексировании базы данных и являются коммерческой тайной поисковой системы, за которой постоянно охотятся веб-мастера и специалисты по продвижению сайтов. Однако основной принцип известен и широко освещается всеми поисковыми системами. Поисковый паук путешествует по странице вашего сайта и определяет соответствие ее содержания поисковому запросу — так называемую релевантность запроса. Главенствующий критерий релевантности документа — частота употребления в тексте ключевых слов, соответствующих запросу. Поэтому выше мы и обратили внимание на то, как повысить релевантность страницы с помощью ключевых слов. Ваша административная задача сводится к тому, чтобы, проанализировав содержимое страницы, выделить эти ключевые слова и словосочетания, правильно их оформить в коде страницы.

Страниц в мире уже не миллионы, а миллиарды, соответственно, скрестные возможности паука не соответствуют запросам времени, поэтому поисковые пауки записывают всю найденную информацию в базу данных — индекс. Таким образом, выполняя информационный поиск в Сети, мы работаем не с реальной сетью, а с ее фотографией, сделанной какое-то время назад. Поэтому не стоит удивляться тому, что зарегистрированный вчера сайт до сих пор не выходит по запросу в поисковой системе. Дело в том, что поисковый робот еще не добрался до вашей страницы и не проиндексировал ее. Подождите. Сколько? Неизвестно, но он вас рано или поздно обязательно найдет, поскольку работает 24 часа в сутки, семь дней в неделю, без выходных и праздников. А пока начните самостоятельную операцию по активному продвижению своего сайта.

Возникает следующий закономерный вопрос: для получения максимальной аудитории необходимо зарегистрироваться во всех поисковых системах? Зачем? Вспомните, какими поисковыми системами вы пользуетесь в повседневной жизни, наверняка их немного. А в мире их уже несколько сотен. Так ли заинтересованы вы в демонстрации своего ресурса и сможете ли передать информацию пользователю, который говорит на другом языке? Ответив, определите для себя необходимость регистрации своего сайта в поисковой системе, например в Боливии, а как определили, то приступайте к регистрации. Процесс регистрации сайтов в поисковых системах достаточно прост и описан в размещенных на них инструкциях, обычно требуется ввести информацию о сетевом адресе сайта, а остальное система сделает сама.

Рассмотрим на примере регистрацию сайта в некоторых широко известных поисковых системах.

Регистрация сайта в поисковой системе yandex.ru сводится к заполнению формы **Сообщить о новом сайте**, расположенной по адресу <http://webmaster.yandex.ru/addurl.xml> (рис. 10.1).

Яндекс.Вебмастер

webmaster.yandex.ru/addurl.xml

Яндекс вебмастер

Мои сайты Добавить сайт

Сообщить о новом сайте

Мы регулярно добавляем новые сайты в нашу базу, находя их по ссылкам на веб-страницах. Вы можете сообщить нам о появлении нового сайта с помощью данной формы. Достаточно добавить URL главной страницы сайта, а остальные страницы будут найдены автоматически.

Стандартный протокол `http` указывать не обязательно. Если же доступ к сайту ограничен и осуществляется по `https`, протокол нужно указать явно. Например: <https://mail.yandex.ru>

www.saturnkom.narod.ru Добавить

519516

Яндекс

Если у вас возникли вопросы по добавлению и индексированию ресурсов в поисковой машине Яндекс, прочтите, пожалуйста, [ответы на частые вопросы](#)

Яндекс Директ – реклама вашего сайта на поиске Яндекса

Рекламная сеть Яндекса – зарабатывайте на своем сайте

Информация о ваших сайтах в Яндекс Баре

Яндекс Визажисты – ссылка на ваш сайт с главной страницы Яндекса

Сообщить о новом сайте

Проверить URL

Анализ файла.txt

Проверка орфографии

Пожаловаться на спам

Удалить URL

Валидатор

Материалы по теме

Рис. 10.1. Форма регистрации сайта в поисковой системе yandex.ru

Введем в поле **URL** доменное имя сайта и подтвердим, что не являемся спам-роботом, введя контрольные цифры с экрана. После нажатия кнопки **Добавить** поисковая система проверит наличие страницы `index.html` или `index.php` по указанному адресу и предложит вам либо создать в корневом каталоге сайта простой текстовый файл с определенным именем, либо разместить в области метатега комбинацию символов. Это делается для подтверждения ваших прав на сайт и в качестве защиты от автоматической регистрации несуществующих сайтов. После выполнения этих шагов происходит повторная проверка выполнения вами необходимых действий, и система выдает сообщение следующего характера (рис. 10.2).

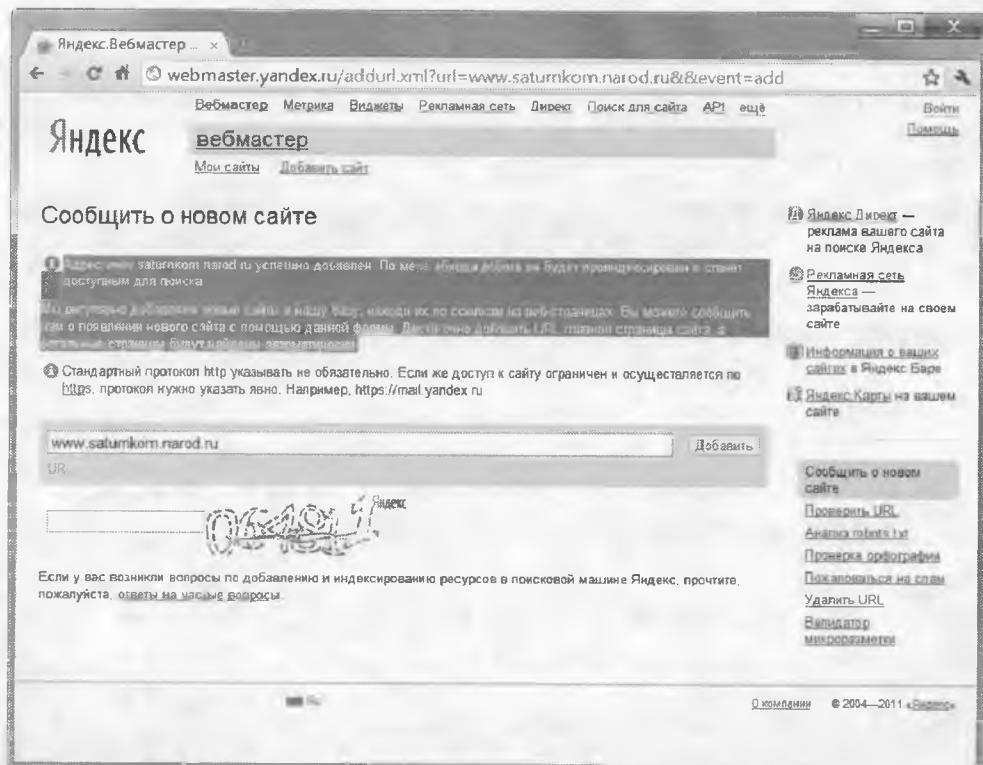


Рис. 10.2. Сообщение поисковой системы yandex.ru об успешной регистрации сайта в базе данных поисковой системы

Аналогично происходит регистрация в поисковой системе Google. Открываем страницу, находящуюся по адресу <http://www.google.ru/intl/ru/addurl.html>, вводим URL своего сайта и необходимый комментарий, который не влияет на индексацию ресурса, а затем подтверждаем все выполняемые действия нажатием кнопки Добавить URL (рис. 10.3).

Как видите, регистрация в поисковых машинах не представляет особой сложности. Следует лишь помнить, что при регистрации на не русскоязычных поисковых ресурсах необходимо продублировать содержимое метатегов на язык используемой поисковой системы и на английский язык.

Как и любая операция, регистрация в поисковых системах требует времени. На регистрацию в одном поисковике надо потратить пять минут, в другом — еще пять, а можно ли автоматизировать процесс? Конечно, для этих целей в Сети можно найти и использовать множество программ, в основном платных и условно бесплатных, предназначенных для автоматической регистрации в поисковых системах.

Но любая поисковая система различными способами борется с автоматически регистрируемыми сайтами. Это может быть и подтверждение реги-

страции вводом текста с графического объекта, и необходимость создания учетной записи, и подтверждение наличия сайта специальной записью в макетке страницы `index.html`, и помещение специально сгенерированного при регистрации файла в каталог с сайтом, и еще много различных способов, призванных отличить вдумчивого человека от программы. Поэтому скажем так: возможности существуют, а будете вы их использовать или нет — решать вам. В любом случае объявления со страниц в Сети предлагают решить проблемы бесплатно или за символическую плату, но при этом помните про бесплатный сыр.



Рис. 10.3. Форма регистрации в поисковой системе Google.ru

10.3. Каталоги и регистрация в них

Поиск информации в Сети может осуществляться не только с помощью поисковых машин, но и в каталогах. Служба каталогов представляет собой структурированную коллекцию ссылок на различные тематические ресурсы. В отличие от поисковых машин, где проверка актуальности и построение рейтинга представляемых ссылок проводятся программным путем, за каталогом следят живые люди, поэтому сортированная вручную информация

более релевантна и тематически сгруппирована. Регистрация в каталогах не так проста, как в поисковых системах, и при условии регистрации коммерческого сайта или ресурса, содержащего коммерческую рекламу, в большинстве раскрученных каталогов будет платной. Подробности регистрации в каталогах и ее условия приведены в инструкциях, расположенных непосредственно на самих страницах каталогов, например для каталога Mail.ru адрес страницы регистрации имеет следующий вид: <http://list.mail.ru/fast-bin/preregistration.bat>.

10.4. Рейтинги

Если вы уверены в востребованности своего информационного ресурса и желаете не только вкладывать в него силы и средства, но и получать отдачу, в частности материальную, то вам не обойтись без регистрации в рейтингах.

В отличие от поисковых машин и каталогов, рейтинги формируются по принципу «Лучшие сайты наверху, не совсем лучшие где-то там...», причем критерии формирования рейтинга четко прописаны в инструкции к нему, например расчет рейтинга популярности на «Рамблер ТОП100» размещен по адресу <http://top100.rambler.ru/news/faq>.

Как рассчитывается индекс популярности на «Рамблер ТОП100»? На данный момент он учитывает следующие аудиторные характеристики:

- количество хостов на главной странице за день;
- число хостов на всем сайте за день;
- среднее суточное количество просмотров на хост за предыдущие 7 дней;
- количество хостов на главной странице за предыдущие 7 дней;
- число хостов на всем сайте за предыдущие 7 дней;
- среднее суточное количество просмотров на хост за предыдущие 7 дней;
- среднее количество просмотров только одной страницы за предыдущие 7 дней;
- среднее количество просмотров от 3 до 5 страниц за предыдущие 7 дней;
- среднее число посетителей из России за предыдущие 7 дней.

Под хостом «Рамблер» понимает уникальный IP-адрес, с которого производился запрос веб-страницы.

- Индекс учитывает глобальную динамику аудитории.
- Индекс пересчитывается один раз в час.

Как видите, рейтинг зависит только от посещаемости страниц сайта, такое же требование к вашему информационному ресурсу предъявят и потенциальные рекламодатели, в первую очередь задав вам вопрос: «Насколько посещаем ваш ресурс?»

Мой знакомый занимается туристическим бизнесом в Карелии и на Ладожском озере и на вопрос об эффективности рекламы услуг посредством сайта достаточно четко ответил: «Шесть процентов с каждой последующей цифры». Иными словами, от всех посетителей сайта 6 % пишут письмо и задают вопросы и 6 % из задававших вопросы покупают путевки на отдых, но это сезонный специфический бизнес и в не очень популярном месте.

Таким образом, чем больше количество посетителей вашего сайта, тем больше вероятность того, что они клюнут на рекламные баннеры, представленные на ваших веб-страницах. Соответственно, у рекламодателя будет больше вероятных покупателей и дороже будет стоить рекламное место на вашем сайте. Поскольку никто не любит, чтобы его обманывали, то в качестве третейских судей выбирают различные рейтинги и упомянутая в деловом разговоре фраза: «Триста посетителей в сутки по топ Рамблеру...» скажет очень много потенциальному деловому партнеру.

При регистрации вы получаете уникальный счетчик, оформленный в виде баннера. Вы обязаны разместить его на тех страницах, участие которых в рейтинге вы предполагаете. А далее вы начинаете всеми силами и честными способами привлекать посетителей на ваш сайт. Слово *честными* выделено неслучайно — при оформлении заявки на участие в рейтинге прописано, что вы можете лишиться рейтинга даже при подозрении в нечестном продвижении, то есть никто даже не будет вам доказывать, что вы неправы — «Джентльмен джентльмену верит на слово».

Регистрация в рейтингах бесплатная и сложности не вызывает, лишь требует четкого заполнения прилагаемой анкеты. Отнеситесь к этому предельно серьезно, так как именно по результатам обработки анкеты ваш информационный ресурс будет отнесен к той или иной группе. В противном случае при затруднении классификации ресурса высокого показателя в рейтинге вам не видать.

10.5. Баннерная реклама

Используя по максимуму предоставляемые сетью возможности для продвижения и раскрутки своего сайта, не стоит забывать и о небольших мелочах, также способных привлечь к вашей информации дополнительных посетителей. Поговорим о баннерной рекламе. По оценкам специалистов, при правильном использовании в продвижении сайта она привлекает до 3 % его посетителей.

Баннер (кнопка) — графический объект прямоугольной формы допустимых форматов GIF или JPEG, связанный гиперссылкой с другим сайтом в Сети. Соответственно, баннерные сети — это сети, использующие баннеры для перехода на целевой сайт.

Существует два метода использования рекламы сайтов с помощью баннеров: метод обмена баннерами и метод покупки рекламного места под баннеры.

В первом случае обмен баннерами производится на безвозмездной, но пропорциональной основе, в основном базирующейся на рейтинге вашего сайта. Иначе говоря, высчитывается средняя стоимость щелчка на баннере и в случае неравноценности стоимости предлагается пропорциональное размещение баннеров на их страницах. Таким образом, за размещение одного своего баннера на одной странице более раскрученного сайта вы можете заплатить размещением его баннеров на нескольких страницах своего сайта либо, при условии, что вы более успешны, можете произвести обмен в свою пользу.

При покупке мест под ваш баннер в рекламной компании информационного ресурса лучше всего использовать фирмы, специализирующиеся на организации баннерных сетей, например обменную службу «ИнтерРеклама», представленную по адресу <http://ir.ru>, либо любую другую, которую сможете найти в Интернете. Привлекая специалистов к рекламе своего сайта, вы заключаете и оплачиваете договор на предоставление количества переходов по вашим баннерам к вам на сайт. Таким образом, вкладывая реальные деньги в баннерную рекламу, вы получаете переход реального пользователя на ваш сайт. Как его там удержать — это уже вопрос контента и интерактивности вашего сайта.

10.6. Влияние интерактивности сайта на привлечение посетителей

Интерактивность сайта — не только очень хороший способ привлечь новых посетителей и расширить контент сайта, способствуя росту его релевантности, но и великолепная возможность заставить пользователей неоднократно приходить на ваш сайт, постепенно превратив их в своих друзей. При этом подразумеваются способности гипертекста по организации переходов внутри сайта и на его страницах, а также возможность для пользователя высказаться и/или получить компетентный ответ на свой вопрос. Для этого используйте различные формы обмена информацией с помощью возвратных форм, располагаемых на веб-страницах, особенно это касается коммерческих сайтов. Подключите гостевую книгу или обязательно модерируемый форум. Если есть такая необходимость, организуйте работу чата или активного консультанта, постоянно присутствующего в Сети. Ваш сайт должен жить интенсивной жизнью, и она должна быть видна на его страницах.

Если модератор форума в течение непродолжительного промежутка времени выполняет определенные действия: дает консультации, вносит замечания и пр., а служба поддержки своевременно отвечает на вопросы и растолковывает ответы для самых непонятливых, телефон консультанта доступен и можно получить ответ на свой вопрос, то у пользователя форми-

руется иллюзия дружелюбности всей команды, работающей над проектом. Он решит, что ему здесь не откажут, и при прочих равных условиях снова придет к вам.

Сформировав определенный круг постоянных посетителей вашего сайта, реализуйте регулярную тематическую рассылку информационных материалов только *зарегистрированным* пользователям, с четко прописанной явной возможностью отказаться от подписки. Не перегружайте контентом почтовые сообщения в рассылке, ведь это просто рекламный релиз, и его задача — не давать информацию, а привлечь посетителя на страницы вашего сайта, где предоставлят все интересующие его материалы. Помните, что информационный поток в настоящее время настолько велик, что мы обычно по первым двум строкам определяем, будем мы читать этот случайный текст или нет.

Таким образом, используя вышеописанные методы привлечения посетителей, а также свои собственные и позаимствованные из других источников методы, вы гарантируете своему информационному ресурсу долгую и успешную жизнь, а себе — небольшой кусочек хлеба с маслом. Наиболее же усидчивые и удачливые могут получить и бутерброд с красной икрой.

10.7. Это волшебное слово SEO

SEO — загадочное слово, которым манипулируют различные агентства, рекламирующие свои возможности в области продвижения сайта в рейтингах поисковых машин. SEO — и все ваши проблемы решены. Используя SEO, мы выведем ваш сайт в первую десятку ссылок поисковой системы, мы являемся специалистами по SEO...

Как часто мы видим это слово и подобные сообщения на страницах сайтов компаний, занимающихся веб-разработкой. И сразу начинает мучать вопрос: «А может я не прав? И следует действительно обратиться к специалистам, заплатить деньги, они же лучше меня в этом разбираются?» Что на это можно сказать? Во-первых, не боги горшки обжигают, а во-вторых, нет ничего сложного в этом страшном слове SEO. Ведь подготовить содержимое создаваемого сайта, чтобы он был наиболее дружественен к роботам поисковых машин, это значит подготовить сайт для целевого потока его посетителей. Структурировать и оформить содержание сайта таким образом, чтобы размещенная на нем информация представляла интерес для широкого круга посетителей сайта и была доступна для них, а значит и для поискового робота, осуществляющего ее индексацию. Это как раз то, ради чего мы с вами и создаем сайт. Ведь поисковые роботы, по идее создателей поисковых систем, являются идеальными библиотекарями, или библиографами, которые стараются максимально точно удовлетворить наш запрос на поиск той или иной информации, определяемой ключевыми терминами в строке поиска поисковой системы.

Создатели поисковых систем заранее оговаривают правила игры, на основании которых присваивают ранг определенной веб-странице, содержащей необходимую, согласно сделанному запросу, информацию. Эти правила в виде рекомендаций для веб-мастеров размещены на всех сайтах поисковых систем. Перед тем как приступать к наполнению создаваемого сайта, а в идеале еще на начальном этапе его планирования, следует не только ознакомиться с ними, но и руководствоваться ими в процессе создания структуры сайта, чтобы впоследствии вносить как можно меньше изменений.

Не будем откладывать дела в долгий ящик, а просто перейдем на веб-страницу поисковой системы «Яндекс» — фаворита русскоязычного поиска в Сети. На ней как раз отражены эти самые правила: <http://help.yandex.ru/webmaster/recomend.pdf>. Приводить весь текст рекомендаций на страницах книги нет необходимости, ведь они находятся в свободном доступе. Поэтому на примере выводов, сделанных в конце каждого раздела инструкции, посмотрим, какие изменения нам следует внести в содержимое нашего сайта, чтобы он стал более дружественен для поисковых систем.

Определимся, на какие вопросы отвечает ваш сайт

«...На положение страницы в результатах поиска по запросу очень сильно влияет то, насколько ее текст соответствует формулировке запроса — „текстовое соответствие“». Например, если пользователь ищет какой-либо термин, а на страницах вашего сайта он вовсе не употребляется, то маловероятно, что поисковая система сочтет ваш ресурс содержащим хороший ответ на запрос пользователя...» «...Проанализировать, что и как именно спрашивают пользователи у „Яндекса“, вам поможет сервис „Статистика ключевых слов“».

Вот к этой статистике ключевых слов мы и обратимся, адрес страницы: <http://wordstat.yandex.ru/> (рис. 10.4 и 10.5).

Проанализируем информацию, выданную сервисом «Статистика ключевых слов» по двум запросам: «туризм в космос» — 66 обращений, «космический туризм» — 2299 обращений в месяц по данному словосочетанию. Как видно, одна и та же смысловая информация, но представленная в различных словоформах, вызывает абсолютно разную реакцию у посетителей сайта. Гости могут искать информацию, которая является эксклюзивной и находится только на вашем сайте в этом уголке галактики, но из-за несответствия формы запроса по ключевым словам тексту, размещенному на веб-странице, они могут ее вообще не найти, поскольку поисковая машина понимает все буквально. В чем причина?

Более полный и аргументированный ответ на этот вопрос, пожалуй, смогут дать только психологи или специалисты по лингвистическому анализу, а я обычно объясняю все гораздо проще. Поставьте себя на место пользователя и напишите первое искомое сочетание слов по заданной теме. Средне-статистический пользователь, скорее всего, напишет «космический туризм»

вместо более официального «туризм в космос». Делаем следующий вывод: следите за соответствием текстов на страницах вашего сайта запросам ваших потенциальных клиентов. И, как следствие, применяем рекомендации, которые предлагают создатели поисковой системы: «...Выберите несколько ключевых фраз, которыми можно охарактеризовать предоставляемые вами услуги. Чтобы страница ранжировалась высоко, она должна давать лучший в Интернете ответ на вопрос пользователя...».

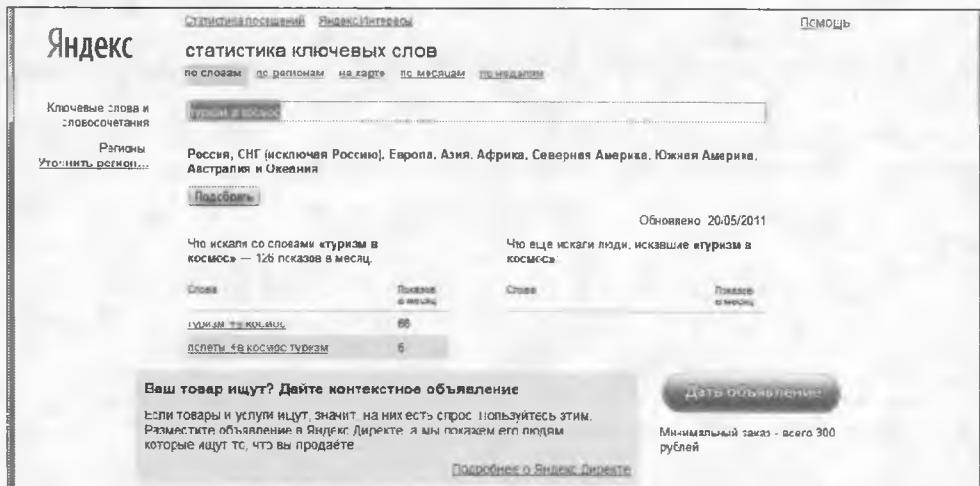


Рис. 10.4. Окно Статистика ключевых слов с запросом «туризм в космос»



Рис. 10.5. Окно Статистика ключевых слов с запросом «космический туризм»

Представление информации на сайте

«...При ранжировании учитывается не только формальное соответствие тексту запроса, полнота представленной информации и ее актуальность, но и то, как она структурирована и оформлена... „Яндекс“ анализирует, насколько представленная на сайте информация удобна для восприятия пользователем...», — говорит нам «Яндекс».

Таким образом, на первое место после побора ключевых слов, определяющих смысловую нагрузку страницы, выносится ее структурное оформление, а также стилевое оформление контента. При этом интеллект поискового робота отсеивает так называемую агрессивную рекламу и не работает с содержимым веб-страницы, требующим дополнительных действий, кроме прямого просмотра. Из этого вытекают следующие рекомендации.

- «Основная суть документа должна быть понятна уже на первом экране браузера.
- Не рекомендуется использовать элементы оформления, делающие неочевидным доступ к тексту для пользователей: окна прокрутки, скрытые выпадающие блоки и т. п.
- Если текст большой, разбивать его на несколько веб-страниц следует по крупным логическим единицам (например, по главам, но не по абзацам).
- Текст должен быть грамотно написан и аккуратно оформлен. Для проверки орфографии на сайте может пригодиться специальный инструмент, доступный в „Яндекс.Вебмастере“.

Как видно из предлагаемых рекомендаций, чтобы структурно правильно и грамматически верно оформить и разбить текст по страницам сайта, не требуется какого-либо специального образования, а достаточно знаний в пределах средней школы, если, конечно, ученик изучал, а не проходил материал на уроках русского языка. Структурную разметку веб-страниц в этой книге мы уже рассмотрели, знаем, что такое теги, и представляем, как использовать заголовки, списки и пр.

Индексирование сайта

Для возможности индексации сайта поисковой системой его необходимо внести в базу данных этой системы, что мы и рассмотрели в разделе 10.2.

Далее можно использовать дополнительные сервисы самой поисковой системы, позволяющие явно указать поисковому роботу последовательность обхода вашего сайта и акцентировать его внимание на тех или иных веб-страницах.

Прежде всего это карта созданного сайта. В создании карты сайта поисковая система оказывает вам помощь и гарантирует работу рекомендуемого устройства, если вы разместите в каталоге с сайтом специальный файл, содержащий эту карту.

Но предоставим слово самому «Яндексу»: «...Файл Sitemap — это файл, расположенный на сайте, с дополнительной информацией о страницах сайта, подлежащих индексированию. С помощью файла Sitemap вы можете сообщить „Яндексу“, какие страницы вашего сайта нужно индексировать, как часто обновляется информация на страницах, а также индексирование каких страниц наиболее важно».

Только вам решать, использовать данную услугу системы или нет, но при наличии на сайте страниц с динамически изменяемыми URL созданная карта сайта — незаменимый поводырь для поискового робота.

Есть еще специально создаваемый файл `robots.txt`: «...Robots.txt — файл, предназначенный для роботов поисковых систем. В этом файле веб-мастер может указать параметры индексирования своего сайта как для всех роботов сразу, так и для каждой поисковой системы по отдельности». Иными словами, разместив данный файл в каталоге с сайтом и прописав согласно инструктирующим директивам поведение заглянувшего к вам на веб-страницу поискового робота, вы почти в буквальном смысле подкладываете ему самые лакомые куски содержимого, оставляя за бортом тот контент, который не будет интересен пользователю и самому поисковому роботу.

Существуют также определенные технические проблемы при индексировании отдельных типов данных, размещенных на веб-странице, которые тоже следует учитывать при создании структуры сайта и при организации ссылок на конкретные документы:

«...Яндекс индексирует основные типы документов, распространенные в Сети. Но существуют ограничения, от которых зависит, как будет проиндексирован документ и будет ли проиндексирован вообще.

- Большое количество cgi-параметров в URL, большое количество повторяющихся вложенных директорий и слишком большая общая длина URL может привести к ухудшению индексирования документов.
- Для индексирования важен размер документа — документы больше 10 Мбайт не индексируются.
- Индексирование flash:
 - индексируется, если он не встроен в HTML и страница передается с HTTP-заголовком, содержащим Content-Type: application/x-shockwave-flash;
 - индексируются файлы *.swf, если на них есть прямая ссылка.
- В документах PDF индексируется только текстовое содержимое. Текст, представленный в виде картинок, не индексируется.
- „Яндекс“ корректно индексирует документы в формате Open Office XML и OpenDocument (в частности, документы Microsoft Office и Open Office). Но следует учитывать, что внедрение поддержки новых форматов может занимать некоторое время...».

Таким образом, если правильно сформировать содержимое, казалось бы, совсем не HTML-кода документа и грамотно проставить ссылки на различные объекты, то вполне законным способом можно поднять релевантность своей страницы в глазах поискового робота.

Структура сайта

Про структуру создаваемого сайта можно говорить долго, и смысл разговора сводится только к одной фразе: структура сайта должна быть грамотно продуманной и четко организованной. Сам «Яндекс» придерживается такого же мнения в отношении структуры сайтов, что и подтверждается тезисами следующей статьи. В ней описываются требования к структурному оформлению веб-страниц, позволяющему поисковому пауку свободно разгуливать по просторам индексируемого сайта.

- «Поддерживайте четкую ссылочную структуру на сайте. Каждый документ должен относиться к своему разделу.
- Используйте карту сайта.
- Ограничивайте индексирование служебной информации. Многочисленные страницы-дубликаты, результаты поиска по сайту, статистика посещений и тому подобные страницы могут тратить ресурсы робота и мешать индексированию основного содержимого сайта.
- Каждая страница должна иметь уникальный адрес (URL). Желательно, чтобы вид URL давал представление о том, что содержится на соответствующей странице.
- Делайте ссылки на другие разделы сайта текстовыми, чтобы предоставить роботу больше информации о материалах, которые в них размещены.
- Проверяйте корректность symlink, чтобы при движении по сайту у страниц не получался бесконечно растущий URL».

Особо хочется отметить к выполнению пункты 4 и 5 данных тезисов: именно этим местам в создаваемом сайте начинающие веб-мастера, к сожалению, не уделяют должного внимания, не продумывают и не прописывают значений соответствующих тегов. Как видно из требований поисковой системы, это один из краеугольных камней, влияющих на повышение рейтинга веб-страницы при ее индексации.

Использование графики

Поскольку ни одна поисковая система на момент написания книги (и, думаю, еще после этого долгое время) не имеет механизмов распознавания графических образов, то все представляемые на странице графические объекты должны описываться уже рассмотренными ранее атрибутами соответствующих тегов. Вот, что по этому поводу говорит «Яндекс»:

«...Во-первых, заполняйте атрибут ALT тега . Все изображения на странице могут иметь альтернативный текст, который задается в атрибуте ALT тега .

Во-вторых, дублируйте текст, представленный картинкой... Если использование графического элемента с текстом необходимо, продублируйте то, что написано на картинке, текстом. Тогда никакая нужная для корректного ранжирования информация не будет упущена.

Нюансы, возникающие при использовании графики.

- Графическое меню. Меню сайта, выполненное в виде изображений, а не текстов, хоть и дает возможность перейти по ссылке на внутренние страницы, но не дает «Яндексу» представления о том, какая именно информация на них размещена. По возможности следует предоставить альтернативное текстовое меню со ссылками на основные разделы сайта.
- Если дизайн сайта выполнен с применением технологии flash, сделайте и текстовую версию сайта.
- Если картинки используются как иллюстрация к тексту, стремитесь размещать их как можно ближе к фрагменту текста, который они иллюстрируют. Если такой возможности нет, сопроводите их подписями непосредственно рядом с картинкой.
- Имена файлов картинок желательно делать осмысленными (то есть не img_123456.jpg, а ktulhu.jpg) и соответствующими содержанию картинки.
- Если в тексте дается ссылка на изображение, то ее текст также должен быть осмысленным (то есть не рис. 1, а Ктулху в естественной среде обитания).
- Закрывайте от индексирования в robots.txt изображения CAPTCHA и другие автоматически генерированные картинки (баннеры, графики, счетчики, графические информеры и т. п.)».

Распространенные ошибки

Поисковая система «Яндекс» особенно пытается предотвратить ошибки, совершаемые веб-мастером. Причем вне зависимости от того, сделаны эти ошибки по незнанию или умышленно, результат всегда один — исключение ресурса из базы данных индексации и занесение доменного имени в черный список. А это уже бьет по карману владельца сайта, как вы понимаете. Поэтому постарайтесь при проектировании и создании своего шедевра избегать следующих приемов:

- «...Навигация через сценарии. Например, (для организации меню) можно использовать технологии JavaScript или Flash. Робот „Яндекса“ не переходит по таким ссылкам, поэтому следует дублировать ссылки, реализованные с помощью сценариев, обычными текстовыми ссылками.

- **Использование фреймов — тегов <FRAMESET>, <FRAME>, <IFRAME>.** Для корректного ранжирования документа не рекомендуется использовать фреймы, так как поисковый робот „Яндекса“ не индексирует документы, подгружаемые в них.
- **Избыточное автоматическое перенаправление (редиректы).** Редирект может быть полезен только в том случае, если адреса страниц меняются по техническим причинам и необходимо перенаправить пользователя на новый адрес страницы.
- **Адреса страниц.** Каждая страница должна быть доступна по единственному и постоянному адресу.
- **Клоакинг.** Избегайте ситуаций, когда поисковый робот индексирует одно содержание страницы, а пользователь при обращении к этой странице получает другое.
- **Изображения вместо текста.** Избегайте создания страниц, не содержащих текст. Если главная страница сайта выполнена в виде изображения, являющегося ссылкой на основную часть сайта, и сама не содержит текста, это может помешать ранжированию сайта.
- **Soft 404.** Одна из распространенных ошибок заключается в замене сообщения об ошибке 404 (страница не найдена) для несуществующих страниц на страницу-заглушку, которая возвращается с кодом ответа 200 (ОК)... Это приводит к более медленному индексированию полезных страниц на сайте.
- **Движок сайта.** Следите за корректностью работы программного обеспечения сайта — ошибки в сценариях сайта могут привести к тому, что одни и те же страницы при переходе на них из разных разделов будут иметь разные адреса....»

Ссылки

Сама основа размещения связанный информации на удаленных компьютерах — это, конечно же, организация ссылок. От них зависит не только структура сайта, но и движение по нему поискового робота при индексации им просматриваемых страниц:

«...Партнерские ссылки. Если на других сайтах стоят ссылки на ваш сайт, проследите за тем, чтобы их текст содержал адекватное описание вашего сайта, а не просто URL.

Ссылки, которые используют интернет-СМИ. Если такие СМИ используют информацию с вашего сайта, будет полезно иметь в тексте ссылку с подходящей ключевой фразой.

Каталоги ссылок (в которых вы производите регистрацию своего сайта). Вот несколько признаков, которыми обладает хороший каталог (на которые следует обращать внимание при поиске каталога для регистрации в нем).

- Каталог должен быть модерируемым.
- Каталог должен иметь удобную для пользователей навигацию.

- В тематических разделах каталога должны присутствовать ссылки на авторитетные в соответствующей области сайты, сайты компаний-лидеров в своей отрасли.
- Названия и описания сайтов не должны содержать грамматических ошибок, должны быть удобными для восприятия и использования.
- На страницах каталога не должно быть рекламы, мешающей восприятию основного содержимого.

Уделите внимание ссылкам, которые пользователи могут размещать на вашем сайте. Если у вас есть разделы, в которых можно оставлять свои сообщения (например, комментарии к записям, форум, обсуждение материалов), внимательно следите за исходящими ссылками. Все ссылки по возможности должны модерироваться владельцем сайта».

Платные ссылки, ссылочные биржи. «Яндекс» ценит ссылки, которые ставятся для того, чтобы помочь посетителям сайта. Ссылки, которые появляются на сайтах за определенную плату (особенно когда они размещаются с использованием ссылочных бирж и предназначены для поисковых систем, а не посетителей сайта), мешают «Яндексу» корректно определять степень доверия к сайтам и их релевантность запросам пользователей. «Яндекс» даже сделал такое заявление: «В настоящий момент мы приняли меры для уменьшения влияния SEO-ссылок с наиболее популярных сайтов Рунета. Мы планируем и далее выявлять подобные ссылки, а также принимать все меры к тому, чтобы:

- сайты, продающие ссылки, потеряли способность влиять таким способом на ранжирование;
- покупка ссылок с сайтов не могла бы привести к повышению ранга сайта-покупателя».

Как видно из тезисного обзора данной статьи, организация ссылок — дело архиважное. Тут главное не помочь, а не навредить.

Метаописания

Некоторые начинающие веб-мастера недооценивают возможности правильно сформированной области метатегов, говоря, что современные поисковые системы игнорируют этот тег. На что мы им прямо ответим, что они полностью не правы, и отошлем их к следующим словам «Яндекса»:

«...Одним из способов влияния на представление своего сайта в результатах поиска (его сниппет) является использование метаописаний — тега *meta description*. При формировании сниппета наравне с фрагментами текста страницы рассматривается содержимое этого тега...».

И самое главное, покажем, чего же ждет «Яндекс» от веб-мастера, описывающего содержимое метаобласти веб-страницы:

- «...Метаописания не должны быть короткими, в несколько слов.
- Должны описывать конкретную страницу сайта, а не сайт в целом.

- Мета-описания должны быть написаны для людей, нормальным человеческим языком — развернуто, правильно выстроенными предложениями, без злоупотреблений ключевыми словами, фразами, заглавными буквами, рекламными лозунгами и пр.
- Мета-описания должны быть емкими и при этом содержательными. Страйтесь выразить основную суть документа в нескольких предложениях.
- Содержимое тега `meta` должно быть актуальным, отражать текущее состояние страницы.
- Мета-описания должны соответствовать языку документа...».

Настройка представления сайта в результатах поиска

Пора познакомиться со сниппетом, это очень интересная штука. «Яндекс» рекомендует:

«...Сниппет — краткая „выжимка“ содержания документа, показывается в результатах поиска и располагается сразу под заголовком найденной страницы. Сниппет формируется автоматически на основе части текста страницы, содержащей слова запроса, и во многих случаях позволяет пользователю получить представление о документе или даже искомую информацию, не переходя по ссылке. В некоторых случаях сниппет может содержать дополнительную информацию, которую веб-мастер может передать самостоятельно...»

Каково?

При формировании сниппета учитывается следующее:

- «...регистр названия сайта;
- быстрые ссылки на разделы сайта (допускается только редактирование автоматически определенных ссылок);
- метаописания;
- содержимое сайта...».

Таким образом, совершенствуя контент своего сайта как в лингвистической области, так и в структурной организации, мы оказываем целенаправленное воздействие на поискового паука, который индексирует наш сайт.

Как видно из вышеизложенного материала, ничего страшного или слишком заумного в волшебном слове SEO нет. Нужно просто делать свое творение для конкретного пользователя — человека, которого вы желаете увидеть у себя в гостях. И не надо обманывать поискового паука — все равно ничего не получится.

Дарите людям радость, работая по известному с древности и вечно мудрому принципу: «Улыбайся! И люди к тебе потянутся!»

Глава 11

ОБЗОР СЕРВЕРНЫХ ТЕХНОЛОГИЙ

11.1. Что дальше

Набор средств и приемы работы, выполняемые при создании сайта и описанные в предыдущих главах, — это даже не вершина айсберга, а просто точка начала пути в мастерство. Но даже с этой точки видно, какими знаниями необходимо обладать веб-мастеру, чтобы только приступить к созданию своего сетевого шедевра. Мы же увидели лишь малую часть того, каким образом может развиваться создаваемый сайт. Причем все технологии, описанные ранее, можно отнести к тем, что выполняются на стороне клиента, за исключением CMS Joomla!, конечно. Но как бы ни были обширны возможности этих технологий — вспомним хотя бы объектную модель документа DOM и возможности манипулирования объектами веб-страниц, — для современного сайта этого мало. Требуется хранить колоссальные объемы данных и предоставлять одновременный доступ к ним для неограниченного количества пользователей. А для этого нужно применять программные продукты, далеко выходящие по своим возможностям за рамки динамического HTML.

Одним из решений этой проблемы как раз и является использование систем управления содержимым. Но CMS Joomla! — уже готовый программный продукт. С помощью чего он создан, как написать свою Joomla! или улучшить существующую, каким образом заставить сайт выполнять те или иные действия уже на стороне сервера, а не на стороне клиента? Постараюсь дать краткие ответы на эти вопросы.

11.2. Язык программирования PHP

PHP (англ. Hypertext Preprocessor — препроцессор гипертекста) — это сценарный язык программирования общего назначения для разработки веб-приложений. Сегодня он считается одним из лидеров среди языков программирования для создания динамических сайтов.

Это средство, с помощью которого реализуются поставленные выше задачи. Язык PHP как раз и предназначен для написания веб-приложений, работающих на стороне сервера. С его помощью в создаваемой программе

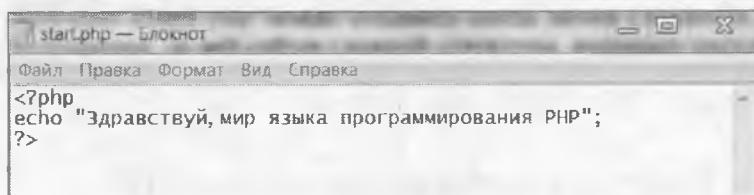
можно объединить все широчайшие возможности модели DOM, чтобы понапрасну не тратить трафик. Например, можно проверять корректность заполнения отправляемых форм и одновременно использовать возможность SQL-сервера хранить огромные объемы данных в своих таблицах, ограниченных только размером жесткого диска сервера. При этом будет возможен одновременный доступ пользователей к этим данным с помощью программы веб-сервера.

С помощью языка PHP можно создавать сайты любой сложности: от простой домашней веб-страницы до сайтов сложной структуры, например интернет-магазина или многопрофильного форума. Мы в этом убедились, когда рассматривали возможности создания и расширения сайта на основе CMS Joomla!.

Язык PHP может создавать и обрабатывать файлы, но это достаточно долго и сложно. Он также позволяет организовывать взаимодействие между различными частями приложения и базами данных, где хранится контент самого сайта. В этом случае написанные на нем приложения используются чаще.

Пожалуй, единственный недостаток PHP состоит в том, что он является языком-транслятором, а не компилятором. Поэтому для работы приложений, написанных на PHP, необходим предустановленный интерпретатор, переводящий текстовый код программы PHP в машинные команды, которые, в свою очередь, и будут выполнять серверы. В современном мире практически на всех платных хостингах и на части бесплатных установлен и поддерживается язык PHP.

Как же выглядит код программы на языке PHP? Очень просто — так же, как код на языке JavaScript, только со своим синтаксисом и в файле с расширением .php. Рассмотрим маленькую программу-сценарий, написанную на языке PHP (рис. 11.1).



```
start.php — Блокнот
Файл Правка Формат Вид Справка
<?php
echo "Здравствуй, мир языка программирования PHP";
?>
```

Рис. 11.1. Сценарий, написанный на языке PHP

Сам листинг сценария достаточно прост:

- <?php — тег, определяющий начало кода программы на языке PHP;
- echo "Здравствуй, мир языка программирования PHP"; — оператор вывода строки на экран, выводит надпись, заключенную в кавычки;

- ?> — тег, определяющий начало кода программы на языке PHP.

Для запуска этого сценария файл с ним необходимо разместить в зоне действия как интерпретатора языка PHP, так и веб-сервера, который и будет обслуживать выполнение сценария. Поскольку у нас уже установлен комплекс этих программ XAMPP (см. главу 9), то просто поместим файл со сценарием в каталоге `htdocs`, где размещали создаваемый сайт (рис. 11.2).

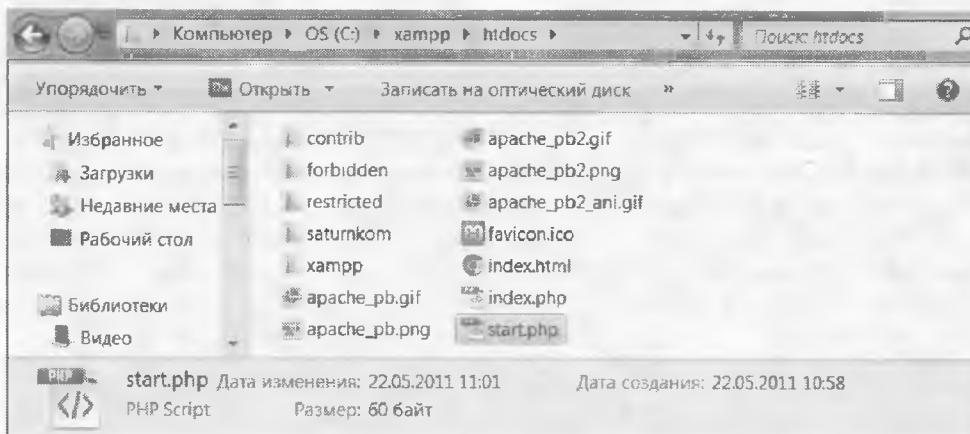


Рис. 11.2. Размещение файла `start.php`, содержащего сценарий, в каталоге `htdocs` для последующего выполнения

Теперь следует набрать в адресной строке нашего браузера адрес файла со сценарием: `http://localhost/start.php`, и можно наблюдать его выполнение (рис. 11.3).

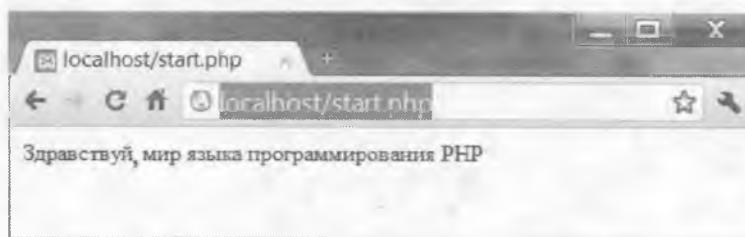


Рис. 11.3. Выполнение сценария, написанного на языке PHP

Как видите, ничего сверхъестественного в PHP нет, поэтому абсолютно не нужно его бояться, а следует совершенствовать свое мастерство, просто осваивая этот язык.

11.3. ASP-технология

В отличие от языков программирования PHP и JavaScript, технология ASP не является в буквальном смысле языком программирования. Это просто технология предварительной обработки информации, позволяющая подключать программные модули во время формирования веб-страницы, то есть позволяющая динамически создавать веб-страницы на стороне сервера.

ASP-технология — это детище компании Microsoft. Она заточена на программные продукты именно этой компании, в частности в операционных системах Windows и на веб-сервере Microsoft IIS. Широкую популярность технология получила благодаря использованию сценарных языков VBScript или JavaScript и возможности применения подключаемых внешних COM-компонентов. На текущее время технология развивается под новым именем, как и вся современная продукция компании Microsoft. ASP.NET — это имя новой технологии создания веб-приложений, основанной на платформе Microsoft.NET.

На самом деле различных языков программирования или технологий формирования веб-страниц динамическим образом на сегодня много, но это не значит, что для успешной работы в области сайтомостроения необходимо владеть всеми — это недостижимый идеал. Ведь постоянно отмирают старые, рождаются новые и совершенствуются живущие способы и методы создания веб-страниц. Начните с малого, а затем совершенствуйтесь в той области, к которой у вас будет лежать душа.