

УЗБЕКИСТОН АЛОҚА ВА АХБОРОТЛАШТИРИШ АГЕНТЛИГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
УРГАНЧ ФИЛИАЛИ

Web дастурлаш

фанидан маъруза матнлари

М.С. Шарипов

Урганч 2007

МУНДАРИЖА

- 1-маъруза . JavaScript. Клиент томондан дастурлаш. JavaScript га кириш.
JavaScript ни HTML-документга жойлаштириш.....
- 2 -маъруза JavaScript да маълумотлар типлари, ўзгарувчилар, ифодалар
ва арифметик операторлар.
- 3-маъруза Java Script дастурини бошқариш элементлари
- 4-маъруза JavaScript дастурини бошқариш элементлари.....
- 5-маъруза Усуллар ва функциялар тушунчаси .Объект ва хоссалар
Java Script да документларнинг объект моделлари.....
- 6-маъруза Java Script да ходисни кайта ишлаш .Интерактив формалар
- 7-маъруза Cookie файллар ва Java Script да хавсизлик.....
- 8 -маъруза PHP .Сервер томонидан дастурлаш .PHP га кириш Ўрнатиш
ва тестлаш.....
- 9-маъруза Виртуал серверга эҳтиёж Apache ни ўрнатиш
.....
- 10-маъруза PHP нинг Асосий синтаксис .Маълумотларнинг типлари
,Маълумотларни бошқариш
структуралари.....
- 11-маъруза Функция .Синф ва объектлар .Хатоларни кайта ишлаш
.PHP функциялар
.....
- 12-маъруза MY SQL да маълумотлар базаси .MY SQL да маълумотлар
базаси Билан ишлаш тамоиллари.....
- 13 Адабиётлар

МАЪРУЗА №1 Javascript га кириш. Javascript ни html-документга жойлаштириш.

Режа 1 JavaScript га кириш

2 Java Script сценарийси ёзиш учун инструментлар

3 JavaScript ни HTML-документга жойлаштириш.

JavaScript га кириш . **Java Script** бу Web-саҳифаларни янада интерактив ва функционал қилиш имконини берувчи сценарийлар тайёрловчи тилдир. Web-саҳифаларни кузатган булсангиз ҳар бир саҳифада бирор матн, бир нечта расмлар, вақтни курсатиб турувчи соат ёки браузернинг ҳолатлар сатрида югурувчи матнни қуришингиз мумкин. Ёки бирор бир тулдирилиши керак булган форма булиши мумкин. Аган анкетанинг бирор бир қисмини тулдиришни унутиб қолдирсангиз хатолик ҳақида хабар пайдо бўлади. Айрим бир саҳифаларда экран бўйлаб ҳаракат қилувчи расм ёки мантнни учратишингиз мумкин.

Шу каби ҳаракатларни **Java Script** да қузатишингиз мумкин, шунга ухшаш эффектларни эса ушбу тилни билувчи ихтиёрый киши яратиши мумкин.

Java Script ни қуллаб-қувватловчи браузер Netscape2.0 , булган.

Netscape браузерларида Java Script нинг қуллаб-қувватланиши

Браузер	Версия
Netscape 2.0	1.0
Netscape 3.0	1.1
Netscape 4.0	1.2
Netscape 4.5	1.3/1.4
Netscape 6.0	1.5

Internet Explorer браузерларида Java Script нинг қуллаб-қувватланиши

Браузер	Версия
Internet Explorer 3.0	1.0/2.0
Internet Explorer 4.0	3.0
Internet Explorer 5.0	5.0
Internet Explorer 5.5	5.5

Java Script 1.5 ва JScript5.5 Web-саҳифангизнинг қуриниши ва унинг функциясини бошқариш имконини берувчи сценарийлар ёзиш тилларидир. Java Script ни хужжатнинг объектли модели (Document Object Model- DOM) ва стилларнинг каскадли жадвали (Cascading Style Sheets -CSS) каби янги технологиялар билан биргаликда қуллаш билан web-саҳифаларингиздаги барча ходисаларни назорат қилишингиз мумкин. Java Scrip ва объектли моделдан фойдаланиб, сичқончанинг босилиши ва клавиатурадан маълумотнинг киритилишига таъсир қилувчи саҳифалар яратиш мумкин. Каскадли стиллар жадвали ёрдамида саҳифадаги матнларни курсатиш ва беркитиш, браузер ойнаси улчамларини бошқариш ва шу қабиларни бажариш мумкин.

Ушбу технологияларни куллаш билан сиз нафакат Web-саҳифа яратасиз балки бошқа компьютер дастурлари каби ҳаракат килувчи фойдаланувчи учун муҳит яратасиз.

Сценария ва дастур орасидаги фарқ нима

Сценария (script) – бу буйруқлар кетма-кетлиги (айрим ҳолларда ҳаттоки дастур) булиб, бошқа дастур томонидан қайта ишланади ва интерпретация қилинади. Бу эса сценарий ёзиш учун матн редактори етарли эканлигини англатади, дастур яратиш учун эса бошқа бир дастур (ҳеч булмаганда компилятор) талаб қилинади. Бошқа суз билан айтганда сценарий C ва C++ каби мураккаб таркибли тилларда яратилувчи дастурлардан кура осон ва тез ёзилади.

Бирок сценарий бажарилиши учун компиляция қилинган дастурнинг бажарилишидан кура куп вақт талаб қилинади. Чунки сценарийнинг ҳар бир буйруги процессор остида эмас, балки бошқа дастур ёрдамида қайта ишланади. Ушбу камчилик сценарийни яратиш учун бупул матн редакторининг етарлилиги ва айни пайтда C++ тили учун компиляторнинг бир неча юз доллар туриши билан изоҳланади.

Java Script сценарийси ёзиш учун инструментлар

Оддий матн редактори

Оддий матн редакторини (масалан Windows Notepad) танлаш ва унга кодни ёзиш осон булса керак. Балки бу Java Script да сценарий ёзишнинг энг осон йули булмаслиги мумкин, бироқ унда қуйидаги ишончли афзалликлар мавжуд:

- Бепуллиги
- Мураккаб дастурни урганиш талаб қилинмайди
- дастурий таъминот комплекти эмас Java Script урганилади

Визуал таҳрирлаш воситаларидан фойдаланиш

Java Script сценарийларини ёзишда визуал таҳрирлаш воситалари сифатида одатда HTML редакторларидан фойдаланилади. Бу турдаги воситаларга Microsoft Front Page, Macromedia Dreamweaver ва Adobe GoLive ларни киритиш мумкин.

Бунда қупинча Java Script кодининг қулда киритилиши ва текширилиши талаб қилинмайди ва бу дастлаб жуда аҳамиятли булиб туюлиши мумкин. Бироқ ишни энгиллаштириш билан ихчамликни йукотиш мумкин.

Бироқ визуал таҳрирлаш воситаларининг ҳам бир нечта афзалликлари ҳам мавжуд:

Тезлик-қулда терилгандан кура натижага тезроқ эришилади;

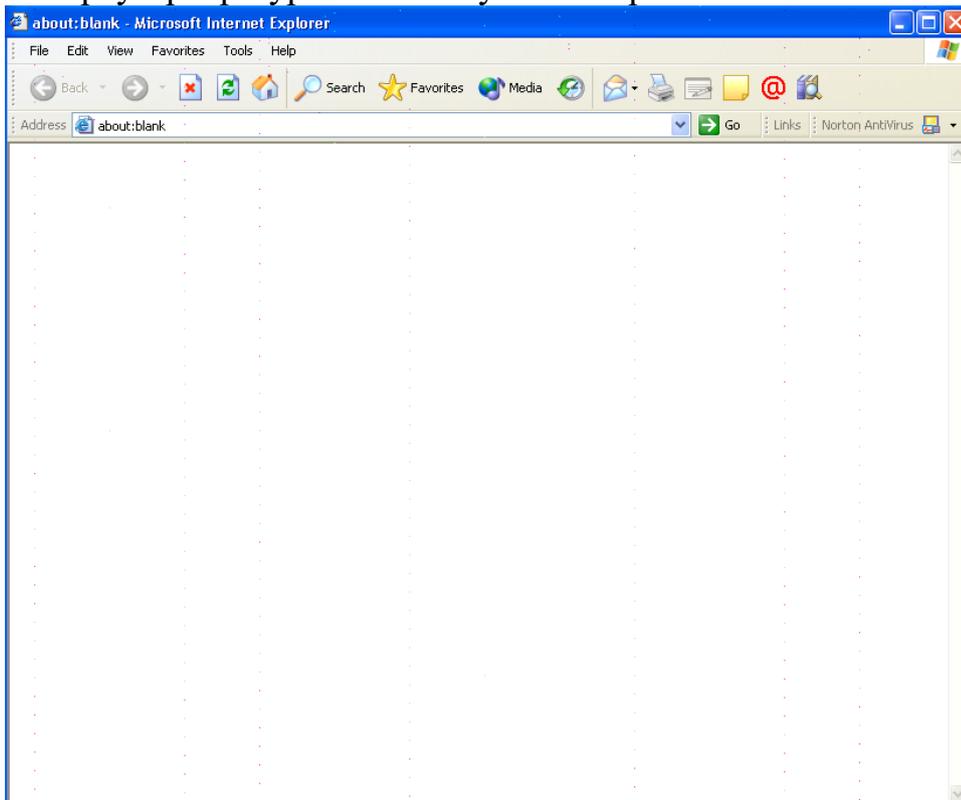
Самарадорлик- визуал таҳрирлаш воситалари ёрдамида ишлайдиган сценарийни олиш мумкин;

Махсус функциялар- масалан ранг билан белгилаш кодларни уқиш имконини осонлаштиради.

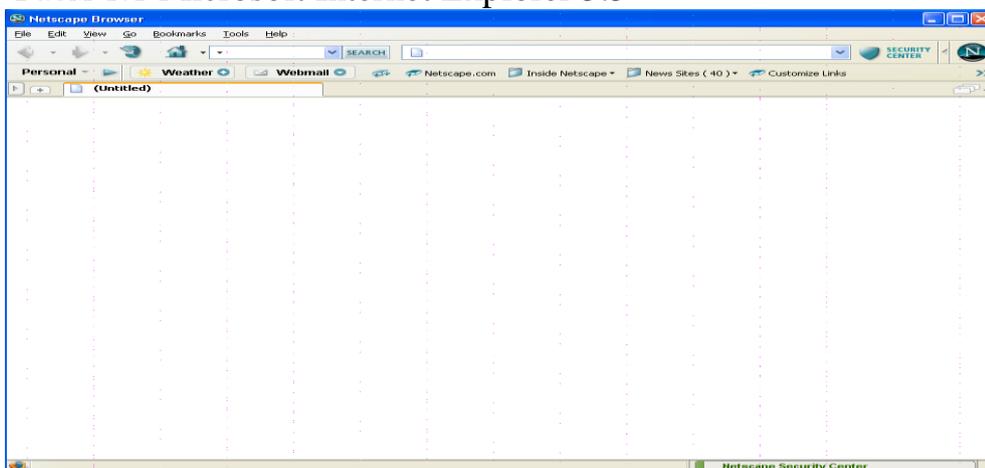
Браузернинг сунги версияси

Сиз Java Script нинг замонавий версиясини урганар экансиз унинг барча имкониятларидан фойдаланиш учун браузерларнинг энг сунги версиясига эга

булишингиз зарур. Минимум Netscape Navigator 4.0 ва Microsoft Internet Explorer 5.5 браузерлари урнатилган булиши керак.



Расм 1.1 Microsoft Internet Explorer 5.5



Расм 1.2 Netscape Navigator 6

Бошка инструментлар

Java Script билан ишни осонлаштириш учун HTML саҳифа шаблонларидан фойдаланиш ва шахсий сценарийлар кутубхонасини яратиш керак.

HTML шаблон

```
<html>
<head>
<title>Html sahifa shabloni</title>
<script language="javascript">
<!--
// - - >
</script>
```

```
</head>
<body>
</body>
```



```
</html>
```

Расм 1.3 Windows Notepad редактори ёрдамида тайёрланган HTML шаблон

Java Script ни HTML саҳифаларга тадбик қилиш

HTML ва Java Script мутлақо бошқа-бошқа технология булганлиги сабабли HTML хужжатга Java Script ни тадбик қилишнинг уз қоидалари мавжуддир. Web-саҳифаларга Java Script ни қиритишнинг бир нечта усуллари мавжуд, бироқ дастлаб энг қуп тарқалган ва ихчам турини қуриб чикамиз, яъни Java Script ни блоклаб <head> тэгига қуйиш усули.

HTML хужжати

Web – саҳифанинг оддий HTML тарқибини қуриб чикайлик. Натижа 2.1 расмда келтирилган:

```
<html>
<head>
<title>Oddiy sahifa</title>
</head>
<body>
<p> Judayam oddiy Web sahifa! </p>
</body>
</html>
```

Бу ерда эътибориниғизни HTML коднинг <head> тэги билан бошланувчи ва </head> тэги билан ёпилувчи блокка қаратинг. Бу HEAD блоқи ёки Web саҳифанинг сарлавҳаси булиб, айнан шу ерда сиз уз Java Script сценарийлариниғизни жойлаштирасиз.

Script блоқи

Head блоқида Java Script сценарийсини қиритишдан олдин унга янги HTML блоқини қиритиш шарт. Бу блок SCRIPT булиб, унинг вазифаси браузерга мазкур блоқнинг очилувчи ва ёпилувчи тэглари орасида махсус сценарий жойлашганлигини қурсатишдир. Бу блок HEAD блоқнинг ихтиёрий жойида қуйилиши мумкин, бироқ одатда у </head> ёпилувчи тэги олдидан қуйилади:

```
<html>
```

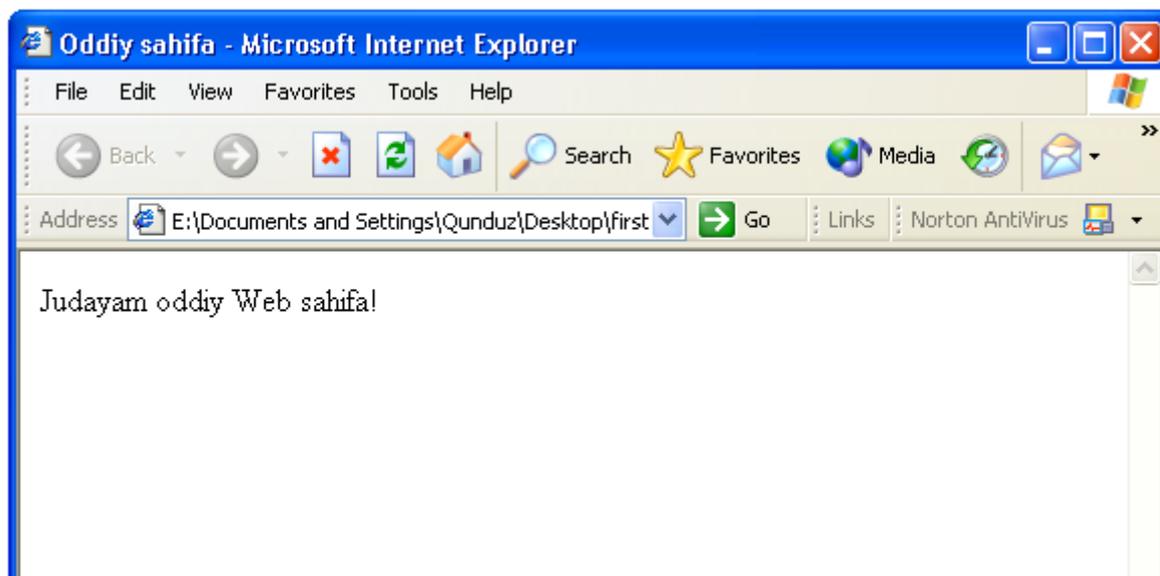
```
<head>
<title>Oddiy sahifa</title>
<script>
</script>
</head>
<body>
<p> Judayam oddiy Web sahifa! </p>
</body>
</html>
```

SCRIPT блокига Java Script сценарийсини куйишдан олдин яна бир неча харакатлар бажарилиши керак. Биринчидан браузерга SCRIPT блокада кандай турдаги сценарий жойлашишини курсатиш керак. Чунки Java Script сценарий тайёрловчи ягона тил эмас. Microsoft компаниясининг VBScript тили хам яна бир сценарий тайёрловчи тил хисобланади. Демак, Java Script ни изохлаш учун <script> тэгига language атрибутини кушилади:

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language=" ">
</script>
</head>
<body>
<p> Judayam oddiy Web sahifa! </p>
</body>
</html>
```

Ва Java Script дан фойдаланаётганлигингиз хакида Java Script сузини куштирнок ичида курсатиб утилади.

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript ">
</script>
</head>
<body>
<p> Judayam oddiy Web sahifa! </p>
</body>
</html>
```



Назорат саволлари

1. JavaScript дастурлаш тилининг вазифаси ?
2. Броузернинг вазифаси ?
3. Броузерлар турлари ?
4. Сценария нима ?

МАЪРУЗА №2

JavaScript да маълумотлар типлари, ўзгарувчилар,
ифодалар ва арифметик операторлар

Режа 1 JavaScript да маълумотлар типлари

2 JavaScript да ўзгарувчилар

3 Ифодалар ва арифметик операторлар

2.1 JavaScript да маълумотлар типлари Оламдаги энг кадрли нарса бу информациядир. Java Script да информациянинг хар бир булагини киймат (value) мисолида курилади. Информация хар хил булгани каби кийматларнинг хам турли категориялари мавжуддир. Java Script да оддий маълумотлар турлари асосий турлар (primitive types) деб номланади.

3 та асосий маълумотлар тури мавжуд:

сатр (string) ; сон (number); мантикий (boolean).

Сатрлар

Сатр турдаги маълумотлар энг куп таркалган булса керак. Сатр – харфлар, сонлар ва тиниш белгиларни уз ичига олувчи белгилар кетма-кетлигидир. Java Script да сатрлар купинча матнлар ифода килиш учун ишлатилади.:

Assalomu alaykum, hush kelibsiz!

Siz kim?

Java Script сценарийсидаги сатрлар апостроф ёки куштирнокка олинади, масалан:

“Assalomu alaykum!”

Иккита турнинг ишлатилишининг сабаби шундаки, куштирнок ичида апострофли сатр ишлатиш ва аксинча мумкин. Масалан:

“Mening o’girligim 60 kg”

“”Siz kimsiz?”- deb hayron buldi u’

Сон

Java Script 2 та турдаги сонларни кабул килади: бутун (integer) ва кузгалувчи нуктали сон (floating-point number).

Бутун сонлар. Бутун сонлар узида мусбат масалан, 1, 2, 3 ва манфий -1, -2, -3 ва сонларни ва 0 ни намоён килади.

Кузгалувчи вергулли сонлар. Кузгалувчи вергулли сонлар

Катта ва кичик сонлар

Java Script да ишлатиладиган сонлар жудаям катта ёки жудаям кичик булиши мумкин. Энг катта сон сифатида 10^{308} ва энг кичик сон сифатида 10^{-308} киймат каралади.

Мантикий ифодалар

Мантикий ифодалар сатр ва сонлардан факат 2 та киймат кабул килиши билан фаркланади: true (рост) ва false (ёлгон).

Алохида турдаги маълумотлар: сонлар, ноаник ва аникланмайдиган ифодалар

КУриб чикилган маълумотлар туридан бошка бир нечта турлар мавжуд булиб куйида сонли кийматларнинг турлари келтирилади:

- Мусбат чексизлик
- Манфий чексизлик
- Мусбат ва манфий нуль
- Мавжуд булмаган сон (not a number -NaN)

Яна 2 та махсус маълумотлар тури мавжуд:

- Ноаник (Null)
- Аникланмайдиган (Undefined)

Дастлаб келтирилган 4 та тур хакида гапирадиган булсак, сиз улар билан ишлашингизга тугри келмайди (албатта сиз дастурингизда чексиз сондан фойдаланмайсиз). Мавжуд булмаган сон деганда маъносиз математик операциялар натижаси тушунилади (масалан, нолга булиш). Чексиз киймат деганда 10^{308} дан катта сон тушунилади.

Ноаник турдаги маълумот факат битта null кийматни олиши мумкин.

Узгарувчилар нафакат Java Script тилида балки барча дастурлаш тилларида музим ахамиятга эгадир. Улар ёрдамида барча турдаги маълумотларни бошкариш мумкин.

Узгарувчи- Java Script сценарийси бажарилаётган пайтда маълум турдаги маълумотларни сакловчи компьютер хотирасининг номланган катакчаси хисобланади. Бу таъриф бироз мураккаб туюлиши мумкин, лекин узгарувчилардан фойдаланиш жудаям осондир.

2.2 Java script да ўзгарувчилар Узгарувчи яратиш

Java Script тилида узгарувчи яратиш жудаям осондир. Яна уша HTML шаблонига мурожаат киламиз.

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript">
</script>
</head>
<body>
</body>
</html>
```

Дастлаб узгарувчини яратиш керак. Буни 2 йул билан олдиндан ва зарур пайтда эълон кили шоркали яратиш мумкин. Дастлаб олдиндан эълон килинадиган турни куриб чикамиз. Ушбу булим охирида «зарур пайтдаги» узгарувчиларнинг яратилишини куриб чикамиз.

Мисол: Java Script тилида узгарувчини эълони килиш (яратиш) учун var оператор ива ундан сунг узгарувчи номи курсатилади. Ушбу мисолда msg номли узгарувчи эълон килинган:

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript ">
var msg;
</script>
</head>
<body>
</body>
</html>
```

Узгарувчи номлари хакида куйидагиларни билишингиз керак:

- Узгарувчилар номида катта ва кичик регистр белгиларини ва уларни биргаликда ишлатиш мумкин;
- Узгарувчини эълон килгунга кадар унинг киймати undefined хисобланади;
- Узгарувчи номи сонлардан бошлана олмайди;
- Узгарувчилар номида пробеллар ишлатилиши мумкин эмас, агар уларни ажратиш керак булса, тагига чизиш белгисидан () фойдаланинг;
- Узгарувчилар номида доллар (\$) белгисини ишлатишдан эхтиёт булинг чунки, Internet Explorer3.02 ва Netscape Navigator 2.02 браузериде танилмайди.

Куйида тугри номланган узгарувчилар келтирилган:

```
Msg
Hello_world
Msg1
Msg_1
```

Куйида нотугри номланган узгарувчилар келтирилган:

```
1msg – сон билан бошланган
Hello world – пробелга эга
Var – Java Script нинг хизматчи оператори
Dollar$ - номда $ белгиси мавжуд
```

Msg ва msg Ушбу номларни битта сценарийда ишлатманг, чунки улар факат регистр билан фаркланади.

Мисол. Агар сиз эълон килинган узгарувчига шу захоти киймат бермокчи булсангиз буни уша сатрнинг узида килиш мумкин:

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript ">
var msg="Java Script ning uzgaruvchilar olamiga hush kelibsiz!";
```

```
</script>
</head>
<body>
</body>
</html>
```

Мисол. Узгарувчини эълон кили шва унга киймат бериш алохида сатрларда ҳам амалга оширилиши мумкин:

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript ">
var msg;
msg="Java Script ning uzgaruvchilar olamiga hush kelibsiz!";
</script>
</head>
<body>
</body>
</html>
```

Мисол. Агар сиз узгарувчингизнинг киймати сатр эмас, сон булишини хохласангиз уни куйидагича килишингиз мумкин:

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript ">
var msg=22;
</script>
</head>
<body>
</body>
</html>
```

Бир нечта узгарувчиларни эълон килиш

Битта эмас балки бир нечта узгарувчиларни эълон килиш керак булса, уларни хар бирини алохида сатрда ёки битта сатрда эълон килиш мумкин.

```
var msg1; var msg2; var num1; var num2;
ёки var msg1, msg2, num1, num2;
```

Бир нечта узгарувчини битта сатрда эълон килганда улар вергул билан ажратилиши ва охирида нукта -вергул билан тугаши керак. Агар вергкл билан ажратилмаса, хатолик юз беради.

Мисол. Битта сатрда узгарувчиларни эълон килишингиз ва киймат беришингиз мумкин.

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript ">
var msg1="Barchaga", msg2="salom", num1=6, num2=52;
```

```
</script>
</head>
<body>
</body>
</html>
```

Узгарувчилар кийматини экранга чикариш

Демак, сиз бир нечта узгарувчини эълон килдингиз ва уларга сатр ва сон кийматини бердингиз. Энди ушбу узгарувчиларни амалда ишлатиб курамиз. Куйидаги кичик мисолда alert() методи узгарувчи кийматини экранга чикариш учун ишлатилади (Расм 4.1).

Мисол. 2 та узгарувчи msg1 ва num1 ни эълон килинг ва уларга “Barchaga salom” (сатр) ва 22 (сон) кийматларни беринг:

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript">
var msg1="Barchaga salom", num1=22;
</script>
</head>
<body>
</body>
</html>
```

Кейин alert() методидан фойдаланиб узгарувчини методнинг кавслари орасига жойлаштиринг.

```
<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript">
var msg1="Barchaga salom", num1=22;
alert(msg1);
</script>
</head>
<body>
</body>
</html>
```



Расм 4.1 msg1 узгарувчиси кийматини огохлантирувчи дарчада куриниши

Энди сиз иккинчи узгарувчининг кийматини бошка дарчада чикаришингиз мумкин (расм 4.2).

```

<html>
<head>
<title>Oddiy sahifa</title>
<script language="JavaScript ">
var msg1="Barchaga salom", num1=22;
  alert(msg1);
  alert(num1);
</script>
</head>
<body>
</body>
</html>

```

Саҳифангизни яна сакланг ва киритилган узгаришларни куриш учун уни браузерингизда кайтадан ишлатинг. Бу сафар огохлантирувчи иккита дарча кетма-кет намоён булади ва иккинчи дарчада 22 сони пайдо булади (расмга қаранг).



Расм 4.2 num1 узгарувчиси қийматини огохлантирувчи дарчада қуриниши

2.3 Ифодалар ва арифметик операторлар

Ифода ва шарт

Узгарувчи ва массивлар ёрдамида сиз узингизга керакли булган информацияни турли қуринишда компьютер хотирасида саклайсиз. Бирок бу информацияни сиз фақатгина экранга чиқариш ёки огохлантирувчи дарчада жойлаштириш учун эмас, балки купрок амаллар бажаришингиз мумкин. Сиз уни узгартиришингиз, уни бошқаришингиз ёки текширишингиз мумкин. Ушбу мақсадлар учун ифода ва шартлар келтирилган.

Ифодалар(expressions) икки ёки ундан ортиқ қийматларни бирлаштириш учун ва натижада янги қийматга эга булиш учун ишлатилади. Қуйидаги йигиндиле ифода бунга мисол була олади:

$$1+2=3$$

Шартлар (conditions) қийматларни солиштириш ва мантикий қийматни (true ёки false) аниқлаш имконини беради.

$$3+3=6 \text{ ? true}$$

Амаллар билан танишув

Шарт ва ифодаларда маълумотлар бир-бири билан амаллар билан бирлаштирилади. Агар ҳаракат битта катталиқ билан амалга оширилса у ҳолда бу амал **битта операндли амал** (unary operator) деб аталади. Агар бунақа

катталиклар 2 та булса, у холда бундай амаллар **2 та операндли амаллар** (Binary operator), агар 3 та булса- **3 та операндли амаллар** (ternary operator) дейилади.

Арифметик амаллар

Арифметик амаллар бу барчага маълум булган математик амаллар хисобланади:

- кушиш (+) $1+3=4$
- айириш (-) $2-1=1$
- булиш (/) $4/2=2$
- купайтириш (*) $2*2=4$
- булишдан колган колдик (%) $9\%5=4$

Булар 2 та операндли амалларга мисоллар эди.

Java Script сценарийларига мисоллар

Куйида арифметик амаллардан фойдаланилган бир нечта мисоллар келтирилади. Бу ерда SCRIPT блокнинг узи берилган.

Мисол. Кушиш амалига мисол 6.1- расм

```
<script language="JavaScript" >  
var a=6, b=4;  
alert(a+b);  
</script>
```



6.1 расм

Мисол. Айириш амалига мисол 6.2- расм

```
<script language="JavaScript" >  
var a=6, b=4;  
alert(a-b);  
</script>
```



6.2 расм

Мисол. Булиш амалига мисол 6.3- расм

```
<script language="JavaScript" >  
var a=6, b=3;  
alert(a/b);  
</script>
```



6.3 расм

Мисол. Купайтириш амалига мисол 6.4- расм
`<script language="JavaScript ">`
`var a=6, b=3;`
`alert(a*b);`
`</script>`



6.4 расм

Мисол. Булишдан колган колдик амалига мисол -6.5 расм
`<script language="JavaScript ">`
`var a=6, b=4;`
`alert(a%b);`
`</script>`



6.5 расм

Такослаш амаллари ифодаларни бир-бири билан солиштириш ишлатилади. Уларга куйидаги амаллар киради:

- кичик (<) $5 < 6$
- кичик ёки тенг (<=) $6 <= 6$ ёки $6 <= 7$
- катта (>) $7 > 4$
- катта ёки тенг (>=) $5 >= 5$ ёки $5 >= 4$
- тенг (==) $5 == 5$
- тенг эмас (!=) $5 != 3$

Буларнинг барчаси 2 та операндли амалларга киради. Бу мисолларда сонли маълумотлар такосланди, бироқ сатрларда ҳам ушбу амаллар ишлатилиши мумкин (бу ҳақда кейинроқ суз юритамиз). Энг асосийси шуки, битта турдаги маълумотлар такосланилиши мумкин. Акс ҳолда Java Script бир турдаги маълумотларни бошқа турга утказишга

харакат килади ва бу хар доим хам амалга оширилаверилмайди. Хатоларга йул куймаслик учун битта турдаг маълумотларни таккосланг.

Мантикий амаллар

Куйида 3 та мантикий амаллар келтирилган:

- мантикий ХА (and) &&
- мантикий ЁКИ (or) ||
- мантикий ЙУК (not) !

Битта операндли амаллар

Номидан хам куришиб турибди бу амал битта катталиқ билан амалга оширилади. Буларга:

- префиксли ва постфиксли усиш (increment) - ++
- префиксли ва постфиксли камайиш (decrement) - --
- унар плюс +
- унар минус -

Унар минус ифода ишорасини карама-каршисига айлантиради. 4 та амал ичида бу энг оддийси хисобланади. Бунан ташкари унар плюс ишорани алмаштириш учун эмас, балки операндни сонга айлантириш учун ишлатилади (масалан, бу сатр булган холда).

Префикс/постфикс усиш ва камайиш амали узгарувчи кийматини биттага усиши ва камайиши учун хизмат килади. Бирок бу амаллар натижаси бу амалнинг преяикс ёки постфиксли эканлигига богликдир.

Префиксли амалда агар $a=5$ булса, у холда $++a+2=8$ булади, чунки узгарувчи кийматига 2 сони кушилмасидаг олдин у биттага орттирилган эди. Шу вақтнинг узида $-a+2=6$, чунки узгарувчи киймати 2 га камайтирилмасдан олдин у биттага камайтирилган эди.

Постфиксли амалда эса аксинча, ифодада узгарувчининг эски киймати ишлатилиб, сунгра унинг киймати биттага камайтирилади ёки орттирилади. Шу тарика агар $a=5$ булса, у холда $a++ +2=7$, чунки узгарувчининг киймати ифода хисоблангандан сунг биттага орттирилади. Бу ерда $a-- +2=7$ хам уринлидир, чунки узгарувчининг киймати ифода хисоблангандан сунгина биттага камайтирилади.

Мисол. Префиксли усиш – 6.6 расм

```
<script language="JavaScript ">
var a=6, b=2;
alert(++a+b);
</script>
```



Узлаштириш амали

Оддий узлаштириш амали (=) булиб, у узгарувчига кийматни юклаш учун хизмат килади. Сиз бу амал билан танишсиз, буни а узгарувчига 3 сонини юклашни мисол килиш мумкин $a=3$.

Бу амалдан фойдаланиб битта кийматни бир нечта узгарувчига юклашни хам бажариш мумкин. Куйидаги мисолда 5 кийматини бир нечта (a,b,c ва d) узгарувчиларига узлаштириш курсатилган:

$$a=b=s=d=5$$

Бошка узлаштириш амаллари кушма амаллар хисобланиб куйида улар келтирилади:

- $x+=3$ бу амал $x=x+3$ билан тенг кучлидир
- $x-=3$ бу амал $x=x-3$ билан тенг кучлидир
- $x*=3$ бу амал $x=x*3$ билан тенг кучлидир
- $x/=3$ бу амал $x=x/3$ билан тенг кучлидир

Мисол. += кушма амали – 6.10 расм

```
<script language="JavaScript">  
var x=6;  
alert(x+=2);  
</script>
```



6.10 расм

Мисол. -= кушма амали –6.11 расм

```
<script language="JavaScript">  
var x=6;  
alert(x -=2);  
</script>
```



6.11 расм

Java Script тилида операторлардан фойдаланиш

Сиз узгарувчи, массив ва амалларни урганиб олдингиз, энди Java Script операторларига утсак ҳам булади.

Операторлар (statements) Java Script тилининг кучли инструменти хисобланади. Операторларни ургани олганингиздан сунг сизнинг сценарийларингиз юкори даражага кутарилади ва мураккаб вазифаларни бажариш имконига эга буласиз.

Ушбу булимда куйидаги мавзулар куриб чикилади:

- операторлар вазифалари
- variable оператори
- if оператори
- цикл ташкил килувчи оператор
- continue оператори
- break оператори
- switch оператори

Оператор нима?

Java Script нинг барча сценарийлари унинг хажми ва мураккаблигидан катъий назар операторлар кетма-кетлигидан ташкил топган булади.

Мисол. Экранга огохлантирувчи дарчани чиқариш операторнинг бажарилиши натижаси хисобланади:

```
<script language="JavaScript" >  
alert("Bu ham operator!");  
</script>
```

Мисол. Куйида операторга яна бир мисол келтирилади:

```
<script language="JavaScript" >  
var ourname="Talaba va domla";  
</script>
```

Java Script да нуктали вергул операторнинг тугадини билдиради. Уларни куйиш шарт эмас, агар сиз куймасангиз Java Script нинг узи сиз учун куяди. Бирок улардан фойдаланиш куйидаги сабаблар учун керак:

- сизнинг сценарийингиз осон укилади
- нуктали вергулни куймаслик айрим холларда хатоликка олиб келади, чунки Java Script хар доим ҳам уларни тугри куймайди

variable оператори

Мисол. Variable оператори билан танишсиз. Яъни

```
Var this_year=2001;
```

Variable оператори this_year узгарувчиси яратилганлиги ва унга 2001 киймати берилганлигини билдиради.

If оператори

If операторидан буйруқлар кетма-кетлигини бошқариш ва турли максадларда фойдаланилади. Java Script сценарийсида бу оператор 2

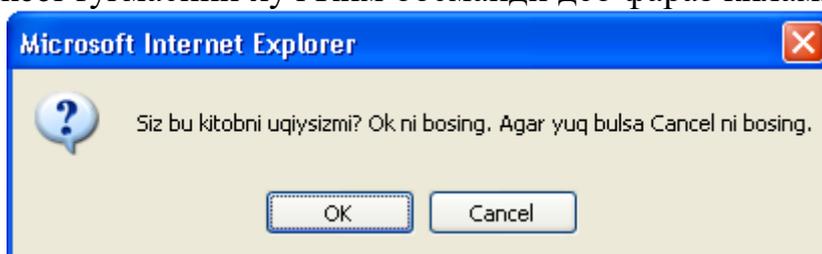
та гурух операторлардан биттасини танлаш ва бажариш имконини беради. Танлаш Буль ифодалар ёрдамида амалга оширилади(true ва false).

Масалан, тасдиқлаш дарчасининг if оператори ёрдамида кандай иш куриши билан танишайлик.

Мисол. Дастлаб confirm() методи ёрдамида оддаий тасдиқлаш дарчасини хосил қиламиз:

```
<script language="JavaScript ">  
confirm("Siz bu kitobni uqiysizmi? Ok ni bosing. Agar yuq bulsa Cancel ni  
bosing.");  
</script>
```

Фойдаланувчига 2та тугмачадан биттасини танлаш имкони берилади(7.1-расм). Келинг, ОК тугмасини босган фойдаланувчилар учун код ёзамиз (Cancel тугмасини хуч Ким босмайди деб фараз қиламиз).



7.1-расм

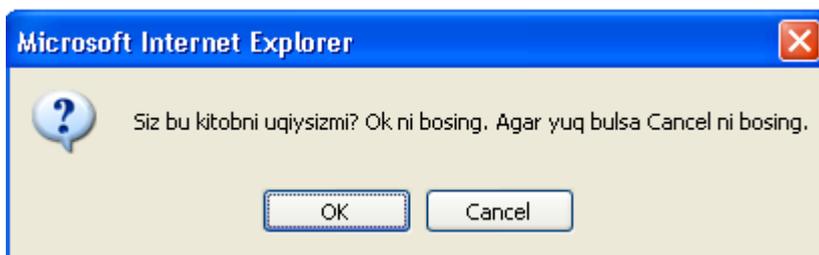
Фойдаланувчи ОК тугмачасини босган пайтда Буль ифода true кийматини олади ва сиз танлов амалга оширилганини билишингиз мумкин. Бунинг учун узгарувчи эълон қилишингиз ва унга confirm() методи кийматини юклашингиз керак булади:

```
<script language="JavaScript ">  
var response=confirm("Siz bu kitobni uqiysizmi? Ok ni bosing. Agar yuq bulsa  
Cancel ni bosing.");  
</script>
```

response узгарувчиси саволга жавобни узида саклаб қолади. prompt() методи ёрдамида фойдаланувчи киритган сатрни ҳам узгарувчи саклаши мумкин. Бошқача қилиб айтсак, агар фойдаланувчи Ок тугмачасини босса, у холда response узгарувчиси true кийматини узлаштиради. Бунга ишонч хосил қилиш учун if операторидан фойдаланамиз.

Мисол. Бу Java Script да қуйидагича амалга оширилади:

```
<script language="JavaScript ">  
var response=confirm("Siz bu kitobni uqiysizmi? Ok ni bosing. Agar yuq bulsa  
Cancel ni bosing.");  
if (response==true)  
{  
alert("Zur tanlov");  
}  
</script>
```



7.2-расм

Шу тарика if оператори куйидаги схема оркали ифодаланади:

```
If (шарт){
```

```
Операторлар
```

```
}
```

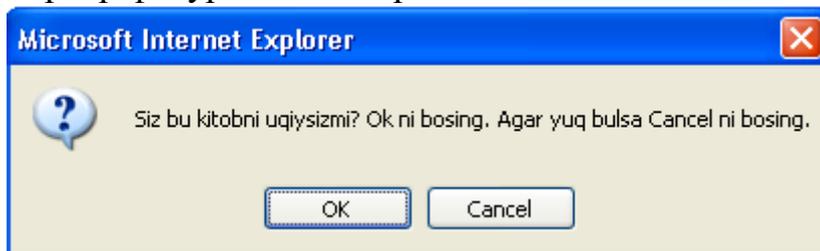
Шарт доимо бул ифода талаб килади. Агар шарт рост (true) кийматни олса фигурали кавс ичидаги операторлар бажарилади.

Агар шарт натижаси Null ёки undefined булса, бу false (ёлгон) киймати билан тенг кучли хисобланади.

Мисол. Энди сизнинг сценарийингиз ок тугмасини босгандан сунг аник бир амалларни бажаради. Cancel тугмасини босгандан сунг if операторининг 2-кисми if...else ишга тушади.

```
<script language="JavaScript ">
var response=confirm("Siz bu kitobni uqiysizmi? Ok ni bosing. Agar yuq bulsa
Cancel ni bosing.");
if (response==true)
{
alert("Zur tanlov");
}
else
{
alert("Ishonchingiz komilmi? Java Script ni urganish judayam quziqarli!")
}
</script>
```

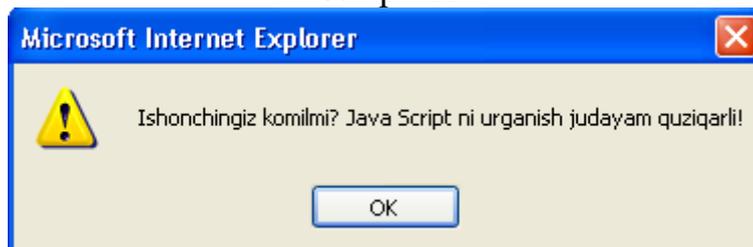
Else кисми ёпилувчи фигурали кавсдан кейин куйилади ва у уз ичига оладиган операторлар фигурали кавслар ичига олинади. 7.3, 7.4, 7.5-расм



7.3-расм



7.4-расм



7.5-расм

Шу тарика if...else оператори куйидаги схема асосида курилади:

```
If ()
```

```
{
```

```
операторлар
```

```
}
```

```
else
```

```
{
```

```
операторлар
```

```
}
```

Назорат саволлари

1 Java Script қандай маълумотлар типлар ?

2 Java Script Ўзгарувчилар турлари ?

3 Java Script Ифодалар турлари ва арифметик оператор ?

4 Java Script да қандай шартли операторлари турлари ишлатилиди ?

МАЪРУЗА №-4 Java Script дастурни бошкариш элементлари

РЕЖА 1 Javascript да формалар.

2 Маълум бир белгилар борлигини текшириш.

3 Формага киритилган маълумотни тақдим этиш.

Java script да формалар. Формалар интернетда кенг ишлатилади. Формага киритилган маълумотлар тез – тез кайта серверга юборилади ёки бирор бир адресга электрон почта орқали жунатилади. Муаммо шундай, фойдаланувчи томонидан киритилган маълумот тугрилигига ишониш. Уни интернетга юборишдан олдин Java S тили ёрдамида осон текшириш мумкин. Олдин бу формани текшириш кандай булишини курсатмокчи эдик, кейин эса интернетга маълумот жунатишнинг кандай имкониятлари борлигини курсатамиз. Дастлаб биз оддий скрипт яратайлик. Айтайлик html-сахифа 2 та матн киритувчи элементга эга булсин. Улардан бирига фойдаланувчи уз исмини, 2 – сизга электрон почта манзилини киритсин. Сиз у ерга бирор бир маълумотни киритиб тугмани босиб куриш.

Агар сиз у ерга хеч нарса киритмаган булсангиз хатолик юз бериш хакида маълумот оласиз. 1 – элементдаги исталган маълумот тугрилик жихатдан текширилади. Албатта агар фойдаланувчи бошка номни киритган булса у буларни кафолатламайди. Браузер хатоларни сонларга хам каршилик курсатмайди. Масалан, сиз '17' деб киритсангиз у холда сиз 'Салом 17' деган саломни оласиз. Демак, бу текшириш идеал була олмайди.

Форманинг 2 чи элементи бир мунча мураккабироқдир. Унга оддий сатр киргизиб, масалан исмингизни. Сиз унга @ ни ёзмагунча бу элемент иш

бажармайди. @ белгини мавжудлиги билан фойдаланувчи электрон почта манзилини тугри киритган була олади. Гарчи маъносиз булсада биргина @ белгиниг узи хам бу шартга тугри келади.интернетдаги хар бир электрон почта манзилида @ белги катнашади. Шундай экан бу белгига текшириш бу ерда жуда уринли. Скрипт бу 2 та элемент билан кандай ишланди ва унда текшириш кай куринишда булади. Бу куйидагича олиб борилади:

```
<html>
<head>
<script language="JavaScript">
<!-- Скрыть

function test1(form) {
  if (form.text1.value == "")
    alert("Пожалуйста, введите строку!")
  else {
    alert("Hi "+form.text1.value+"! Форма заполнена
корректно!");
  }
}

function test2(form) {
  if (form.text2.value == "" ||
      form.text2.value.indexOf('@', 0) == -1)
    alert("Неверно введен адрес e-mail!");
  else alert("OK!");
}
// -->
</script>
</head>

<body>
<form name="first">
Введите Ваше имя:<br>
<input type="text" name="text1">
<input type="button" name="button1" value="Проверка"
onClick="test1(this.form)">
<p>
Введите Ваш адрес e-mail:<br>
<input type="text" name="text2">
<input type="button" name="button2" value="Проверка"
onClick="test2(this.form)">
</body>
</html>
```

Дастлаб </body> булимидаги кодни куриб чикамиз. Биз бу ерда факат 2 та тугмачани матн киритиш учун яратамиз. Бу тугмачалар test1(...) ва test2 (...) функцияларини кай бири босилганини караб чикаришда.

Функцияга аргумент сифатида this.form комбинациясини берамиз, чунки у кийинчилик бизга уша функцияга ва худди уша бизга керак элементга мурожат килишга рухсат беради. Test1 (form) функция берилган сатр бушлигини текширади. If (form.text1.value= = "").... оркали бажарилади. Бу ерда узгарувчи 'form' га функция чакирилганда 'this form' дан олинган киймат берилади. Биз элементга киритилаётган сатрни form text ига value сузини кушиб аниклаб олишимиз мумкин. Сатр буш эмаслигига ишониш учун уни " " билан таккослаб курамиз. Агар киритилган сатр " " билан мос келса у холда сатрга хеч нима киритилмаган булади ва булардан фойдаланувчи хато хакида хабар кабул килади. Агар бирор бир нарса киритилган булса фойдаланувчи "ок" тасдигини олади. Кейинги муаммо шундан иборатки фойдаланувчи форма майдонида пробел киритиш мумкин ва бу тугри маълумот сифатида кабул килинади! Агар сида бу пробелларни олиб ташлаш истаги булса у холда албатта унга бу имкониятни текширишни кушишингиз мумкин. Энди test2(form) функциясини курамиз. Бу ерда яна киритилган сатр бушлик билан текширилади. (укувчи томонидан киритилган маълумот хакикатдан бор йуклигига ишониш учун). Лекин биз if буйругига яна нимадир кушдик. // белгилар йигиндиси "ок" буйруги деб аталади. If буйркги биринчи ва иккинчи таккослаш нима билан тугалланишини текширади. Агар булардан биттаси бажарилса if буйруги true кийматига эришади ва кейинги скрипт бажарилади. Суз билан тушунтирганда келтирилган сатр ёки буш ёки унда @ белгиси йуклиги хакида хатолик деган хабар оласиз. (if буйругидаги 2 чи оператор киритилган сатр @ белгисига эга эканлигини текширади).

Маълум бир белгилар борлигини текшириш.

Айрим холларда формага киритилаётганлар факат айрим белгилар ёки сонлардан иборат булиши яъни маълумотни чегаралаш керак булиб қолади. Келтирилган маълумот факат сонлардан иборат булган телефон номерлари хакида эслашимиз кифоя. Бунда сизга келтирилган маълумотлар сонми йукми, шуни текшириш керак булади. Бу холатда мураккаблик шундан иборатки купчилик одамлар телефон номерларига яна хар хил белгилар, масалан, 01234-5678, 01234/56789 ёки 1234_56789 кушишлари мумкин. Фойдаланувчидан телефон номерларида бундай белгиларни фойдаланишини ман килиб булмаиди. Шу боис биз скриптимизни сонлар ва бир неча белгиларни текширувчи процедурани кушишимиз керак. Скриптнинг дастлабки коди куйидагича:

```
<html>
<head>
<script language="JavaScript">
<!-- hide
function check(input) {
    var ok = true;

    for (var i = 0; i < input.length; i++) {
```

```

    var chr = input.charAt(i);
    var found = false;
    for (var j = 1; j < check.length; j++) {
        if (chr == check[j]) found = true;
    }
    if (!found) ok = false;
}

return ok;
}

function test(input) {

    if (!check(input, "1", "2", "3", "4",
        "5", "6", "7", "8", "9", "0", "/", "-", " ")) {

        alert("Input not ok.");
    }
    else {
        alert("Input ok!");
    }
}

// -->
</script>
</head>

<body>
<form>
Telephone:
<input type="text" name="telephone" value="">
<input type="button" value="Check"
    onClick="test(this.form.telephone.value)">
</form>
</body>
</html>

```

Формага киритилган маълумотни такдим этиш.

Формага киритилган маълумотларни беришнинг кандай имкониятлари мавжуд? Форма маълумотларини беришнинг энг оддий усули бу электрон почта оркали узатишдир. (бу усулни кенгрок урганамиз). Агар сиз формага киритилаётган маълумотларга сервер куз-кулок булиб туришини хохласангиз у холда сиз CGI интерфейсидан фойдаланишингиз керак. У сизга маълумотларни автоматик равишда кайта номлашга имкон беради. Масалан, сервер мижозлардан айримлари кира олиш имконияти булган маълумотлар базасини яратиш мумкин.

Бошка мисол- уаhоо сингари кидириш саҳифалари. Одатда унда шахсий маълумотлар базасидан яратилган суровни кидириш учун форма ташкил килинган. Натижада фойдаланувчи мос тугмачани босишдан олдинрок жавобни олади. Унга токи жорий серверни куллаб-қувватловчи одамлар уларга курсатилган маълумотларни уқиш ва кидиришни кутиб утиришга ҳожат қолмайди. Буларнинг барчасини сервернинг узи бажаради. Java S бундай имкониятларни бажаришга йул қуймайди.

Java S маълумотларни сервердаги ыирор бир файлга ёзиб олиш имконияти булмагани сабабли у ёрдамида укув тақризлари китобини яратаолмайсиз. Буни фақат CGI интерфейси орқали бажариш мумкин.

Албатта, сиз фойдаланувчилар электрон почта орқали юборган маълумотлар учун тақриз китоби яратишингиз мумкин. Лекин бу ҳолда сиз электрон почта орқали олинган тақризларни қулда битта файлга киритишингиз керак. Агар сиз ҳар қуни 1000 тақриз олишни назарда тутмасангиз шундай қилишингиз мумкин. Шунга мос скрипт оддий HTML матндан иборат булади ва ҳеч қандай Java S дастурлаштириш керак эмас. Албатта битта ҳолни ҳисобга олмаганда, яъни агар сизга маълумотларни юборишдан олдин текшириш керак булса бу ерда ҳақиқатан ҳам Java S керак булади. Фақат биз ҳамма ерда ҳам ишламайдиган mailto буйругини киритамиз, у фақат Microsoft Explorer 3.0 да куллаб-қувватланмайди.

```
<form method=post action="mailto:your.address@goes.here"
enctype="text/plain">
Нравится ли Вам эта страница?
  <input name="choice" type="radio" value="1">Вовсе
нет.<br>
  <input name="choice" type="radio" value="2"
CHECKED>Напрасная трата времени.<br>
  <input name="choice" type="radio" value="3">Самый плохой
сайт в Сети.<br>
  <input name="submit" type="submit" value="Send">
</form>
```

Enctype="text/plane" параметри матнни худди узини ҳеч қодланган қисмларсиз юбориш учун ишлатилади. Бу сезиларли даражада бундай почтани уқишни соддалаштиради.

Агар сиз формани сетга узатилмасдан текширишни истасангиз, у ҳолда onSubmit ходисалар ишловчи дастурдан фойдаланишингиз мумкин. Сиз бу дастур учун < form > тегида чакирик жойлаштиришингиз мумкин. Масалан:

```
function validate() {
  // check if input ok
  // ...

  if (inputOK) return true
```

```

    else return false;
}

...

<form ... onSubmit="return validate()">

...

```

Агар шу тарика тузилган формада нотугри маълумотлар киритилган булса бу форма интернетга юборилмайди.

§ Форманинг аник элементини белгилаш.

Focus() усули ёрдамида формани янада самимий килиш мумкин. Демак, сиз кайси элемент биринчи булиб белгиланишини танлашингиз мумкин. Демак, сиз кайси элемент биринчи булиб белгиланишини танлашингиз мумкин. Ёки браузерга нотугри маълумотлар киритилган форматни белгилашни буюришингиз мумкин. Яъни браузернинг узи курсорни сиз курсатган формат элементига урнатади ва фойдаланувчига бирор нарса курсатишда форматни чиқаришга тугри келмайди. Буни куйидаги скрипт формати оркали килиш мумкин:

```

Function setfocus() {
    Document.first.text1.focus(); }

```

бу ёзув мен курсатиб утган критилувчи матннинг биринчи элементини бегилайди. Сиз барча формаланинг жисмларни курсатишингиз керак, жорий холда у first деб аталади ва форманинг элементи номи text1. Агар сиз саҳифа юкланганда жорий элемент белгиланишини истасангиз бунинг учун сиз <body> тэгини onload атрибути билан тулдиришингиз мумкин. Бу шундай курунишда булади:

```
<body onload = "setfocus()">
```

Яна бир мисол билан тулдиришингизга тугри келади:

```

Function setfocus() {
    Document.first.text1.focus();
    Document.first.text1.select(); }

```

Яна бир кдни курунг:

Бунда факат элемент белгиланмайди, балки ундаги матн ҳам белгиланади.

Назорат саволари

- 1 Форма нима ?
- 2 Java Script да бошқариш элементлар қайсилар ?
- 3 Форманинг асосий хоссалари ?
- 4 Форманинг асосий компоненталари ?

МАЪРУЗА №5

Усуллар ва функциялар тушунчаси .Обект ва хоссалар Java Script да документларнинг объект моделлари

Режа 1 Усуллар ва функциялар тушунчаси

2 HTML ХУЖЖАТЛАР ВА JAVASCRIPT ДАГИ ОБЪЕКТЛАР ИЕРАРХИЯСИ

Усуллар ва функциялар тушунчаси Ходисалар ва уларни ишлаб чиқарувчилар JavaScript тилида дастурлаштиришнинг энг зарур қисми ҳисобланади. Ходисалар фойдаланувчининг у ёки бу ҳаракатларидан ташкил топади. Агар у қайсидир тугмани босса, мурожаатини кесиб ўтса, “Mouse Over”

ходисаси юз беради. Ходисаларнинг бир неча турлари мавжуд. Биз JavaScript дастуримизда улардан бир нечасини бажаришимиз мумкин ва буни ходисаларни ишловчи махсус дастур ёрдамида амалга оширишимиз мумкин. Демак, тугмачани босиш оркали тушиб кетувчи ойна ходисасини яратиш мумкин. Бу шуни англатадики, Click – босиш ходисаси билан ойна яратиш реакцияси булиши керак.

Жорий вақтда биз ишлатишимиз керак булган дастур – ходисаларни ишлаб чиқарувчи OnClick деб аталади. Ва у компьютерга ходиса руй берганда нима килиш кераклигини хабар беради.

Берилаган мисол бир нечта янги хусусиятларга эга, уларни тартиб буйича қуриб чиқамиз. Биринчи янги хусусият бу OnClick=”alert(‘Y₀’)” булиб у <input> тегига тегишли. Айтиб утганимиздек, бу тугмани босганингизда нима содир булишини аниқлайди. Шу тариқа, агар Click ходисаси юз берадиган булса, компьютер alert(Y₀) чақиригини бажариши керак. Бунинг узи JavaScript тилидаги кодга мисол булади (Шунга эътибор қаратингки, бу ҳолатда биз ҳатто <script> тегидан фойдаланмадик).

Alert функцияси тушувчи дарчаларни ҳосил қилиш имконини беради. Уни чақирингда сиз кавслар ичига бирор сатр беришингиз керак. Бизнинг ҳолатда бу ‘Y₀’. Ва бу худди шу матн тушаётган дарчада пайдо булади. Шу тариқа, укувчи тугмачани шикирлатган пайтда, бизнинг скрипт ‘Y₀’ матнли дарча ҳосил қилади.

Берилган мисолда бир канча чалқашликлар тугилиши мумкин: биз document.write() буйруғида жуфт куштирнок, alert() конструкциясида эса бирлик куштирнок ишлатдик. Нега? Қупчилик ҳолларда 2 та ҳолдан ҳам фойдаланиш мумкин. Бирок, сунги мисолда биз OnClick=”alert (‘Y₀’)”да бирлик ва 2 лик куштирнок ишлатдик. Агар биз OnClick=”alert (“Y₀”)” деб ишлатганда компьютер будай скриптга тушунмаган булар эди. Ва қайси конструкция OnClick функциясига тегишли, қайсиси тегишли эмаслигини аниқлай олмасди. Шунинг учун бундай ҳолларда 2 та куштирнок турини ишлатишга мажбурсиз. Куштирнокни қандай тартибда ишлатишингиз муҳим рол уйнамайди, дастлаб иккилик куштирнок, сунгра бирилик куштирнок ва аксинча. Яъни сиз қуйидагича ёзишингиз мумкин: OnClick=’alert (“Y₀”)’.

ФУНКЦИЯЛАР.

JavaScript тилида тузилган қупчилик дастурларда функциялардан фойдаланамиз. Шунинг учун ҳозир бу тилнинг муҳим элементи ҳақида гапириш жуда муҳим. Функциялар қупчилик ҳолларда узида бир неча буйруқларни боғлаш усулини номоён қилади. Қелинг мисол учун бирор бир матнни уч марта бирданига чон қилувчи скрипт ёзамиз. Дастлаб оддий йулни қурайлик:

```
<html >  
< script language= “JavaScript”>  
< ! - - hide >  
document.Write( “Менинг саҳифамга хуш қелибсиз! < br >”);
```

```
document.Write( "Бу JavaScript! < br >");
document.Write( "Бу JavaScript! < br >");
document.Write( "Бу JavaScript! < br >");
< /Script >
< /html >
```

Ва бу скрипт кушимча матнни босиб чиқаради:

Менинг саҳифамга хуш келибсиз!

Бу JavaScript!

Бу JavaScript!

Бу JavaScript!

Агар дастлабки кодига қарайдиган бўлсак, керакли натижани олиш учун коднинг аниқ бир қисми 3 марта такрорланган. Ахир бу самаралими?

Ўйук биз ушбу масалани осонирок ечишимиз мумкин:

```
< html >
< script language= "JavaScript">
< ! - - hide
function myfunction () {
document. Write( `Менинг саҳифамга хуш келибсиз! < br >`);
document. Write( `Бу JavaScript! < br >`);
}
myfunction ();
myfunction ();
myfunction ();
< /Script >
< /html >
```

{ } кавс ичида жойлашган барча скриптлар myfunction() функциястга тегишли. Бу шуни англатадики document.write() дан иборат 2 та буйруқ бирлаштирилди ва улар функция чақирилиши билан биргаликда бажарилади. Ҳақиқатдан ҳам бизнинг мисолимизда функция 3 марта чақирилган ва буни функциянинг узи аниқлагандан сунг myfunction() сатрини 3 марта ёзганимизни куришимиз мумкин. Ва уз навбатида бу шуни билдирадики, функциянинг таркибидаги нарсалар (буйруқлар, { } кавс ичидаги курсатилганлар) 3 марта бажарилган. Бу функциянинг ишлатилишига етарли даражадаги оддий мисол ва сизда нима учун функция JavaScript учун шу даража муҳим деган савол тугилади. Ушбу кулланмани уқиш мабойнида сиз албатта функциянинг фойдасини билиб оласиз. Функцияни чиқаришда узгарувчиларнинг узатиш имкониятининг узи бизнинг скриптлар учун жуда ихчамлик яратади ва буни биз кейинроқ кураимиз.

Функциялар яна ходисаларни ишловчи процедуралар билан биргаликда ишлатилиши мумкин. Навбатдаги мисолни кураимиз:

```
< html >
< script language= "JavaScript">
< !-- hide
Function calculation()
Var x = 12;
Var y = 5;
```

```

Var result = x+y;
Alert (result);
// - -
</script>
</head>
<body>
<form>
<input type = “button” value = “calculate” onclick = “calculation()”>
</form>
</body>
</html>

```

Бу ерда тугмачани босиш билан calculation() функциясини чиқариш жараёни руй беради. Сизган булсангиз, бу функция x,y ва result узгарувчиларидан фойдаланиб, кушиш амалини бажаради. Узгарувчиларни var калит сузи билан эълон қилиш мумкин. Узгарувчилар турли катталикларни сон, сатр ва ҳақозаларни саклаш учун ишлатилади. Скриптнинг var result = x+y сатри браузерга result узгарувчисини ҳосил қилиш ва унга x+y арифметик амални жойлаштиришни (5+12) хабар беради. Бундан сунг result узгарувчиси 17 қийматни олади. Бу ҳолда alert (result) ва alert(17) буйруғи бир хил вазифани бажаради. Бошқача қилиб айтганда, бу 17 сон ёзилган дарчага эга буламиз.

HTML ХУЖЖАТЛАР ВА JAVASCRIPT ДАГИ ОБЪЕКТЛАР ИЕРАРХИЯСИ

Web саҳифанинг барча элементлари JavaScript тилидаги иерархик тузилиши буйича қурилади. Хар бир элемент объект қуринишида номаён булади. Ва бундан хар бир объект аниқ таркибга ва усулга эга булади. Уз навбатида JavaScript тили Web саҳифа объектларини осон бошқариш имкониятини беради. Ва бунда сиз объектлар иерархиясини тушунишингиз жуда муҳим. Буларнинг қандай руй беришини навбатдаги мисол орқали тез тушунишингиз мумкин. Оддий HTML саҳифани қурамиз:

```

<html>
<head>
<title My homepage >
</head>
<body bgcolor = # ffffff>
<center>
<img scr = home.gif name=“pic1” width = 200 height = 100>
</center>
<p>
<form name = “My form”>
Name:
<input type = “text” name = “home” value = “ ”> <br>
e – mail:
<input type=“button” value=“pushme” name= “mybutton” onclick = “alert(Y)”>

```

```

</form>
<p>
<center>
<img scr = "my.gif" name = "pic4" width = 300 height = 15>
<p>
<a href = http://maks/index.html> </a>
</center>
</body>
</html>

```

Демак биз 2 та тасвир, 1 та мурожат ва кандайдир матн чиқариш учун 2 майдондан иборат форма ва 1 та тугмачага эгамиз. JavaScript нуктаий назаридан браузер дарчаси – аллакандай Window объекти булиб хисобланади. Бу объект уз навбатида бир неча расмийлаштириш элементларига эга. Дарча ичкарасига биз html документини урнатишимиз урнатишимиз мумкин (ёки бошка тилдаги файл, хозирча html файли билан чегараланиб турамыз). Бундай саҳифа document объектидан сира ҳам фарк қилмайди. Бу шуни англатадики document объекти JavaScript тилида хозирги пайтда юкланган html документни намоеън этади. Document объекти JavaScript тилида жуда муҳим объектлардан бири хисобланади ва сиз ундан куп марта фойдаланасиз. Масалан Document объектининг таркибига , Web саҳифанинг фонинг ранги қиради. Шундай қилиб биз учун шуниси муҳимки html объектларидан барчаси document объектининг таркибига қиради. Html объектларига мисол, мурожаат ёки тулдирилган форма була олади.

Куйидаги расмда мисолимиз орқали яратилган html саҳифанинг объектлари иерархияси тасвирланган:

Албатта биз бу иерархия ёрдамида турли объектлар хақида маълумотга эга булишимиз ва уни бошқариш имконига эга булишимиз керак. Бунинг учун биз JavaScript тилида турли объектларга рухсат кандай ташкил этилганлигини билишимиз шарт. Қуриниб турибдики иерархик тузилишга кура хар бир объект уз номига эга. Мос равишда агар сиз html саҳифадаги 1-расмга мурожат қилмоқчи булсангиз объектлар иерархиясидаги уз йуналишингизни танлаб олишингиз керак ва энг юкори чуқкидан бошлаш керак. Бу тузилишдаги 1-объект document деб аталади. Саҳифадаги 1-расм images [0] объекти каби тасвирланган. Бу шуни англатадики JavaScript да document.images[0] деб ёзиб бу объектга рухсат олган буламиз. Масалан агарда сиз форманинг 1-элементига кандай матн киритилганини билмоқчи булсангиз датслаб кандай қилиб объектга рухсат олишни аниқлаб олишингиз керак ва яна объектлар иерархиясининг энг юкорисидан бошлаймиз. Сунгра elements[0] объектга йул курсатамыз ва кетма–кетлик билан барча объектларни номини ёзамиз ва натижада шу нарса аниқланадики 1-майдонга рухсатни шуларни ёзиш орқали олиш мумкин:

Document.forms[0].Elements[0]. Энди фойдаланувчи томонидан киритилган матнни кандай килиб билиш мумкин. Объектнинг кайси тури ёки хоссаларидан бири бу информацияга рухсат бериши мумкинлигини аниклаш учун JavaScript хакидаги маълумотномаларга муржат килиш керак. У ерда сиз киритилган матн майдонига мос келувчи элемент value хоссаси борлиги ва у киритилган матнга мос келади. Демак энди биз киритилган кийматни укиш учун барча керакли нарсаларга эгамиз. Бунинг учун JavaScript сатрида куйидагини ёзиш мумкин.

Name = document.froms[0].elements[0].value

Олинган сатр **name** узгарувчисига юкланади. Энди биз бу узгарувчидан хохлаган керакли пайтда фойдаланишимиз мумкин. Масалан, биз alert (“Hi”+name) буйруги ёзилган дарча яратишимиз мумкин. Натижада, агар фойдаланувчи бу майдонга ‘Stefan’ сузини киритган булса alert (“Hi”+name) буйругига кура янги дарчада ‘Hi Stefan’ салони пайдо булади. Агар сиз катта саҳифалар билан иш курсангиз турли объектларга номлар буйича адреслаш процедурасида жуда чалкашлик булиши мумкин. Масалан document.forms[3].elements[17] document.forms[2].elements[18] объектларига муружаат килиш керак булсин? Бундай муоммадан кутилиш учун турли объектларга узингиз ном беришингиз мумкин. Буларни кандай килишни мисол оркали куришингиз мумкин:

```
<form name = “myForm”
```

```
  Name
```

```
    < input type = “text” name = “kirit “ value = “ “ ><br>
```

бу ёзув шуни билдирадики, forms[0] объекти энди 2-чи исм myForm га эга булди. Худди шундай elements[0] урнига kirit ни ёзишингиз мумкин (<input> тегида охирги атрибути name да курсатилган). Шу тарика

```
  name = document.forms[0].elements[0].value урнига
```

```
  name = document.myForm.kirit.value ни ёзишингиз мумкин. Бу
```

сезиларли даражада JavaScript да дастурлашни соддалаштиради, айникса катта Web-саҳифалар куп микдорда объектларга эга булса. (Номларни ёзишда регистрларни, ката-кичик харфлар холатини эътиборга олиш керак, яъни myForm урнига myform ёзиш мумкин эмас) JavaScript да объектларнинг купчилик хусусиятларида рухсат факат укиш учун эмас. Яна сиз уларга янги кийматлар ёзиш имкониятига хам эгасиз.

JavaScript воситалари оркали курсатилган майдонга янги сатр ёзишингиз мумкин.

```
<form name = “myForm”
```

```
<input type = “text” name = “input” value = “ха ха ха”
```

```
<input type = “button” value = “write”
```

```
  onclick = “ document.myForm.input.value = ‘уф! ;’”
```

Дастлаб куйидаги мисолни куриб чикамиз:

Скриптни даслабки коди:

```
< html >
```

```
< head >
```

```
< title>Objects < /title >
```

```

    < script language = “JavaScript”>
    < !- -hide
    function first() {
// форма киритилган
// матн майдонида дарча хосил килади
alert (“Текст элементининг киймати :”+ document.myForm.myText.value) ;
}
    function second() {
// Жорий функция кучирувчилар кийматини аниклайди.
Var mystring = “ The checkbox is ” ;
// Кучирувчи учганми ёки йук?
    If (document.myForm.mycheckbox.checked.) mystring += “checked”
    else mystring += “notchecked”;
// Экранга сатрни чикариш
    alert ( mystring ) ;
}
// - -
< /script >
< /head >
<body bgcolor = lightblue >
    < form name = “myForm >
    < input type = “text” name = “myText” value = “ха ха ха” >
    < input type = “button” name=“button1” value = “button1” >
        onclick = “first ()”
    <br>
    < input type = “checkbox” name = “mycheckbox”>
    < input type = “button” name=“button 2” value = “button2” >
        onclick = “second()”
    < /form >
    < p > < br > < br >
< script language = “ JavaScript ”>
    < !- - hide
    document.write( “Фон рангги бу : ”) ;
    document.write( document.bgcolor+ “<br>”);
    document.write( “ 2 чи button даги матн :”);
    document.write( “ document.myForm.button 2.value”);
// --
< /script >
< /body >
< /html >

```

Location объекти

JavaScript да window ва document объектларидан ташкари location деб аталувчи яна бир муҳим объект мавжуд. Бу объектда HTML документ юкланган адрес

сакланади. Агар сиз `http://www.xyz.com/page/html` саҳифасини юклаган булсангиз, у холда `location.href` нинг киймати ҳам худди шу адресга мос тушади. Шундай булсада биз учун жуда муҳим нарса шуки, сиз `location.href` га янги киймат ёзишингиз мумкин. Масалан, куйидаги мисолда тугмача жорий дарчадан янги саҳифага утишни юклайди :

```
< form >  
< input type = button value="Yahoo">  
    onclick = "location.href = ` http://www.Yahoo.com ` ;"  
< /form >
```

Назорат саволлари

- 1 Ходиса нима ?
- 2 Функция нима ?
- 3 Объект нинг асосий хоссалари ?
- 4 **Location объекти** нинг асосий имконияти ?

МАЪРУЗА №6

Java Script да ходисни кайта ишлаш .Интерактив формалар

РЕЖА 1 Java Script да ходисни кайта ишлаш

2 Интерактив формалар

Java Script да ходисни кайта ишлаш

Ходисалар ва уларни ишлаб чиқарувчилар JavaScript тилида дастурлаштиришининг энг зарур қисми ҳисобланади. Ходисалар фойдаланувчининг у ёки бу ҳаракатларидан ташкил топади. Агар у кайсидир

тугмани босса, мурожаатини кесиб утса, “Mouse Over” ходисаси юз беради. Ходисаларнинг бир неча турлари мавжуд. Биз JavaScript дастуримизда улардан бир нечасини бажаришимиз мумкин ва буни ходисаларни ишловчи махсус дастур ёрдамида амалга оширишимиз мумкин. Демак, тугмачани босиш оркали тушиб кетувчи ойна ходисасини яратиш мумкин. Бу шуни англатадики, Click – босиш ходисаси билан ойна яратиш реакцияси булиши керак.

Жорий вақтда биз ишлатишимиз керак булган дастур – ходисаларни ишлаб чиқарувчи OnClick деб аталади. Ва у компьютерга ходиса руй берганда нима килиш кераклигини хабар беради.

Берилаган мисол бир нечта янги хусусиятларга эга, уларни тартиб буйича куриб чиқамиз. Биринчи янги хусусият бу OnClick=”alert(‘Y0’)” булиб у <input> тегига тегишли. Айтиб утганимиздек, бу тугмани босганингизда нима содир булишини аниқлайди. Шу тарика, агар Click ходисаси юз берадиган булса, компьютер alert(Y0) чакиригини бажариши керак. Бунинг узи JavaScript тилидаги кодга мисол булади (Шунга эътибор қаратингки, бу ҳолатда биз ҳатто <script> тегидан фойдаланмадик).

Alert функцияси тушувчи дарчаларни ҳосил килиш имконини беради. Уни чакиришда сиз кавслар ичига бирор сатр беришингиз керак. Бизнинг ҳолатда бу ‘Y0’. Ва бу худди шу матн тушаётган дарчада пайдо булади. Шу тарика, укувчи тугмачани шикирлатган пайтда, бизнинг скрипт ‘Y0’ матнли дарча ҳосил қилади.

Берилган мисолда бир канча чалқашликлар тугилиши мумкин: биз document.write() буйругида жуфт куштирнок, alert() конструкциясида эса бирлик куштирнок ишлатдик. Нега? Қўпчилик ҳолларда 2 та ҳолдан ҳам фойдаланиш мумкин. Бирок, сунги мисолда биз OnClick=”alert (‘Y0’)”да бирлик ва 2 лик куштирнок ишлатдик. Агар биз OnClick=”alert (“Y0”)” деб ишлатганда компьютер будай скриптга тушунмаган булар эди. Ва қайси конструкция OnClick функциясига тегишли, қайсиси тегишли эмаслигини аниқлай олмасди. Шунинг учун бундай ҳолларда 2 та куштирнок турини ишлатишга мажбурсиз. Куштирнокни қандай тартибда ишлатишингиз муҳим рол уйнамайди, дастлаб иккилик куштирнок, сунгра бирилик куштирнок ва аксинча. Яъни сиз қуйидагича ёзишингиз мумкин: OnClick=’alert (“Y0”)’.

Янги ходисалар

Java 1.2 қуйидаги ходисаларни қайта ишлашни қўллаб қуватлайди:

Abort	Dblclick	Keydown	Mousedown	Mouseup	Select
Blur	DragDrob	KeyPress	Mousemove	Move	Submit
Click	Error	KeyUp	Mouseout	Reset	Unload
Change	Fox	Load	Mouseover	Resize	

Жадвални урганиш давомида қўришимиз мумкинки бу янги ходисаларнинг бир нечаси руёбга чиқарилган. Биз бу қисмда уларнинг бир нечасини қўриб чиқамиз. Дастлаб resize ходисасини қўрайлик. Ушбу ходиса ёрдамида дарча улчами укувчи томонидан узгартирилганини аниқлашингиз мумкин. Қуйидаги скрипт буни қандай қилишни қўрсатади:

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

```

<script language = “ JavaScript ” >
    window.onresize = message;
    function message ( ) {
        alert (“дарча улчами узгартирилди”);
    }
</script>
</head>
< body>
    Илтимос бу дарча учамини узгартиринг
</body>
</html>

```

window.onresize = message Сатрида биз уша ходиса процедурасини беремиз. Аниқрок килиб айтганда фойдаланувчи дарча улчамини узгартирганидан дарров ушбу message () функциеси чакирилади. Эхтимол сиз ходисаларни кайта ишловчи программаларни бундай тарзда бериш билан топип билмасангиз керак. Бирок бу ерда хеч кандай янгилик йук. Масалан, агар сизда button объекти бор булса, бу ходисани кайта ишлаш процедурасини куйидаги тарзда аниклаш мумкин.

```

<form name = “my form” >
<input type = “button” name = “my button”onclick = “alert (‘дарча улчами
узгартирилди’ )”>
</form>

```

Бирок сиз буни яна бир бошка курунишда беришингиз мумкин:

```

<form name = “my form” >
<input type = “button” name = “my button”>
</form>

```

```

-----
<script language = “JavaScript”>
document.myform.mybutton.onclick = message;
function pressage ( ) {
alert+ (‘дарча улчами узгартирилди!’);
}
</script>

```

Уйлаш мумкинки 2 чи вариант бироз мураккаброк. Лекин нима учун биз айнан ундан 1 чи скриптда фойдаландик? Сабаби шундаки биз window объектини бирор бир аниқ тег ёрдамида аниқлай олмаймиз ва шунинг учун биз 2 чи вариантни куллашимизга тугри келади. 2 та мухим эслатма: 1 – дан window.onResize деб ёзмаслигингиз керак балки барчасини катта харфлар билан ёзиш керак. Агар сиз window.onResize = message () деб ёзсангиз браузер message() ни функция чикарувчи деб кабул килади. Лекин бизнинг холатда биз функцияни тугридан – тугри чикаришни хохлаймиз, факат ходисаларни кайта ишловчини аниклаб куймокчимиз.

Event объекти.

Java S1.2 тилига янги event объекти кушилган. У узида бирор ходисани ёзиб борувчи хусусиятга эга. Хар сафар бирор ходиса юз берганда event объекти мос кайта ишловчи дастурга юборилади.

Навбатдаги мисолда экранга бирор бир тасвир чикарилади. Сиз унинг каергадир сичконча курсаткичи билан туртишингиз мумкин. Натижада экранда сичконча жорий вақтда турган нуктасининг координаталари курсатилган хабар чикади.

```
<layer>
```

```
<a href = “ # ” onclick = “alert(‘x:’ +event.x+’y:’+event y); return false;”>
```

```
<img src = davinci.jpg “width = 209 height = 264 border = 0”>
```

```
</a>
```

```
</layer>
```

куриб турганингиздай JavaS нинг олдинги версияларида бажарганимиздагидек <a> тэгда onclick ходисани кайта ишловчи программани жойлаштирдик. Бу ерда янги булиб event.x ва event.y лар билан яратилган хабар дарчаси хисобланади. Худди шунинг узи бизга сичконча координаталарни аниқлаб берувчи бизга керак булган event объектида. Бундан ташкари мен барча буйруқларни <layer> тэгига жойлаштирдим. Шу туфайли биз жорий каватга нисбатан яъни бизнинг холатда тасвирнинг узига нисбатан координаталар хакидаги хабарни кобул киламиз. Акс холда биз браузер дарчасига нисбатан координаталар олган булар эдик. (return false инструкцияси бу ерда браузер жорий мурожатни узокрок кайта ишлаш учун фойдаланилган).

Event объекти куйидаги хусусиятларга эга:

Data – DragDrop ходисаси юз берганда тухтатилган объектларнинг URL адреслар массиви.

Layer x Каватга нисбатан курсаткичнинг горизонтал холати (нукталарда) resize ходисаси билан биргаликда бу хусусият браузер дарчаси кенглигини билдиради.

Layer y Каватга нисбатан курсаткичнинг вертикал холати (нукталарда) resize ходисаси билан биргаликда бу хусусият браузер дарчаси баландлигини билдиради.

Modifiers Модификатор калитини берувчи сатр – alt_mask, control_mask, meta_mask ёки shift_mask .

Page x Браузер дарчасига нисбатан курсаткичнинг горизонтал холати (нукталарда)

Page y Браузер дарчасига нисбатан курсаткичнинг вертикал холати (нукталарда)

Screen x Экранга нисбатан курсаткичнинг горизонтал холати. (нукталарда)

Screen y Экранга нисбатан курсаткичнинг вертикал холати. (нукталарда)

Target Дастлаб ходиса юборилган объектни намоён килувчи сатр.

Type Ходиса тилини курсатувчи сатр.

Which Сичконча ёки тугмачанинг ASCII киймати.

X layer x синоними

Y layer y синоними

§ Ходисани камраб олиш.

Java S тилининг асосий хусусиятларидан бири ходисани камраб олишдир.

Масалан, агар кимдир тугмачани туртса у холда шу тугмачага мос onclick ходисасини ишловчи дастур чакирилади. Сиз ходисаларни ишлаш ёрдамида дарча, документ ёки каватга мос курсатилган тугмача объектнинг ходисасини ишловчи чакирилишига кадар ходисаларни ишлаш ва камраб олишга эришасиз. Худди шундай сизнинг дарчангиз, документингиз ёки каватингиз объекти узининг оддий адресини топгунга кадар ходиса хакида сигнални ишлаш мумкин. Бу нимага кераклигини куйидаги мисол ёрдамида курайик:

```
Html
Head
Script Language = "JavaScript"
Window.Capture Events (Event.click);
Window.onclick = haudle;
Function.haudle (L) {
    Alert ("window объекти бк ходисани камраб олади!");
    Return true;
}
</script>
</head>
<body>
<a href = "test.htm"> "Кликните" по этой ссылке; </a>
</body>
</html>
```

Куриб турганингиздай бу <a> тегда ходисани ишловчи дастурни курсатмаймиз. Бунинг урнига

```
Window.Capture Events (Event.click);
```

Сатрни window объектини click ходисаси камраб олиши учун ёзилади. Одатда window объекти click ходисаси билан ишламайди. Бирок караб олиб у кайтадан уни window объектуга йуллайди.

Шуни эътиборга олиш керакки Event.click даги click фрагменти бош харфлар билан ёзилиши керак. Агар сиз бир неча ходисаларни камрашини истасангиз у холда сизга уларни бир биридан / белгилари оркали ажратишингиз керак булади. Масалан:

```
Window.Capture Events (Event.click/Event.move);
```

Haudle () функциясида ходисани ишлаш дастури сифатида return true дан фойдаланамиз. Бу хакикатдан шуни англатадики, haudle () функцияси тамом булгандан сунг браузер мурожатнинг узини ишлаб чикиш керак. Агар сиз унинг урнинга return false ёзсангиз шу ернинг узида барчаси тамом булади. Агар энди сиз <a> теги onclick ходисасини ишловчи дастур берсангиз, у холда ушбу ходисани руй беришида жорий дастур чикарилмаслигини тушуниб етасиз. Хайрон коларли жойи йукки, window объекти link объектига эга булгунга кадар, у ходиса хакида сигнални кулга киритади. Албатта агар сиз haudle () функцисини куйидагича аниклаймиз:

```
Function haudle (l) {
    Alert ("window объекти ушбу ходисани кулга киритади");
    Window.route Event (l);
```

```
Return true;
}
```

Return узгарувчиси Event объекти хисобланади.

Бундан ташқари сиз ихтиёрий объектга бевосита сигнал юборишингиз мумкин. Бунинг учун сиз handle Event () усулидан фойдаланишимиз мумкин. Бу куйидаги куринишда бўлади. Бу куйидагича куринишда бўлади:

```
<html>
<Script Language = JavaScript>
window.addEventListener (Event.click);
window.onclick = handle;
function handle (e) {
document.getElementById (e);
}
</script > <a href = "test.htm"> "Кликните" по этой ссылке </a> <br>
<a href = "test.htm">
onclick = "a href ('2 чи муружат учун ходиса ишловчи!');">
2 чи муружат </a>
</html>
```

Агар сиз ҳеч бир муружатни тўртмаган бўлсангиз ҳам click ходисасидаги сигналлар 2 чи муружатни ишлаш учун юборилади.

Навбатдаги скрипт тугмача босилишидаги сигналларни сизнинг скриптингиз қандай ишлаганини намоён қилади:

```
<html>
<Script Language = JavaScript>
window.addEventListener (Event.keypress);
window.addEventListener = pressed;
function pressed (e) {
alert ("key pressed! ASCII – value:" + e.which)
}
</script> </html>
```

Назорат саволлари

- 1 Ходисаларни қайси ишлаш функциялари ?
- 2 Event объектининг асосий вазифаси ?
- 3 Ходисани қанча олиш деганда нимани тушунасан ?

МАЪРУЗА №8 PHP .Сервер томонидан дастурлаш .PHP га кириш ўрнатиш ва тестлаш

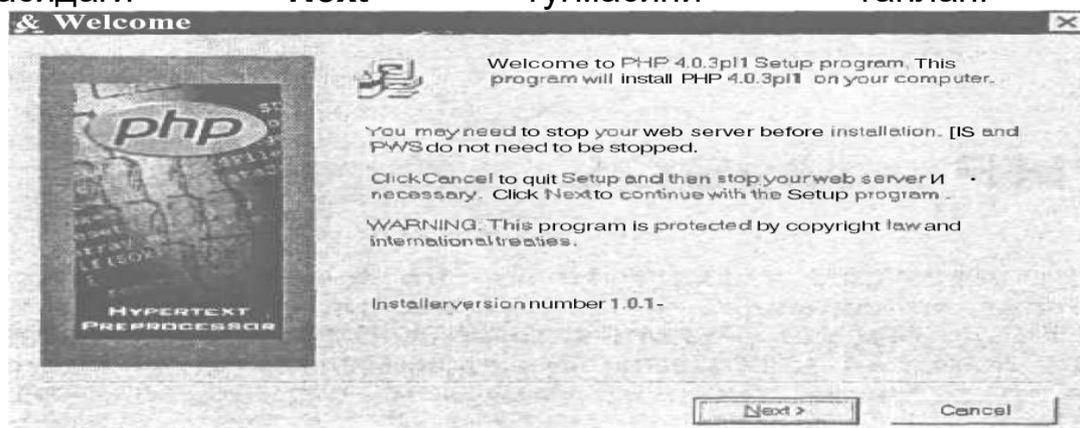
- 1 PHP .Сервер томонидан дастурлаш
- 2 PHP га кириш
- 3 PHP ни ўрнатиш ва тестлаш

Бу бўлимда PHP тилини 4 версиясини ўрнатишни ўрганамиз. Афсуски ушбу китоб ёзилиш даврида PHP барча зарур модуллари билан худди Apache каби битта амалда ўрнатиб берувчи нормал setup-программаси йўқ. Шунинг учун уни ўрнатиш сизга биров мураккаб бўлиб туюлиши мумкин.

Авалло PHP ни расмий <http://www.php.net> адресли сайтидаги **Downloads** секциясидаги иккита: биринчиси zip, иккинчиси эса – ехе кенгайтмали файлни юклаб олинг. Бу файлларга кўрсатгич сайт схфасини юқорисида, **Win32 Binaries** сарлавхасдан кейин жойлашган. Биринчи файл PHP 4 ни тўла версияни ташкил этади, бироқ қулай ўрнатиб берувчи программага эга эмас. Иккинчиси эса аксинча, автоматик ўрнатилувчи программа, лекин уни таркибида кўп қўлланиладиган модуллари йўқ.

Шунингдек сизга PHP бўйича документацияни ҳам нухсалаб олинг. Унинг кўрсатгичи сайт саҳифасида биров пастрокда жойлашган. Уйлайманки бу сизга PHP тилини ўзганишда жуда асқотади. **выручит.**

Бу ерда PHP тили версияси хақида фикр билдириш хам ўринли. Бу тил доимий равишда такомиллашмоқда ва бу китоб яратилиш вақтида унинг охириги версияси 4.0.3. эди. Вақт ўтиши билан унинг кейинги, масалан, 4.0.10 версияси яратилиши мумкин. Ўйлашимча унинг энг янги версиясини танлаган тўғри. Чунки унда хатоликлар тузатилган бўлади. Асосийси унда 4 рақами мавжуд бўлсин, чунки 4 – версия 3 – версияга қараганда кўпроқ функцияни қўллаб – қувватлайди. **PHP ни ўрнатиш** Сайтдан олинган ехе – файлни ишга туширинг. Хосил бўлган мулоқат ойнасидаги **Next** тугмасини танланг (расм



5.1).

Расм. 5.1. PHP ни ўрнатиш

1. Лицензия шартларига розилик билдириш учун I Agree тугмасини босинг. Хосил бўлган мулоқат ойнасидан ўрнатишни Standard типини танланг.
2. Энди PHP ўрнатиладиган директорияни кўрсатинг. Жимликка кўра c:\PHP каталоги таклиф этилади. Бироқ C:\program Files\PHP4 каталоги танланса тўғрироқ бўлади, яъни Apache га “яқинроқ” жойлашади (расм 5.2). бу каталогни кўрсатиш учун **Browse...** тугмасини босинг ва каталог номини киритинг. Навбатдаги мулоқат ойнасига ўтиш

учун одатдагидек Ок тугмасини, кейин Next тугмасини танланг.

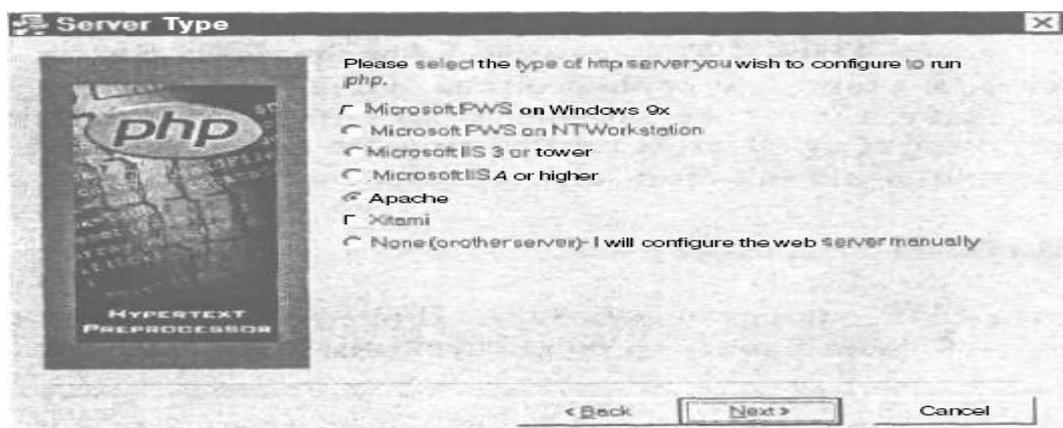


Расм 5.2. PHP ни ўрнатиш каталогини танлаш

- Ўзингизни SMTP-сервер (Send Mail Transfer Protocol — почта хабарларини алмашиш протоколи) адресинигизни ҳамда электрон почтангиз адресини кўрсатинг. Айнан мана шу сервер ва тескари адрес PHP тилини Mail функцияси чақирилганда чиқувчи почта сўрови учун қўлланилади. Умуман олганда, бу сервер орқали сиздаги одатдаги почта клиенти – масалан, Outlook Express почтани жўнатади – бунда Mail() функцияси локал машинада ишлмайди.
- PHP мосланадиган серверни танланг. Бизнинг холатда бу **Apache**. Расм 5.3.

Файлларни нусхалаш жараёни бошланади. Бу жараён тугаганидан кейин яна бир неча мулоқат ойналари пайдо бўлади. уларга эътибор қилманг.

Шу билан PHP тилини ўрнатилди деб ҳисоблаш мумкин. Фақатгина PHP-сценарийларни таниб олиши учун Apache ни ростлаш, ҳамда биз интернетдан олган zip-архивдаги қўшимча модулларни улаш иши қолди.



Расм. 5.3. Серверни танлаш

Apache ни PHP билан ишлашини таъминлаш

1. Блокнотда Apache ни C:\Program Files\Apache Group\Apache\conf каталогда жойлашган httpd.conf конфигурация файлини очинг. Хусусан, бу файлини **Пуск | Программы | Apache Web Server | Management** менюсидаги **Edit configuration** пунктини танлаб очиш мумкин.
2. Файлдаги текстдан қуйидаги сатрни топинг:

```
#AddType application/x-httpd-php php
```
3. Сатр олдидаги изоҳ белгисини ўчиринг:

```
AddType application/x-httpd-php php
```

Бу билан биз барча php кенгайтмали файлларга application/x-httpd-php типини бердик.
4. Юқоридаги сатрдан кейин қуйидаги сатрларни киритинг:

```
ScriptAlias /_php/ "C: /Program Files/PHP4/"
Action application/x-httpd-php "/_php/php.exe"
```

Бу билан, биринчидан, PHP процессори жойлашган каталогга Apache мурожат этиши учун бу каталогни **_php** номли синоними яратилади, иккинчидан барча application/x-httpd-php типли файллар php.exe дастури билан боғланади.
5. Конфигурация файлига киритилган ўзгаришларни мақланг. Агар Apache ишлаётган бўлса уни ишини тўхтатинг (**Пуск | Программы | Apache Web Server | Management | Stop Apache** пункти) ва қайта ишга туширинг. Агар Apache ишга тушмаса (уни ойнаси очилиб дархол ёпилади), демак синтактик хатога йўл қўйилган. Бу хатоликларни бартараф этиш учун 4 бобда келитирлган кўрсатмалардан фойдаланинг

PHP ни тестлаш

Энди PHP – сценарий ишлаётганлигини текшираамиз. Бунинг учун z: /home/localhost/www каталогда test. php файлини яратамиз ва қуйидаги текстни киритамиз:

Листинг 5.1. Тестлаш учун сценарий

```
echo "It works ! <br>\n";
phpinfo ( ) ;
?>
```

Энди браузерни адрес майдонига қуйидагини киритинг: <http://localhost/test.php>. Натижада phpinfo() функцияси хосил қилган PHP ҳақидаги турли маълумотлар хосил бўлади.

Эслатма

PHP-сценарий— бу CGI-сценарий билан бир хил нарса эмасдир. Хусусан, агар CGI-сценарий одатда /cgi-bin/ ёки /cgi/ каталогда жойлашса, у холда php-сценарий документлар жойлашган каталогда жойлашиши керак.

Агар браузерда саҳифа тасвирланмаса, демак, httpd.conf файлида хатоликка йўл қўйилган. Бу файлни очиб хатони тўғирланг. Кейин Apache ни қайта ишга туширишни унутманг.

Қўшимча модулларни ўрнатиш

PHP ни ишлашга яроқлилигига ишонч хочил қилгандан кейин унга zip - файлдаги қўшимча модулларни киритиш мумкин. Бу модуллар — расмлар билан, календар, FTP (File Transfer Protocol — Файллар алмашиш протоколи) ва х.к. билан ишлаш воситаларидир. сезган бўлсангиз архивда фақат модуллар эмас, PHP ни тўла версияси мавжуд. Модулларни ўрнатиш учун қуйидаги амалларни бажаринг.

1. zip-архивни PHP ўрнатилган каталогга очинг (бизнинг мисолда бу c:\Program Files\PHP4 каталоги). Баъзи файллар қайта очилади, баъзилари қўшилади. Хусусан, extensions каталоги хосил бўлади. Бу каталогда PHP учун зарур барча файллар жойлашади.
2. Энди PHP га қайси модуллардан фойдаланишини кўрсатиш ва баъзи ўрнатмаларни кўрсатиш зарур. Бунинг учун **Блокнот** ёрдамида Windows файллари каталогидаги (одатда C:\Windows) php.ini файлини очинг. Бу файл PHP ўрнатувчи программа томонидан яратилади. Файлда бир неча сатр мавжуд бўлиб уларни хар бири битта параметрни қийматига мос келади. Сатрни бир қисми -, символидан кейин жойлашган бўлиб, бу қисми изох сифатида қаралади.
3. **magic_quotes_gpc** параметрини топинг ва уни бекор қилинг:

```
magic_quotes_gpc=0ff
```

Бу билан PHP га формалардан келган баъзи символлар олдида тескари слэш символини қўйиш тақиқланади. Ушбу ва бошқа параметрлар хақида кейинчалик батафсил маълумотлар келтирилади.

4. Навбатдаги параметрни топинг ва қуйидагича ўзгартиринг:

```
extension_dir=C:\Program Files\PHP4\extensions
```

Бу билан PHP га модулларни C:\Program Files\php4\extensions каталогидан излаши кераклиги уқтирилади. Эътибор қилган бўлсангиз бу параметрда жимликка кўра. / қиймати ўрнатилган, яъни излаш PHP ўрнатилган каталогда амалга оширилади. Бу албатта, ноқулай.

5. ;extension= ёзуви билан бошланувчи "изохли" сатрларни топинг. Улар ичидан бизга керакли модулларга мос сатрлардаги изох символини ўчириш зарур. Ушбу китобда тасвирлар билан ишлаш учун GD библиотекаси тавсиланади. Шунинг учун бизга php_gd.dll библиотекаси зарур бўлади. MySQL ва календар функцияларини қўллаб қувватлаш функциялар PHP ни ўзида мужассамлашган.
6. php.ini файлидаги ўзгартишларни сақлашни унутманг. Ўзгартиришлар кучга кириши учун Apache ни қайта ишга тушириш зарур эмас, чунки биз PHP сервер модули эмас, балки алоҳида программа шаклида ўрнатганмиз

Назорат саволлари

- 1 PHP нима ?
- 2 PHP дастурлаш тилининг имкониятлари ?
- 3 PHP компьютерга ўрнатишдаги асосий кўрсаткичлар ?
- 4 PHP тестлаш бубруклари ?

МАЪРУЗА №9

Виртуал серверга эhtiёж Apache ни ўрнатиш

- Режа
- 1 Виртуал серверга эhtiёж
 - 2 Apache ни ўрнатиш
 - 3 Apache ни тестлаш

Китобни бу қисми сизга энг яхши серверлардан бирини Интернетдан олишга ва ўрнатишда ёрдам беради. Бу — Apache серверидир. Шунингдек баъзи иловалар ҳам тавсия этилади. Булар PHP ва машхур MySQL МББС интерпретаторидир. Китобни бу қисми билан танишгандан ва интернетдан дистрибутивларни олгандан (хаммаси бепул) кейин Web да профессионал даражада ишлаш учун зарур барча инструментларга эга бўласиз

Агар сиз хатто келажакда PHP дан фойдаланмасдан бошқа тилдан (масалан Perl) фойдаланишни маъқул кўрсангиз ҳам китобни бу қисми билан танишиш сценарийларни яратиш ва уларни отлатка қилишда катта ёрдам беради.

Китобни бу қисми айтиб ўтилганидек, фақат PHP да дастурловчилар учунгина фойдали эмас. Чунки баъзи холларда HTML – саҳифани тўла кўринишини текшириш зарур бўлади. Бироқ уйда ишлашда буни имкони йўқ — SSI технологияси (Server-Side Includes — Сервер томонда активлаш), CGI (Common Gateway Interface — Умумий шлюзовой интерфейс) ва, албатта, PHP сервердан фойдаланишни талаб этади. Нима қилиш керак? Бунинг учун ўзингизни компьютерга (хатто Интернетга уланмаган бўлса ҳам) махсус дастурни – Web – серверни ўрнатиш. Умуман серверлар жуда кўп — ёмон ва яхши, секин ва тез... Мен эса сизга Apache серверини ўрнатишни тавсия этаман. Бу сервер Windows муҳитида PHP, CGI технологиялари ва Perl – сценарийлар билан худди Unix даги каби ишловчи ягона сервер дейиш мумкин.

Apache ни ўрнатиш

Шундай қилиб, компьютерга windows учун мўлжалланган Apache ни ўрнатамиз. Бунинг учун бироз сабр қилишимиз зарур, яъни ишни бошлашдан олдин Apache ни расмий <http://www.apache.org> сайтидан сервер дистрибутивини олишимиз зарур. Windows платформаси учун

серверни энг охирги версиясини танлашни тавсия этаман. Кейин сизнинг системага мослаб Apache ростланади.

Эслатма

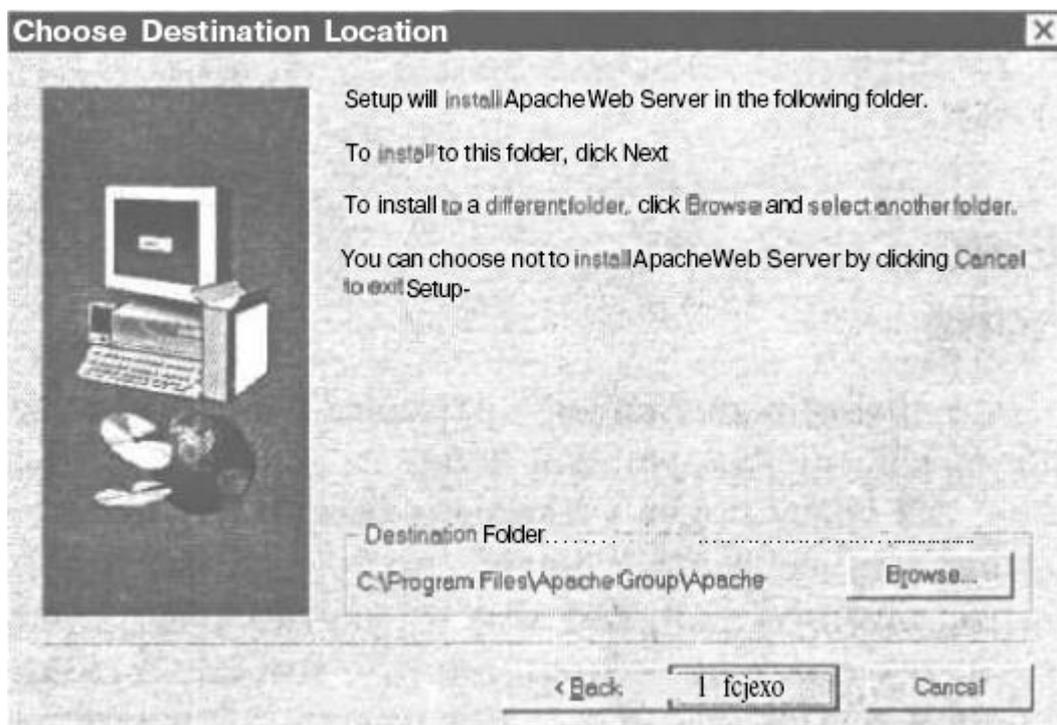
Сиздан қуйида келтирилган амалларни ҳеч бирини қолдирмасдан ва аниқлик билан бажариш сўралади. Чунки Apache ни конфигурациялаш ва ростлаш етарлича мураккаб жараён. Навбатда икир – чикирларигача тушинтирилган кўрсатмалар берилган ва уларни сиз кўрсатилганидек бажарасиз деган умиддамиз. Акс холда анча вақтингизни (ёки кунни) Apache ни документациясидаги конфигурациялаш билан боғлиқ қисмини ўрганишинигизга тўғри келади.

Б и р и н ч и б о с қ и ч : ў р н а т и ш

1. Олинган Apache дистрибутивини ишга туширинг. Хосил бўлган мулоқат ойнасида (расм. 4.1) **Next** тугмасини- кейин лицензия шартларига розиликни билдириш учун —**Yes** ни босинг.

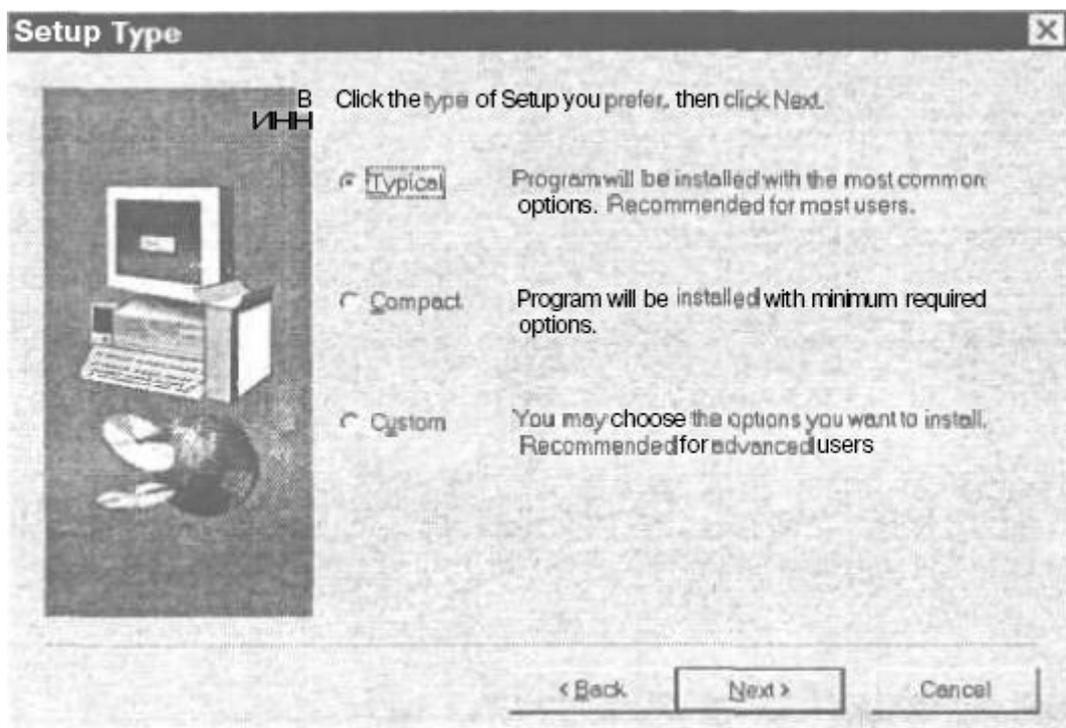


Расм. 4.1. Apache ни ўрнатиш



Расм. 4.2. Серверни ўрнатиш каталоги

2. Apache ни ўрнатиш учун каталогни танлаш сўрови (расм 4.2) ҳосил бўлгунча ҳосил бўлган ойналарда Next тугмасини босинг. Одатда бунинг учун жимликка кўра таклиф этилган каталог қолдирилади (бу C:\Program Files\Apache Group\Apache каталоги). Буни эслаб қолинг.
3. Ҳосил бўлган ойнадан **Typical (Обычная)** байроқчасини ўрнатиш ва Next тугмасини босинг (расм. 4.3).
4. Apache ни ўрнатиш программаси Пуск менюсини Программа менюсида папка яратишни таклиф этади. Next тугмасини босиб бунга рухсат этинг. Программа таъминотини нусхалаш жараёни бошланади.
5. Нусхалаш тугагандан кейин Finish тугмасини босинг. Шунинг билан Серверни ўрнатиш жараёни тугади. Энди уни ростлаш керак.



Р а с м. 4.3. Ўрнатиш типи

И к к и н ч и б о с қ и ч : А р а с h e н и
к о н ф и г у р а ц и я л а ш
ф а й л и н и р о с т л а ш

Бу босқичда сизнинг сайт жойлашадиган каталог аниқланади. Жимликка кўра бунинг учун Apache қуйидагидаги каталогни белгилайди: **C:\Program Files\ ApacheGroup\Apache\htdocs**. Бу каталогдан сервер бўйича документацияларни кўриш мумкин. Менимча, жиддий мақсадларда бу йўл тўғри келмайди, чунки йшл жуда узун. Шунинг учун барча сайт учун Windows даги subst утилити ёрдамида виртуал диск (масалан, z: дискни) яратишни таклиф этамиз. Қуйидаги кўрсатмаларни бажаринг.

1. Сизнинг сайтлар (улар бир нечта бўлиши мумкин) сақланадиган каталогни танланг. Масалан бу каталог C:\INTERNET каталоги бўлсин. Бу каталог янги z: дискни ўзак каталогини ташкил этади.
2. C:\autoexec.bat файлини бошига (бироқ @echo off командасидан кейин, агар бу команда мавжуд бўлса) қуйидаги сатрни киритинг:
subst Z: C:\INTERNET
3. Янги мантиқий z: диск яратилиши учун компьютерни ўчириб қайта юкланг. Энди C:\INTERNET каталогда сақланган файллар z: диск панелида тасвирланади. Бу диск одатдаги қаттиқ диск хусусиятларига эга бўлади.

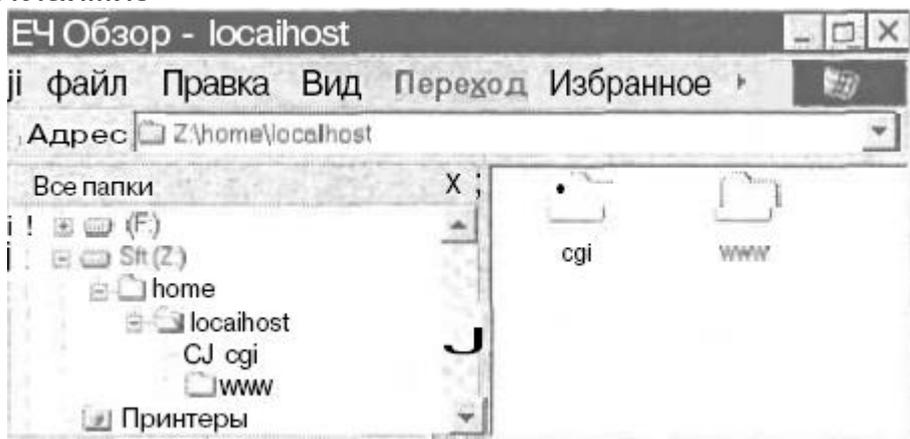
Z: дискни бошқа программа ёрдамида (масалан, Windows таркибидаги *DriveSpace* ёрдамида) яратиш мумкин. Бу усул subst дан фойдаланишга қараганда анча яхшироқ. Хотирани тежаш ва тезлик

нуқтаи назарида ҳам бу тўғри. Чунки Web – сайт нима, бу катта бўлмаган файллар тўплами. DriveSpace эса файллар билан ишлашни оптималлаштиради. DriveSpace дан фойдаланиш ҳақида Windows документациясидан топиш мумкин.

■ Z: дискда home каталогини, унинг ичида эса localhost каталогини яратинг. Унда Apache ни асосий хости маълумотлари сақланади. Бу маълумотларга **http://localhost** адреси орқали мурожат этилади. Охирги яратилган каталогга кириш ва унда cgi ва www каталогларини яратинг. Уларни биринчисида CGI - сценарий, иккинчисида эса – сизнинг документлар ва PHP да тузилган программалар сақланади. Келтирилган амаллар ҳар бир янги виртуал хост (бу ҳақда кейинроқ сўз юритилади) яратилганда бажарилади. Каталогларни яратилган структураси расм 4.4 да кўрсатилган.

Блокнот ёрдамида Apache каталогининг (бизнинг мисолда бу C:\Program Files\Apache Group\Apache)conf нимкаталогда жойлашган httpd.conf конфигурация файлини очинг. Бу файлни очиш учун уни қидирмасдан Асосий менюда **Пуск | Программы | Apache Web Server | Management. Httpd.conf** йўли бўйича бориб **Edit configuration** командасидан фойдаланиш мумкин. Бу файлдаги баъзи сатрларни топиб уларни ўзгартириш керак. Тушунмовчилик келиб чиқмаслиги учун файлдаги бошқа сатрларни ўзгартирманг. Файлни очган бўлсангиз, файлдаги ҳар бир параметрдан кейин изох сатрлари жойлашган. Бир қарашда уларни англаб олиш қийинроқ. Шунинг учун уларга эътибор қилманг.

Аввало Apache ни асосий – localhost хостининг параметрларини ростлаймиз



Расм. 4.4. Асосий хост каталогларининг структураси

■ **serverName** параметри қийматини қуйидагича ўзгартиринг:

```
ServerName localhost
```

serverName майдонидаги изох белгисини ўчиринг, яъни бу параметр олдидаги # символини (жимликка кўра ўрнатилган) ўчиринг. Чунки бу символдан кейинги сатрни Apache инobatга олмайди.

- **DocumentRoot** майдонида сизнинг барча HTML – файллар сақланадиган каталогни кўрсатинг. Олдин келишганимиздек бу `z:\home\localhost\www` каталогидир.

DocumentRoot `z:/home/localhost/www`

- `<Directory/>` сатри билан бошланувчи ва `</Directory/>` сатри билан туговчи секцияни топинг (бундай блоклар кўрсатилган каталог ва унинг нимкаталоглари учун ўрнатмаларни сақлайди). Бу блок кўпгина изохларга эга - бунга эътибор қилманг. Уни қуйидагича шаклда ўзгартиринг:

```
<Directory z:/>
Options Indexes Includes
AllowOverride All
Allow from all
</Directory>
```

Бу блокда барча каталоглар учун жимликка кўра белгиланган ўрнатмалар сақланади (чунки `z:` - ўзак каталог). Бу ўрнатмалар барча каталогларда жимликка кўра индексни автоматик ҳосил қилиш имкониятини, SSI ни қўллаб – қувватлаш ва каталогларни индивидуал ростлаш учун `.htaccess` файллардан фойдаланшга рухсат этиш имониятларини белгилайди.

- Юқоридаги сатрга ўхшаб `<Directory "C:/ProgramFiles/Apache Group/Apache/htdocs">` сатри билан бошланиб `</Directory/>` сатри билан туговчи сатрни топинг. У ерда жуда кўп изох келтирилган, унга эътибор қилманг. Бу секцияни ўчириш зарур, чунки саҳифалар жойлашган каталог ўрнатмаси биз юқорида белгилаган жимликка кўра ўрнатма билан мос бўлиши керак.

DirectoryIndex параметрини қуйидагича ўзгартиринг:

```
DirectoryIndex index.htm index.html
```

- Бу – индекс файллари бўлиб, HTML- документ кўрсатилмаган холларда сервер бирор каталогга мурожаат этганида автоматик қайтаради. Принцип жихатдан олганда, бу ерда бошқа номларни, масалан, `index.php` ва х.к кўрсатиш мумкин. Қолаверса қўшимча ўрнатмаларни ҳар бир алоҳида сайт учун `.htaccess` файлларда белгилаган мақсадга мувофиқ.

- Қуйидаги параметрни топинг ва қийдагича ўзгартиринг:

```
ScriptAlias /cgi-bin/ "z:/home/localhost/cgi/"
```

Ундан кейин қуйидаги сатрни ҳам киритинг:

```
ScriptAlias /cgi/ "z:/home/localhost/cgi/"
```

Бу сизнинг CGI – сценарийлар жойлашадиган каталог бўлади. Бундай параметр Apache га агар **`http://localhost/cgi-bin`** кўринишидаги йўл

кўрсатилганда аслида системадаги **z:/home/iocaihost/cgi** каталогига мурожаат этиш кераклигини билдиради. Биз CGI – каталог учун 2 та тахаллус кўрсатдик, чунки /cgi-bin/ га фақат localhost асосий хост эмас, балки бошқа виртуал хостларни ҳам мурожаат этиш имконияти бўлиши керак. Шу билан бирга уларни хар бирининг ўзининг /cgi/ номли CGI-каталоги мавжуд бўлади.

- Энди <Directory "C:/Program Files/Apache Group/Apache/cgi-bin"> сатри билан бошланувчи ва </Directory> сатри билан туговчи параметрлар блокини топинг. Бу — CGI каталог ўрнатмаларидир. Бу ердаги параметрларни ўзгартириш шарт эмас. Бу блокни ўчириш.
- Қуйидаги параметрни топинг ва узгартиринг (изох белгисини ўчиришни унутманг):

```
AddHandler cgi-script .bat .exe .cgi
```

Бу параметр Apache га exe, bat ва cgi кенгайтмали файлларга CGI – модуллари сифатида қараш кераклигини билдиради.

- Нихоят охиргиси— қуйидаги параметрларни ўрнатинг:

```
AddType text/html .shtml  
AddHandler server-parsed .shtml .html .htm
```

Бунинг натижасида кўрсатилган кенгайтмали файлларни Apache SSI процессори билан қайта ишлайди.

- Энди барча ўзгартиришларни сақланг ва **Блокнотни** ёпинг.

У ч и н ч и б о с қ и ч : А р а с h e н и т е с т л а ш

Юқорида биз Apache параметрларини ўрнатдик ва энди уни ишлашини тестлаб кўрамиз. Серверни ишга тушириш учун **Пуск** тугмасини , кейин **Программы, Apache Web Server, Management** ва **Start Apache** ни танланг. Бунинг натижасида экранда **Сеанс MS-DOS** ойнасига ўхшаш ойна хосил бўлади. Уни ёпманг ва Apache билан ишлаш давомида унга тегманг.

Агар ойна хосил бўлиб шу захоти ёпилса , у холда бу httpd.conf файлида қандайдир хатоликка йўл қўйилганини билдиради. Бу холда хатоликни топиш керак бўлади. Бунинг учун қуйидаги амалларни бажаринг:

1. **Сеанс MS-DOS** ни ишга туширинг. Бунинг учун Пуск тугмасини кейин **Выполнить** ни танланг. Хосил бўлган мулоқат ойнасида command сатрини танланг ва <Enter> тугмасини босинг. Командалар сатри хосил бўлади.
2. Қуйидаги DOS командаларини киритинг:

```
c :  
cd "\Program Files\Apache Group\Apache"  
apache.exe
```

3. Агар бу вақтда Apache ишламаётган бўлса, у холда хатолик ҳақидаги хабар ва httpd.conf файлида хатолик юз берган сатр номери хосил

бўлади. httpd.conf файлидаги хатоликни тўғирланг ва келтирилган амалларни қайта бажаринг. Агар хатолик йўқ бўлса "Apache/1.3.14 (Win32) running..." ёзуви хосил бўлади.

Энди сервер тўғри ростланганлигини текшириб кўрамиз.

HTML ни текшириш

Apache ни HTML – документлари сақланадиган z:/home/localhost/www каталогида ихтиёрий текстли index.html файлини яратинг. Кейин браузерни ишга туширинг ва қуйидаги киритинг:

```
http://localhost/index.html
```

ёки

```
http://localhost/
```

натихада сиз яратган HTML – файл хосил бўлади.

П р о в е р к а SSI

Apache ни HTML – документлари сақланадиган z:/home/localhost/www каталогида қуйидаги текстли test.shtml файлини яратинг (include директивасидаги пробелларга эътибо қилинг ва тўғри киритинг):

Листинг 4.1. test.shtml файли

```
SSI Test!<hr>
<!--#include virtual="/index.html" -->
<hr>
```

Кейин браузерни адрес сатрида қуйидагини киритинг:

```
http://localhost/test.shtml
```

Натижада "SSI Test!" тексти ва ундан кейин 2 та горизонтал чизиқлар орасида index.html файлидаги текст жойлашган файл очилиши керак. Агар бу содир бўлса, демак SSI тўғри конфигурацияланган.

CGI ни текшириш

CGI – сценарийлар сақланадиган z:/home/localhost/cgi каталогида қуйидаги буйруқлар келтирилган test.bat файлини яратинг

Листинг 4.2. Файл test.bat

```
@echo off
echo Content-type: text/html
echo.
echo.
Dir
```

Кейин браузерни адрес сатрида қуйидагини киритинг:

```
http://localhost/cgi-bin/test.bat
```

Ойнада DOS ни `dir` командасини натижаси тасвирланиши керак.

Эслатма

Сиздан куйида келтирилган амалларни хеч бирини қолдирмасдан ва аниқлик билан бажариш сўралади. Чунки Apache ни конфигурациялаш ва ростлаш етарлича мураккаб жараён. Навбатда икир – чикирларигача тушинтирилган кўрсатмалар берилган ва уларни сиз кўрсатилганидек бажарасиз

Агар бирор хатолик мавжуд бўлса , ёки Apache ойнаси очилиб шу захоти ёпилса, демак у холда қандайдир хатоликка йўл қўйилган. Бу хатолик `httpd.conf` файлида аниқланади. Бу хатоликларни аниқлаш учун `C:\Program Files\Apache Group\Apache\logs` каталогида жойлашган `log`-файлларни кўриб чиқинг.

Назорат саволлари

1 Apache ни вазифаси ?

2 Apache ни ўрнатиш коидалари ?

**3 Apache ни конфигурациялаш файлини
ростлаш ?**

4 Apache ни тестлаш нима ?

МАЪРУЗА №10

**PHP нинг Асосий синтаксис .Маълумотларнинг типлари
,Маълумотларни бошқариш структуралари**

Режа 1 PHP нинг Асосий синтаксис

2 Маълумотларнинг типлари ўзгарувчилар

3 Ифодалар ва Маълумотларни бошқариш структуралари

PHP да Китобни ушбу қисмигача тўлиқ танишган бўлсангиз, демак сиз СИ тилида сценарий яратиш ноқулай деган тўхтамга келгансиз. (Агар бундай тўхтамга келмаган бўлсангиз , демак китобни юқоридаги қисмлари ёмон ёзилган).

Хуш, сценарийни қайси тилда яратамиз? Кўпчилик бу саволга :”Албатта одатда сценарий ёзиладиган —Perl да!” деб жавоб беради. Ха, бу кенг тарқалган қарашдир. Бироқ Perl тилини ўзининг имкониятлари билан камчиликлари хам мавжуд. Қолаверса бу камчиликлари анча жиддий. Мана масалан, улардан бири: Perl тили бевосита сценарий яратиш учун мўлжалланмаган. Бу тил қайсидир маънода универсал тил, шунинг учун бевосита бизга зарур нарсаларни қўллаб қувватламайди. Мана иккинчи камчилиги: Perl тили синтаксиси дастур матнини ўқилиши қийинлаштиради. Бу масаладаPerl тилб Си га хам, Паскаль тилига хам ўхшамайди.

PHP — тили Интернетда ишлаш учун мўлжалланган бўлиб, Си тили каби универсал ва аниқ синтаксисга эга ва ўзида Perl ва Си тили имкониятларини мужассамлаштириб олган. Бу тил яқинда пайдо бўлган

бўлсада (аниқроғи уни интерпретатори) у миллионлаган серверларга ўрнатилган ва ўрнатилишида давом этмоқда.

PHP тилини Perl га нисбатан битта жиддий камчилиги бор: бу — унинг катта ва мураккаб сценарийлар билан секин ишлашидир. Бироқ бу камчиликни PHP ни яратувчилар бартараф қилиш ниятида эканлигини билидиршган.

Интерпретаторми ёки компилятор?

Эхтимол, PHP 4 аввалги версияларидан фарқли равишда компилятор эканлигини эшитгандирсиз. Бу унчалик тўғри эмас. Терминлардаги тушунмовчиликларни ойдинлаштириш учун нима компилятор, нимани интерпретатор деб аташни келишиб оламиз. Кўпинча компилятор деб интерпретацияловчи транслятор дастурни, яъни асосий функцияси интерпретатор бўлган дастурни аташади. PHP ни 4 версиясида ҳам худди шундай.

Келинг, PHP 4 версияси қандай ишлашини кўриб чиқамиз. У дастурни кодини олиб, биринчи навбатда уни тахлил қилади (хусусан, синтаксисини текширади) ва махсус *ички кўринишга трансляция* қилади. Бу кўриниш махсус байт – коддан иборат бўлиб, уни кўз билан кўриб тушуниб олиш имкони йўқ. Бироқ уни PHP амалга ошириши учун жуда қулай. Мана шу жараённи кўпинча кўпчилик нотўғри холда компиляция деб аташади. Кейин PHP хосил бўлган байт – кодни бажаради (интерпретациялайди). Бу жараён эса одатдаги интерпретатор ишини акс эттиради.

Юқорида такидлаганимиздек, PHP бир – биридан деярли мустақил бўлган 2 та блок — транслятор ва интерпретатордан ташкил топган. Нима учун бундай тузилган? Албатта, бу тезликни оширишни кўзлаб шундай қилинган. Ўзингиз тахлил қилиб кўринг: синтактик тахлил фақатгина трансляция пайтида бир марта амалга оширилади, тайёр бўлган "яримфабрикат" — байт-код кўп марта бажарилади.

Фараз қилайлик, дастурда кўп марта такрорланувчи цикл мавжуд бўлсин. PHP ни трансляция фаза йўқ бўлган 3 версиясида такрорлашни навбатдаги хар бир қадамида цикл коди тахлил қилиниши, сатрлари тахлил қилиниши, синтаксиси текширилиши ва х.к. лар амалга оширилиши зарур. PHP 4 версиясида бу *бир марта* амалга оширилади ва циклни хар бир итерациясида тайёр байт – коддан фойдаланилади.

Хусусан, юқорида келтирилган PHP ни ишлаш схемаси амалдагига ҳамма вақт ҳам тўғри келмайди. Чунки, бу тилда шундай конструкцияларни яратиш мумкин, уларни трансляция пайтида (бунга масалан, дастур кодига номи фойдаланувчи томонидан кўрсатиладиган ташқи файлдаги кодни киритиш инструкцияси мисол бўлади) ички тасвирга ўтказишни сира имкони йўқ. Бундай пайтда PHP бу конструкциясини қолдириб кетади ва бу конструкцияси бажариш пайтида трансляция қилади. Албатта бу дастур бажарилишини секинлатади.

Интерпретаторни қулайликлари

ва камчиликлари

Агар сиз— собиқ системшиқ ёки амалий дастурловчи ва Perl тили билан таниш бўлмасангиз, у холда PHP тили бошқа Web га мўлжалланган тиллар каби интерпретатор эканлигига кўникишингиз қийин бўлади. Ҳа, PHP тилида яратилган сценарий Си тилидаги сценарийга нисбатан минг марта секин (бироқ Perl тилидаги билан деярли тенг тезликда) ишлайди ва бунга кўникишингизга тўғри келади. Масалан, агар Си тилида қуйидагидек миллион итерацияли бўш цикл яратсак:

```
for (long i=0; K1000000; i++);
```

у холда бу секундни улушларида ишини тугатади. Худди шундай PHP даги цикл:

```
for ($i=0; $i<1000000; $i++) ;
```

Pentium 100 процессорида бир неча секундда бажарилади.

Бироқ таркибида катта цикллар цўқ бўлган сценарийларни ишлаш вақти унчалик фарқ қилмайди. Умуман, маълумотларни интернет каналарида модем орқали 5 секундда алмашинадиган бўлса, у холда сценарий ишлаш вақтини, работает 0,01 секунд ёки 0,1 секунд бўлганлигини нима фарқи бор?

Кўпчилик қуйидаги саволни беришади, — агар минглаган фойдаланувчи, адреслар ва телефон рақамлари сақланувчи файллар билан иш кўрувчи сценарийларни қандай чратиш керак? Чунки, бирор фойдаланувчини топиш учун (айниқса уни исми аниқ берилмаган бўлса) уларни хаммасини кўриб чиқиш зарур бўлади. Бу жараён эса кўп сонли итерациядан иборат циклни талаб этади. Агар жуда кўп маълумотлар массивини қайта ишлаш керак бўлса, у холда Си ёки бирор ...маълумотлар базасидан фойдаланган мақсадга мувофиқ. Маълумотлар базаси — бу бир хил структурали кўп сонли ёзувлар ва бу билан бирга ёзувларни излаш, қўшиш, ўчиришни бажарувчи дастурлар (кўпинча Си тилида яратилган) тўпламидир. PHP кўпгина маълумотлар базаларибилан ишлашни қўллаб – қувватлайди. Шунинг учун маълумотлар базасини қўлловчи сценарийларни яратишда муаммо келтириб чиқармайди.

PHP интерпретаторини қуйидаги устунликлари мавжуд.

- Дастурни бажариш пайтида хатони аниқлаш соддалашади. Агар дастур бажарилишида узилиш бўлса интерпретатор бу ҳақда хабар қилади.
- Эгалланган хотирани бўшатиш ҳақида уйлмаса ҳам бўлади. Интерпретаторни ўзи дастурдаги у ёки бу ўзгарувчи керак ёки керакмаслигини аниқлайди ва ўзгарувчи учун ажратилган хотирани бўшатади.
- Ўзгарувчи типлари ҳақида ўйламаса ҳам бўлади.

PHP – дастурга мисол

Анъанавий равишда дастурлаш хақидаги барча китоблар "Hello world!" дастуридан бошланади. Традиционно, любая книга начинается с программы "Hello world!". Биз хам ушбу анъанани давом эттирган холда 2 та мисол келтирамиз. биринчиси:

```
<?
echo "Hello world!
?>
```

Сценарийни браузерда ишга туширинг. Уни ишлаётганлигига гувоҳ бўласиз. Маълумотларнинг типлари ўзгарувчилар ва ифодалар

Кўриниб турганидек PHP – дастурни структураси СИ ва Бейсик аралашмасини эслатади ва HTML таркибига хам киритилади. Бироқ биз юқорида PHP тилидаги оддий дастур мисолини кўриб чиқдик. Бу тил хақидаги умумий тасаввурни бермайди. Бу бўлимда PHP тили конструкцияларини ўрганамиз, яъни тил асосларини ўрганамиз...

Ўзгарувчилар

Худди бошқа тиллардаги каби PHP да хам ўзгарувчи тушунчаси мавжуд. Юқорида келтирилган оддий мисолда 3 та ўзгарувчидан фойдаланилган!

При программировании на PHP принято не скупиться на объявление новых переменных, даже если можно обойтись и без них. Масалан, келтирилган оддий мисолда фақат битта – цикл счетчигидан фойдаланиш мумкинэди. Бироқ кўпроқ ўзгарувчидан фойдаланса дастур тушунарлироқ бўлади. Интерпретаторга идентификатор яратиш жуда осон. Бу функциялардаги барча ўзгарувчилар жимликка кўра локал эканлиги билан боғлиқ (локал ўзгарувчилар хақида кейин тушунча берилади).

Ўзгарувчи номидаги бош ёки кичик харф ишлатилиши ахамиятга эга: масалан, `$my_variable` — номи `$My_Variable` ёки `$MY_VARIABLE` номи билан бир хил эмас. Бундан ташқари ўзгарувчи номи `$` белгиси билан бошланади. Шундай бўлса интерпретатор ўзгарувчиларни “тушуниб” ва фарқ олиши осон бўлади, масалан, сатрдаларда. Аввалига бу қондан эслаб қолиш қийин, бироқ бунга вақт ўтиши билан кўникиш жуда осон.

PHP тилида ўзгарувчи — ўзига хос объект бўлиб қиймати ихтиёрий бўлиши мумкин. Агар дастурда бирон нима сақланадиган бўлса, у холда у албатта ўзгарувчида сақланади (константада фақат сон ёки сатр сақланади. Бу тилда кўрсаткич тушунчаси йўқ — ўзлаштиришда ўзгарувчи қандай мураккаб структурага эга бўлмасин бирга –бар нусхаланади. Бундан ташқари PHP ни 4 версиясида **ссилка** тушунчаси мавжуд. Булар қатъий ва символик **ссилка**.

Ююқорида такидланганидек PHP да ўзгарувчиларни олдиндан тавсифлаш ва типини кўрсатиш зарур эмас. Бу ишларни барчасини интерпретаторни ўзи амалга оширади. Бироқ бунда баъзи холларда интерпретатор хатога йўл қўйиши мумкин (масалан, агар текст сатрида

ўнлик сон берилган бўлса). Шунинг учун камдан – кам холларда у ёки бу ифода қандай типли эканлигини кўрсатиш керак бўлади.

Баъзи холларда бевосита дастур бажарилиши жараёнида ўзгарувчи типини аниқлаш зарур бўлиб қолади (масалан, функция параметрида берилган ўзгарувчини). PHP тилидаги маълумотлар типлари билан танишиб чиқамиз.

Ў з г а р у в ч и т и п л а р и

PHP тили ўзгарувчиларни 5 та типини қўллаб – қувватлайди. Уларни қисқача тарифлаймиз.

integer

Ишорали бутун сон. Узунлиги 32 бит (2 147 483 648 дан 2 147 483 647 гача)

double

Етарлича катта аниқликдаги хақиқий сон (математик ҳисоб – китоблар учун етарли аниқликда)

string

Ихтиёрий узунликдаги сатр. Си тилидагидан фарқли равишда сатр таркибида **нулевой символлар** ҳам қатнашади ва бу дастур ишига таъсир қилмайди. Бошқача айтганда, сатрлардан бинар маълумотларни сақлаш учун фойдаланиш мумкин. Бунда сатр узунлиги бўш оператив хотира хажми билан чегараланади. Шунинг учун битта сатрга хажми катта бўлган, 200 – 300 килобайт файлдаги маълумотни ўзлаштириш мумкин. Сатрлар стандарт функциялар ёрдамида қайта ишланиши мумкин ва сатрдаги ихтиёрий символга мурожаат этиш мумкин.

array

Ассоциатив массив (ёки кўпинча, PHP да бундай аташ унчалик тўқ-ри бшлмасида хэш деб аталади). Бу бир нечта элементлар тўплами бўлиб, уларни ҳар бири калит => қиймат шаклидаги жуфтликдан иборат (=> симболи билан маълум калитга бирор қийматни молиги белгиланган). Маълум элементга калити кўрсатилиб мурожат этилади. Си тилидаги массивлардан фарқли равишда фақат 0 дан бошланувчи бутун сонлар эмас, балки ихтиёрий сатр қўлланилиши мумкин. Масалан, қуйидаги командалар ишлатилиши мумкин:

```
// "0", "a", "b" ва "c" калитли массивни яратиш;  
$a=array(0=>"zzzz", "a"=>"aaa", "b"=>"bbb", "c"="ccc");  
echo $a["b"]; // "bbb" чатрни чиқаради;  
$a["1"]="qq"; // массивни янги элементини яратади ва унга  
    "qq" қийматни ўзлаштиради;  
$a["a"]="new_aaa"; // мавжуд элементга "new_aaa" ни  
    ўзлаштиради;
```

Array() оператори элементлари қавс ичида кўрсатилган холда массив яратади.

object

Объект типни объектга – ориентирланган дастурлашни энг оддий хусусиятларини ўзида мужассамлаштирган. Объектни ички структураси хэш га ўхшаш. Фақат бунда алохида элемент ва функцияга мурожат этиш учун квадрат қавслар эмас, балки -> оператори ишлатилади. Объектлар тўғрисида кейинчалик батафсил маълумотлар келтирилади.

Мантиқий ўзгарувчилар

PHP да яна ўзгарувчиларни гипотетик типни — мантиқий типни мавжуд. Мантиқий ўзгарувчи иккита: false (ёлғон) ёки true (рост) қийматдан бирига эга бўлади. Нолга тенг бўлмаган ихтиёрий сон (ва бўш бўлмаган сатр), ҳамда true хизматчи сўзи рост қийматни билдиради. 0 сони, бўш сатр ва false сўзи ёлғон қийматни билдиради. Шундай қилиб, ихтиёрий нолга тенг бўлмаган қиймат (хусусан, ўзгарувчиларни қиймати ҳам) мантиқий контекстда (ифодада) рост қиймат сифатида қаралади. Дастурда қийматни кўрсатиш учун false ва true сўзларидан ҳам фойдаланиш мумкин.

false ва true калит сўзлари — оддий константалар (ўзгармаслар) эмас. Юқорида айтилганидек, false бўш сатрни синоними, true эса — бирни синоними ҳисобланади. Бунга қуйидаги келтирилган операторлар бажариб ишонч ҳосил қилишингиз мумкин:

```
echo false; // бўш сатрни чиқаради, яъни ҳеч нима  
чиқармайди  
echo true; // 1 сонини чиқаради
```

Келинг энди қуйидаги дастурни кўриб чиқамиз (листинг 7.1).

Л и с т и н г 7.1. Мантиқий катталиклар

```
<?  
$a=100;  
if ($a==100) echo "ўзгарувчи 1 га тенг!<br>"  
if ($a==true) echo "ўзгарувчи рост!<br>"  
?>
```

Агар true фақат 1 га тенг бўлса, у ҳолда фақат биринчи сатр чиққан бўларди. Бироқ иккала сатр ҳам браузерда чиқади. Солиштириш операторларида (масалан, == солиштириш операторида, ҳамда >, < операторида ва х.к.) агар иккала операнд ҳам мантиқий бўлса PHP операндлардан бирига мантиқий операнд сифатида қарайди. Навбатдаги мисолда (листинг 7.2) PHP ҳар бир ўзгарувчи учун мантиқий эканлигини билдирувчи белги сақлашини кўрсатади..

Л и с т и н г 7.2. Мантиқий ўзгарувчилар

```
<?  
$a=100;  
$b=true;  
echo "a = $a<br>";
```

```
echo "b = $b<br>";
if($a==$b) echo 'a bга "тенг" ! ' ;
?>
```

Қизиқ, дастур аввал "a=100 ва b=1" ёзувларини чиаради, кейинги сатрда эса a bга "тенг" ! ёзувини чиқаради. Бу дастур натижаси худди шундай бўлиши энди бизга маълум (чунки ўзгарувчилар мант иқий ўзгарувчилар сифатида қаралади). Шунинг учун \$a ўрнида, масалан, функция натижаси бўлган сон ишлатилган эҳтиёт бўлинг. Акс холда бу излаб топиш учун бир неча соат вақтни оладиган хатоликни келтириб чиқаради.

Албатта арифметик амалларни бажаришда у мантиқий ўзгарувчидан одатдаги, сонли ўзгарувчига айланади. Мантиқий ўзгарувчиларни қийматини биттага ошириш ёки камайтириш учун ++ ёки – операторларидан фойдаланиш бефойда. (листинг 7.3):

Л и с т и н г 7.3. ++ ва — операторларини хусусиятлари

```
<?
$b=true;
echo "b: $b<br>";
$b++;
echo "b: $b<br>";
?>
```

Бу дастур иккала чиқаришда ҳам 1 қийматни чиқаради. Бу дастур сиз йўл қўйишингиз мумкин бўлган хатоликни олдини олиш учун келтирилган.

Ўзгарувчилар устида амаллар бажариш

Ўзгарувчи типи қандай бўлишидан қатъий назар, ўзгарувчи устида 3 та асосий амални бажариш мумкин.

Қиймат ўзлаштириш (ёки бериш)

Ўзгарувчига бошқа бир ўзгарувчини қийматини (ёки функция натижасини), бошқа ўзгарувчига ссилка, ёки константа ифодани ўзлаштириш мумкин (объект типидан ташқари). Юқорида айтилганидек, типларни ўзгартиришни интерпретатор ўз зиммасига олади. Бундан ташқари ўзлаштирганда ўзгарувчини эски қиймати, энг асосийси ўзгарувчини эски типи йўқотилади ва “қиймат берувчини” нусхаси бўлиб қолади. Яъни, агар биз массивга сон ўзлаштирадик, у холда бутун массив йўқатилади.

Мавжудлигини текшириш

Бирор ўзгарувчи бор ёки йўқлигини текшириш мумкин. Бу IsSet() оператори ёрдамида текширилади. Масалан:

```
if (IsSet ($MyVar))
echo "<Бундай ўзгарувчи мавжуд. Унинг қиймати $MyVar";
```

Агар жорий вақтда ўзгарувчи йўқ бўлса (яъни унга қиймат ўзлаштирилмаган ёки олдинроқ unset() оператори ёрдамида ўчирилган

бўлса), у холда IsSet () оператори натижаси ёлғон, акс холда рост қийматли бўлади.

Дастурда инициализация этилмаган ўзгарувчидан фойдаланиш мумкин эмас, акс холда хатолик келиб чиқади (сценарийда мантиқий хатолик ҳақида хабар хосил бўлади). Албатта, зарур ўзгарувчини бўш сатрга тенглаб яратиш мумкин. Бироқ бу усулдан фойдаланишни маслаҳат этмайман. Бунинг ўрнига ўзгарувчини ишлатишдан олдин мавжудлигини текшириш мақсадга мувофиқ. Бунда хатолик хабарларини чиқаришни @ оператори ёрдамида бекор қилиш мумкин.

Йўқ қилиш

Ўзгарувчини йўқ қилиш Unset() оператори ёрдамида амалга оширилади. Бунинг натижасида ўзгарувчи ички таблицадан ўчирилади, яъни дастур йўқ қилинган ўзгарувчи яратилмаган деб ҳисоблайди. Масалан:

```
// $a ўзгарувчиси мавжуд эмас
$a="Hello there!";
//Энди $a ўзгарувчи яратилди
// ... $a дан фойдаланувчи қандайдир операторлар
echo $a;
// энди $a ўзгарувчини ўчирамиз
Unset ($a) ;
// Энди $a ўзгарувчиси йўқ
echo $a; // Хатолик: $a ўзгарувчиси йўқ
```

Впрочем, применение Unset () для работы с обычными переменными редко

бывает целесообразно. Куда как полезнее использовать его для удаления

элемента в ассоциативном массиве. Например, если в массиве \$A нужно удалить элемент с ключом for_del, это можно сделать так:

```
Unset ($A["for_del"]);
```

Теперь элемент for_del не просто стал пустым, а именно удалился, и последующий перебор элементов массива его не обнаружит.

Кўрсатгич типли ўзгарувчилар

PHP тилида кўрсатгич тушунчаси йўқ бўлсада бошқа ўзгарувчиларга кўрсатгич яратиш мумкин. Кўрсатгични 2 та тури бор: қатъий ва символик. Қатъий кўрсатгичлар PHP 4 да пайдо бўлди. (3 версиясида фақат символли кўрсатгичлар мавжуд эди.

Қатъий кўрсатгичлар

Қатъий кўрсатгич оддий ўзгарувчи бўлиб, бошқа ўзгарувчини синоними ҳисобланади. Кўп даражали кўрсатгичлар қўллаб қувватланмайди. Шунинг учун қатъий кўрсатгичларга оддий синоним сифатидагина қараш мумкин. Қатъий кўрсатгич яратиш учун & (амперсанд) операторидан фойдаланилади. Масалан:

```

$a=10;
$b = &$a; // энди $b – ўзгарувчиси $a билан бир хил
$b=0; //аслида $a=0
echo "b=$b, a=$a"; // "b=0, a=0" ёзувини чиқаради

```

Фақат ўзгарувчини эмас, шунингдек массив элементини ҳам кўрсатиш мумкин (бу билан қатъий кўрсатгич символли кўрсатгичдан ажралиб туради). Масалан:

```

$A=array('a' => 'aaa', 'b' => 'bbb');
$b=&$A['b']; //энди $b — массивни'b'индексли элементи билан бир хил
$b=0; //аслида $A['b']=0;
echo $A['b'] //экранда 0 чиқади

```

Хусусан, навбатдаги мисолдаги каби яратилаётган символик кўрсатгич кўрсатувчи массив элементи мавжуд бўлмаслиги ҳам мумкин :

```

$A=array('a' => 'aaa', 'b' => 'bbb');
$b=&$A['c']; //энди $b – массивни 'c' индексли элементи билан бир хил
echo "'c' индексли элемент: (".$A['c'].")";

```

Ушбу дастур бажарилиши натижасида \$b кўрсатгичига қиймат ўзлаштирилмаган бўлсада, \$A массивида 'c' калитли элемент яратилади ва уни қиймати бўш сатрга тенг бўлади (буни echo оператори иши натижасидан билиб олиш мумкин). Яъни қатъий кўрсатгич мавжуд бўлмаган объектни кўрсатиши мумкин эмас ва агар бунга ҳаракат қилинса кўрсатилувчи объект яратилади.

Қатъий кўрсатгич – у кўрсатаётган объектни абсолют аниқ синоними эмас. Чунки қатъий кўрсатгичларга нисбатан қўлланиладиган unset() оператори кўрсатилаётган объектни ўчирмайди, балки кўрсатгич ва объект ўртасидаги боғлиқликни бекор қилади.

Демак, қатъий кўрсатгичлар ва кўрсатилаётган ўзгарувчилар (объектлар) умуман олганда бир хил имконга эга, бироқ улардан бирини қиймати ўзгартирилса бу иккинчида ҳам ўз аксини топади. Unset () оператори объект ва кўрсатгич орасидаги алоқани узади.

Қатъий кўрсатгичлардан функцияларга параметрни беришда ва функция қайтарадиган қийматни ўзлаштиришда фойдаланиш мақсадга мувофиқ. Бу қандай амалга оширилишини PHP да функциялар яратишни ўрганишда кўриб чиқамиз.

Символик кўрсатгичлар

Символик кўрсатгич — бу бошқа бир ўзгарувчини номини сақловчи сатрли ўзгарувчидир. Символик кўрсатгич кўрсатаётган ўзгарувчи қийматини кўриш учун қайта номлаш операторидан (кўрсатгич номи олдида кўшимча равишда \$ белгиси кўрсатилади) фойдаланиш зарур. Буни мисолда кўриб чиқамиз:

```

$a=10;

```

```

$b=20;
$c=30;
$p="a"; //ёки $p="b" ёки $p="c" ($p га бошқа ўзгарувчи номи
ўзлаштирилади)
echo $$p; // $p кўрсатаётган ўзгарувчи қиймати, $a қиймати чиқарилади
$$p=100; // $a га 100 қиймати ўзлаштирилади

```

Кўриб турганимиздек, оддий сатрли ўзгарувчини кўрсатгич сифатида фойдаланиш учун уни олдида яна битта \$ белгисини қўйиш керак. Бу эса интерпретаторга \$p ни қийматини эмас, балки \$p да номи сақланаётган ўзгарувчи қийматини олиш кераклигини билдиради.

Символик кўрсатгичлар кам ишлатилади. Бундай кўрсатгичлардан фойдаланиш дастурдаги хатоликларга йўл қўйишга сабаб бўлади.

Баъзи шартли белгилашлар

Юқорида эслатилганидек, PHP да бирор ўзгарувчи ёки ифода типини аниқ кўрсатиш зарур эмас. Бироқ дастурдаги хар бир катталikka конкрет тип мос келади ва бу уни типи дастур бажарилиши давомида бошқа типга ўзгартирилиши мумкин. Бундай ўзгартириш, масалан, “20” сатрига 10 сонини қўшиб, натижада 30 (“2010” эмас) қийматга эга бўлсак, у холда бу PHP да тўғри амалга ошган сондан сатрга ва аксинча ўзгартиришга мисол бўлади.

Фараз қилинг, \$a ўзгарувчи типини сонли типга ўзгартириш зарур. \$a ўзгарувчи эса массив бўлсин. Олдиндан маълумки, типни бундай ўзгартириш хеч қандай маънога эмас. Агар бунга ҳаракат қилинса, масалан, \$a га 10 ни қўшишга ҳаракат қилинса PHP бу ҳақда хатолик хабари беради.ёки хабар бермаслиги ҳам мумкин (масалан, массив сатрга ўзгартирилса, у холда доимо “Array” сатри хосил бўлади). Шунинг учун PHP даги стандарт функция ва операторларни (улар жуда кўп) ўрганишда функция ва операторларни у ёки бу параметри типини аниқлаштириб олиш керак бўлади. Функция ва операторларда мос бўлмаган типлар қўлланилмаслиги керак. Шунинг учун матнда зарур бўлганда ўзгарувчи ва функцияни қайтарадиган қиймати типини, ҳамда бошқа метаимволлар кўрсатиб ўтилади. Масалан, FuncName номли функцияни таърифлаш мисоли:

```

<return_type> FuncName (<type1> $param1 [, <type1>
$param2])

```

Масалан, бу функция ҳам қандайдир қийматларни қабул қилади ва натижа қайтаради. Бу ерда <return_type> типли қиймат қайтарувчи ва битта ёки иккита аргументли (квадрат қавсда кўрсатилган иккинчи аргумент кўрсатилиши шарт эмас) функция таърифланган. Биринчи параметр типи <type1>, иккинчи параметрники — <type2>. Бурчак қавсларда келтирилган мумкин бўлган типларни таърифлари навбатдаги мавзуларда келтирилган.

string

оддий сатр ёки сатрга ўзгартириш мумкин бўлган тип.

int, long

Ёки бутун, ёки ҳақиқий сон (каср қисми тушириб қолдирилади), ёки маълум форматдаги сонлардан иборат сатр. Агар сатрни `int` типга ўзгартириш имкони бўлмаса уни ўрнига `0` қиймати олинади ва ҳеч қандай хабар ҳосил бўлмайди.

double, float

Қуйидаги типли сондан иборат ёки ҳақиқий сон, ёки бутун сон, ёки сатр.

bool

Мантиқий тип бўлиб, ёлғон қиймат (нол қиймат, бўш сатр ёки `false` константаси) каби, ёки рост қиймат (қолган барча қийматларда) каби қабул қилинади.

array

Умумий ҳолда ассоциатив бўлган массив (пастга қаранг). Яъни калит = қиймат шаклидаги жуфтликлар тўплами. Хусусан бу ерда `list` рўйхати ҳам бўлиши мумкин.

list

Одатда бу индекслари `0` дан бошлаб кетма - кет бутун сонлар билан номерланган массив. Рўйхат ассоциатив массивни бир шаклидир. Шунинг учун функцияларни `list` типли параметри сифатида `array` типли параметрни кўрсатиш мумкин. Бунда функция “ҳеч нимани сезмайди” ва массив элементларини номерлаб худди рўйхат каби иш олиб боради.

object

Бирор структурага эга бўлган объект. Одатда уни структураси аниқланган бўлади.

void

Энг оддий тип бўлиб, фақат функция қайтарадиган қийматни аниқлаш учун қўлланилади. PHP да функция ҳеч нима қайтармаслиги мумкин эмас, шунинг учун амалда барча `void` функциялар `false` қиймат (яъни бўш сатр) қайтаради.

mixed

Ихтиёрий тип. Бу тип бутун, каср сон, сатр, массив ёки объект бўлиши мумкин. Масалан, `gettype ()` ёки `settype ()` стандарт функциялари `mixed` типли параметрга эга. Яъни, агар функция `mixed` типли қиймат қатарадиган бўлса, у ҳолда бу қиймат (натижа) типни операндлар олрқали аниқланади ва функция таърифи орқали аниқланади.

Ифодалар ва Маълумотларни бошқариш структуралари

Ифода — бу PHP даги асосий элементлардан биридир. Ҳақиқатдан ҳам дастурда ёзилган барча матн — бу ифодалардан иборат.

Ифодага энг оддий мисол — ўзлаштириш операторини ўнг томонида турган ўзгарувчи ёки константа. Масалан, қуйидагича шаклда жойлашган 5 сони ифодага мисол бўлади

```
$a=5;
```

Бу ифода, уни қиймати 5 га тенг. Бу ўзлаштиришдан кейин \$a қиймати 5 га тенг деб бемалол айта оламиз. Энди қуйидаги ифодани олсак

```
$b=$a;
```

Бунад \$b қиймати 5 га тенг, чунки ўзлаштириш операторини ўнг томонидаги \$a ифодани қиймати 5 га тенг эди.

Яна бир мисол. Эсласангиз, биз юқорида дастурнинг ўзи ифодалардан ташкил топишини айтгандик. Демак, \$b=\$a – ҳам ифода! Навбатда келтириладиган ифодалар Си ва Perl тилини билганлар учун дарров тушунарли бўлади:

```
$a=($b=10); // или просто $a=$b=10
```

Бунда \$a ва \$b ўзгарувчиларга 10 қиймати ўзлаштирилади. Яна бир нисбатан мураккаброқ мисол:

```
$a=3*sin($b=$c+10)+$d;
```

Бу командалар бажарилганидан кейин ўзгарувчилар қиймати нимага тенг бўлади? Бунда қуйидаги операторлар бажарилишида ҳосил бўладиган натижалар ҳосил бўлади:

```
$b=$c+10;
```

```
$a=3*sin($c+10)+$d;
```

Кўриб турганимиздек, PHP да мураккаб ифодалардан фойдаланиш мумкин. Яъни ўзлаштириш операторида қатнашган ўзгарувчи қийматини ифода ичида аниқлаш мумкин. Бу имконият дастур кодини қисқартиради ва дастурни ўқишни соддалаштиради. Шунинг учун бу усулдан фойдаланиб туриш лозим.

Ихтиёрий ифодани қийматини типи мавжуд, яъни олдиндан аниқлаш мумкин. Масалан:

```
$a=10*20;
```

```
$b="" . (10*20);
```

```
echo "$a:", gettype($a), " $b:", gettype($b);
```

```
// ҳосил бўлувчи текст "200:integer 200:string"
```

Бирор қийматни типини бошқасига ўзгартириш учун (масалан, сонли сатрдан иборат string типли \$b ни типи бизга маъқул бўлмаслиги мумкин) типни ўзгартириш операторидан фойдаланилади. Бу операторлар функционал ва префиксли операторлар шаклида қўлланилиши мумкин. Масалан, қуйидаги 2 та операторлар ўзаро тенг кучлидир:

```
$a = intval($b);
```

```
$a = (int) $b;
```

Типни ўзгартириш операторлари:

■ `$b=intval(ифода)` ёки `$b=(int)(ифода)`
Ифода қийматини бутун типга ўзгартиради ва уни `$b` ўзгарувчига ўзлаштиради.

■ `$b=doubleval(ифода)` ёки `$b=(double)(ифода)`
Ифода қийматини хақиқий типга ўзгартиради ва `$b` ўзгарувчига ўзлаштиради.

■ `$b=strval(ифода)` ёки `$b=(string)(ифода)`
Ифода қийматини сатрга ўзгаритради.

■ `$b=(bool)(ифода)`
Ифода қийматини мантиқий типга ўзгаритради. Яъни бу оператор бажарилганидан кейин `$b` қиймати ёки `true`, ёки `false` га тенг бўлади.

Умуман яна 2 та: `(array)` ва `(object)` оператори мавжуд бўлиб улар кам ишлатилганлиги учун улар ҳақида тафсилотлар келтирмаймиз.

Мантиқий ифода

Мантиқий ифода — бу ёлғон ёки рост (ёки 0, ёки 1) қийматга эга бўлувчи ифодадир. Умуман ихтиёрий ифода “мантиқий” контекстда мантиқий ифода (масалан, `if – else` конструкциясини шартда) сифатида қаралади. Чунки, юқорида такидлаганимиздек, ихтиёрий 0 га тенг бўлмаган, бўш бўлмаган сатр ва х.к. катталик “рост” қиймат, қолган барча катталик “ёлғон” қиймат сифатида қаралади.

Мантиқий ифодалар учун ҳам мантиқий ўзгарувчиларни хусусиятлари ўринлидир. Мантиқий ифодалар `>`, `<`, (тенглик), `||` (мантиқий ИЛИ), `&&` (мантиқий И), `\` (мантиқий НЕ) ва бошқа операторлар ишлатилганда ҳосил бўлади. Масалан:

```
$a = 10<5; // $a=false  
$a = $b==1; // агар $b=1 бўлса $a=true  
$a = $b>=1&&$b<=10 // агар $b 1..10 ичида бўлса $a=true  
$a = !($b||$c)&&$d; // агар $b ва $c ёлғон бўлса, $a=true,  
$d эса – рост
```

У ёки бу мантиқий ўзгарувчини ростлиги қандай аниқланишини эслаб кўринг. Худди шундай шаклда мантиқий ифодани ростлиги аниқланади:

```
$b = $a>=1&&$a<=10; // $b га мантиқий ифода қийматини  
ўзлаштириш  
if($b) echo "a ўзгарувчи қиймати жорий диапазонда";
```

Сатрли ифодалар

PHP да сатрлар — асосий объектлардан биридир. Юқорида такидланганидек, сатрлар форматлаш символлари ва хатто бинар маълумотлар қатнашган текстдир. Қўштироноқ ёки апостроф ичига

олинган текст битта сатрда бошланади ва бошқа сатрда тугайди. Қуйида синтактик жихатдан тўғри бўлган мисол келтирилган:

```
$a="бу текст битта сатрдан бошланади  
Ва бошқасида давом этади,  
Учинчи сатрда ҳам давом этади ва х.к.";
```

Юқоридаги мисолларда қўлланилгани каби текст қўштирноқ ёки апостроф ичига олинади. Буларни бир – биридан фарқини тушунтириш вақти етиб келди.

Назорат саволлари

- 1 PHP нинг вазифаси
- 2 PHP да маълумотларнинг типлари ?
- 3 Асосий операторлар қайсилар ?
- 4 PHP да ифодаalar турлари ?

12-маъруза MY SQL да маълумотлар базаси .MY SQL да маълумотлар базаси Билан ишлаш тамоиллари

Режа 1 MY SQL да маълумотлар базаси хақида умумий маълумот

2 MY SQL да маълумотлар базаси дамаълумотларнинг типлари

3 MY SQL да маълумотлар базаси да жадваллар Билан ишлаш

MY SQL да маълумотлар базаси хақида умумий маълумот

Маълумотлар Базаси ва WWW

Маълумотлар базаси ута тез ривожланган ҳамда MY SQL ва M SQL ўзини кўрсатган соҳа, Internet учун дастурлар яратишдир. Internet учун мураккаб ва ишончли дастурларга еhtiёж ошган сари маълумотлар базасига еhtiёж ҳам ошиб бормокда. Сервер маълумотлар базаси Internetда кўп функцияларни қўллаши мумкин. Хар қандай веб- саҳифа маълумотлар базаси томонидан бошқарилиши мумкин.

Мисол тарикасида ўз катталогини WWW да еълон килмокчи бўлган ва Internet оркали буюртмалар кабул килмокчи бўлган катталог бўйича сотувчини кўрайлик.

Агар катталогни ХТМЛ-файлар шаклида еълон килинса Янги товар кўшилганда ёки нарх ўзгарганда кимдир катталогни тахрирлаши лозим бўлади Агар бунинг ўрнига катталог маълумотларини реляцион маълумотлар базасида сакланса катталогдаги ўзгаришларни маълумотлар базасидаги товар ёки нарх хакидаги маълумотларни ўзгартириш ёъли Билан реал вақт масштабида еълон килиш имконияти тугъилади.

Бундан ташкари катталогни мавжуд буюртмаларни кайта ишлаш электрон тизимлари Билан интеграция килиш имконияти тугъилади. Шундай килиб бундай веб-сайтни бошқариш учун маълумотлар базасидан фойдаланиш сотувчига хам олувчига хам кулайликлар тугъдиради.

Шу тарзда веб- сахифа маълумотлар базаси билан богъланади. Маълумотлар базаси сизни веб-серверингизда ёки сизни серверингиз маълумот алмашиши мумкин бўлган бошка машинада жойлашган бўлиши мумкин.(яхши МББТ бундай вазибаларни таксимлашни осон ташкил кила олади). Сиз ўзингизнинг веб-сахифангизга форма жойлаштирасиз ва фойдаланувчи узатиш керак бўлган сўров ёки маълумотни шу формага киритади. Формани серверга узатгандан сўнг, сервер сиз ёзган дастурни ишга туширади ва бу дастур фойдаланувчи узатган маълумотларни ажратиб олади. Бу дастурлар кўпинча СГИ-сценарийлар ёки Жава да сервер дастурлари шаклида яратилади, лекин дастурни ХТМЛ-сахифага тўгъридан тўгъри жойлаштириш хам мумкин.

Енди сизни дастурингиз фойдаланувчига кандай маълумотлар керак ва у маълумотлар базасига нима киритмокчилигини билади. Дастур маълумотларни танлаш ёки ўзгартириш учун СКЛ команда яратади, маълумотлар базаси бўлса колганини бажаради. Маълумотлар базасидан олинган натижаларни сизни дастурингиз Янги ХТМЛ-сахифа шаклига келтириб кайтадан фойдаланувчига юборади.

М SQL тарихи

То 1994 йилгача СКЛ кўлловчи РМББТ сотиб олиш учун анча кўп пул кетказишга тўгъри келар еди. Бозорда Орасле, Сйбасе ва Информих хокимлик килар еди.Бу маълумотлар базасини бошқариш тизимлари мураккаб богъланишларга ега бўлган катта хажмдаги маълумотларни кайта ишлаш учун мўлжалланган еди. Улар катта кувватга ва имкониятларга ега бўлиб, катта хисоблаш ресурсларини талаб килар едилар ва нархи киммат едилар. У пайтда \$2000 га 200-МХз Пентиум ли сервер сотиб олиш мумкин емас еди. Бундай МББТ учун керакли ресурслар ўнг минглаб доллар турар еди.

Катта корпорациялар ва йирик университетлар учун бундай серверлар комплектлари ва МББТ лар учун бир неча миллион доллар сарфлаш муаммо тугъдирмас еди.

Кичик ташкилотлар ва хусусий фойдаланувчилар ожиз кичик Амалий дастурлардан фойдаланишга мажбур едилар. Бир нечта арзон клиент/ сервер

архитектурали МББТ лар ўша пайтда мавжуд еди, лекин уларнинг ҳеч бири сўровлар тили сифатида СКЛ дан фойдаланмас еди. Энг кўзга кўринганларидан бири Ингрес коммерциал маълумотлар базаси билан битта аждодга ега бўлган Постгрес еди. Лекин бахтга карши коммерциал аналоглари каби ресурсларни талаб киларди ва СКЛ дан сўровлар тили сифатида фойдаланиш имкониятини бермас еди. Ўша пайтда Постгресда КУЕЛ тилининг кўриниши бўлган ПостКУЕЛ тилидан фойдаланарди.

Девид Хюз

Давид Хюз (Давид Хугхес) (яна Бамбй сифатида маълум) Австралияда Бонд Университетида ёзган диссертация бир қисми мониторинг тизимини яратиш ва тизимлар гуруҳини бир ёки бир неча жойдан бошқаришга бағъишланган еди. Лойиха Минерва Нетворк Манагемент Сйстем деб номланган еди. Минерва асосий элементи тармоқдаги ҳамма компютерлар хақидаги маълумотларни сакловчи маълумотлар базаси еди. Университет талабаси бўлгани ва катта коммерциал маълумотлар базалари ишлайдиган серверларга мурожаат қилиш имконияти ёёк бўлгани учун, Хюз унинг талабларига Постгрес – жавоб беради деган қарорга келди.

Унинг ҳамкасблари СКЛ тилидан Минерва учун стандарт сўровлар тили сифатида фойдаланишни таклиф қилишди. СКЛ га асосланган ҳолда Минерва дунёнинг СКЛ ни кўлловчи МББТ мавжуд ихтиёрий нуктасида кўлланиши мумкин еди. Бошқача қилиб айтганда Постгрес фойдаланувчилари билан чегаралаб қўйган ПостКУЕЛ га нисбатан СКЛ Минерва учун кенгрок фойдаланувчилар билан ишлашга имкон берар еди. Охири қелиб ҳатто Постгрес ҳам бугун СКЛ но кўллайди.

Бир томондан СКЛ стандартидан фойдаланиш истаги ва бошқа томондан СКЛни кўлловчи маълумотлар базасига мурожаат қилиш имконияти ёёклиги, Хюзни қийин ахволга солиб қўйди. Агар Минервада СКЛга асосланган сўровлар тилидан фойдаланилса, мос ишлаш механизмига ега МББТ топиб бўлмайди. Қиммат РМББТ сотиб олиш имкониятига ега бўлмаган Хюз масаланинг ажойиб ечимини топди: СКЛ сўровларни ПостКУЕЛ сўровларига трансляция қилувчи дастур яратиш. Бу дастур Минервага узатилган СКЛ сўровларни илиб олиши, ПостКУЕЛга айлантириши ва натижани Постгресга узатиши керак еди. Хюз шундай дастур яратди ва уни миниСКЛ ёки M SQL деб атади.

ПостКУЕЛ трансляторидан РМББТ га

Бир неча давр мобайнида бу конфигурация Хюзни каноатлантирар еди. Минерва учун агар СКЛни тушунса қандай МББТ дан фойдаланишнинг фарқи ёёк еди ва у Постгрес СКЛни тушунади деб ҳисоблар еди, чунки ўртада ПостКУЕЛга трансляция қилувчи M SQL жойлашган еди. Бахтга карши Минерва ўсиши билан унинг иши қийинлашиб борди. Аниқ бўлдики на Постгрес на бошқа катта РМББТ Минерва учун керак бўлган чекли ресурслар асосида кам сонли имкониятларни кўллай олмас еди. Масалан, Минерва учун бир вақтнинг ўзида бир неча маълумотлар базасига уланиш талаб қилинар еди. Бунинг учун Постгрес бир вақтнинг ўзида маълумотлар базаси сервери бир неча нусхасини

ишга туширишни талаб килар еди. Бундан ташкари бир неча потенциал лойиха катнашчилари унда катнаша олмас едилар, чунки Постгрес уларнинг тизимларини кўлламас еди, улар бўлса СКЛга асосланган киммат РМББТ сотиб олишга имконлари ёък еди.

Бу муаммоларга дуч келгандан сўнг Постгресга муносабатини ўзгартирди. Ўзини нг катталиги ва мураккаблиги билан Минерва талабларидан юкори еди. Минерва томонидан генерация килинадиган сўровлар асосан ИНСЕРТ, ДЕЛЕТЕ ва СЕЛЕКТ содда операторларидан иборат еди. Постгрес да мавжуд ва унумдорликни камайтирувчи колган хамма имкониятлар Минерва учун керак емас еди.

Хюзда СКЛга трансляцияни амалга оширувчи М SQL мавжуд еди. Унга талабларига жавоб берувчи маълумотлар базаси серверини яратиш учун маълумотлар омбори ва маълумотларни ажратиб олиш имкониятини кўшиш колган едию Бу еволюция бугунги кунда мавжуд М SQL га олиб келди.

MY SQL тарихи

MY SQL ни факат М SQL камчиликларига жавоб сифатида караш нотўғридир. Унинг ихтирочиси Майкл Видениус (яна Монгй сифатида маълум) швед компанияси ЦХ ходими маълумотлар базаси билан 1979 йилдан бери ишлайди. Якин пайтгача Видениус ЦХ да факат дастурчи еди. 1979 йилда фирма ичида фойдаланиш учун УНИРЕГ номли маълумотлар базасини бошқариш воситасини яратди. 1979 йилдан сўнг УНИРЕГ бир неча тилларда ёзилди ва катта маълумотлар базаларини кўллаш учун кенгайтирилди.

Битта дастур бажарилаётган хар бир жараён бу дастур нусхаси дейилади, чунки худди ўзгарувчи нусхаси каби хотирадан жой олади.

1994 йилда ЦХ WWW учун Амалий дастурлар ярата бошлади ва бу лойихани кўллашда УНИРЕГ дан фойдаланди. Бахтга карши, УНИРЕГ катта харажатлар талаб килгани учун, ундан веб-сахифаларни динамик генерация килиш учун муваффакиятли фойдаланиб бўлмади. Ва ЦХ шундан сўнг СКЛ ва М SQL га мурожаат килди Лекин ўша пайтда М SQL факат 1.x релизлари шаклида мавжуд еди. Юкорида айтганимиздек М SQL 1.x версиялари хеч кандай индексларни кўлламас еди ва шунинг учун УНИРЕГ дан унумдорлиги паст еди.

Видениус М SQL автори с Хюз билан богъланди ва М SQL ни УНИРЕГ даги В+ ИСАМ кайта ишловчисига улаш фикри Билан кизиктирмокчи бўлди. Лекин Хюз шу пайтга келдиб М SQL 2 яратиш ёълида анчага кетган ва индекслар Билан ишлаш воситаларини яратган еди. ЦХ ўз талабларига кўпрок мос келувчи маълумотлар базалари серверини яратишга карор килди.

ЦХ ходимлари Янгидан велосипед ихтиро килиб ўтирмадилар. Улар УНИРЕГ ни асос килиб олдилар ва сони ошиб бораётган ўзга дастурчилар утилиталаридан фойдаландилар. Ўз тизимлари учун бошида М SQL учун яратилган АПИ билан деярли устма-уст тушувчи АПИ яратдилар. Натижада Янги кенгрок имкониятга ега ЦХ маълумотлар базасига ўтмокчи бўлган М SQL фойдаланувчиси ўз кодига жуда кам ўзгартириш киритиши талаб килинарди. Шу билан бирга Янги маълумотлар базаси коди тўла оригинал еди.

1995 йил май ойига келиб ЦХ компания ички талабларини каноатлантирувчи маълумотлар базаси, - MY SQL 1.0 га ега еди. Фирма бизнес-партнери Детрон ХБ дан Давид Аксмарк (Давид Ахмарк) ЦХ га ўз серверини Internetда кўрсатишни таклиф килди. Серверни Internetда кўрсатишдан мақсад - биринчи бўлиб Аладдин Петер Дейч (Аладдин Петер Деуцч) кўллаган бизнес моделдан фойдаланишдир. Натижада MY SQLни M SQL га нисбатан «текинрок» килувчи ута мослашувчан авторлик ҳуқуқлари олинди.

Номига келганда Видениус бу хакида шундай дейди: «Хозиргача номаълум MY SQL номи каердан келиб чиккани. ЦХ да асосий катталог, ҳамда библиотекар ва утилиталар кўп қисми бир неча ўн йилдан бери «му» префиксига ега. Шу билан бирга менинг кизим (бир неча йилга кичик) исми ҳам Май (Мй). Шунинг учун халигача сир, бу икки манбанинг қайси бири MY SQL номини берган».

MY SQL ни Internetда еълон қилингандан бери у кўпгина УНИХ-тизимларга, Вин32 и ОС/2 га кўчирилди. ЦХ фикрича, MY SQL ни 500 000 га яқин серверлар ишлатади.

MY SQL да маълумотлар базаси дамаълумотларнинг типлари

Маълумотлар типлари.

MYSQL тилида қуйидаги асосий маълумотлар типлари ишлатилиб, уларнинг форматлари хар хил МББТ лар учун фарк қилиши мумкин:

- INTEGER - бутун сон (одатда 10 тагача қийматли ракам ва ишора).
- SMALLINT - "қиска бутун" (одатда 5 тагача қийматли ракам ва ишора).
- DECIMAL(p, q) - ўнли сон, P ракам ва ишорадан иборат ($0 < P < 16$). Ўнли нуктадан сўнг ракамлар сони к орқали берилади ($Q < P$, агар $q = 0$ бўлса, ташлаб юборилиши мумкин).
- FLOAT - хақиқий сон 15 та қийматли ракам ва бутун даражадан

иборат. Даража МББТ типи билан аникланади (масалан, 75 ёки 307).

CHAR(n) - узунлиги ўзгармас, n га тенг бўлган символли катор ($0 < n < 256$).

VARCHAR(n) - узунлиги ўзгарувчи, n символдан ошмаган символли катор ($N > 0$ ва хар хил МББТ ларда хар хил лекин 4096 дан кам емас).

DATE - махсус команда оркали аникланувчи форматдаги сана (Субсе да кўзда тутилган бўйича йй/мм/дд); сана майдонлари бизнинг ерамиздан олдин бир неча мингйилликлардан бошланувчи ва бизнинг ерамиз бешинчи- ўнинчи мингйиллиги билан чекланган хакикий саналарни ўз ичига олиши мумкин.

TIME - махсус команда оркали аникланувчи форматдаги вақт (кўзда тутилган бўйича hh.mm.ss).

DATETIME - сана ва вақт комбинацияси. (Sybase da **TIMESTAMP**).

MONEY - махсус команда оркали аникланувчи форматдаги пул. Формат ўз ичига пул бирлиги символи (\$, руб, ...) ва унинг жойлашуви (суффикс ёки префикс), каср қисм аниклиги ва пул қийматини кўрсатиш шартларини олади.

Жадваллар билан ишлаш

Жадвалларни яратиш.

Жадваллар **CREATE TABLE** командаси билан яратилади. Бу команда каторларсиз б ўш жадвал яратади. **CREATE TABLE** командаси жадвал номини ва жадвал ўзини маълум тартибда к ўрсатилган устунлар номлари кетма – кетлиги таърифи к ўринишида аниклайди. У маълумотлар типлари ва устунлар ўлчовини аниклайди. **ХАР** бир жадвал жуда б ўлмаганда битта устунга ега б ўлиши керак.

CREATE TABLE командаси синтаксиси:

CREATE TABLE <табле-наме >

(<column name> <data type>[(<size>)],

<column name> <data type>[(<size>)], ...);

Аргумент киймати катталиги маълумот турига богъликдир. Агар сиз махсус к ўрсатмасангиз, тизим автоматик кийматни ўрнатади.

Бундан буён куйида келтирилган 3 та жадвалдан иборат маълумотлар базасини к ўрамиз.

Жадвал Сотувчилар (Salepeople):

Snum	Sname	City	Comm
11	Peel	London	0.12
12	Serres	San Jose	0.13
14	Motika	London	0.11

SNum – ХАР бир сотувчи уникал номери,

SName – сотувчи номи,

Ситй – сотувчи адреси (шаҳар),

Сомм – сотувчиларнинг ўнли шаклдаги комисион фойдаси.

Жадвал Буюртмачилар (Customers):

Cnum	Cname	City	Rating	SNum
21	Hoffman	London	100	11
22	Giovanni	Rome	200	13
23	Liu	SanJose	200	12

CNum – ХАР бир буюртмачи уникал номери,

CName – буюртмачи номи,

Ситй – буюртмачи адреси (шаҳар),

Ратинг – буюртмачининг бошқалардан устунлик даражасини к ўрсатувчи код

(рейтинг),

SNum – шу буюртмачига тайинланган сотувчи номери.

Жадвал Буюртма (Orders):

CNum	AMT	ODate	SNum
------	-----	-------	------

CNum m

38	4723.00	1990/10/0	26	11
	5			
310	1309.95	1990/10/0	24	12
	6			

CNum – XAP бир сотиб олиш уникал номери,
 CName– сотиб олиш суммаси киймати,
 ODate – сотиб олиш санаси,
 CNum – сотиб олувчи буюртмачи номери,
 CNum – сотувчининг номери.

Мисол учун сотувчилар жадвалини яратишни к ўриб чикамиз:

CREATE TABLE Salepeople

(SNum интегер,
 SName чар (10),
 City чар (10),
 Comm decimal();

Жадвалларни ўчириш.

Жадвални ўчириш имконига ега б ўлиш учун, жадвал егаси (Яъни яратувчиси) б ўлишингиз керак. Факат б ўш жадвални ўчириш мумкин. Каторларга ега б ўлган, т ўлдирилган жадвални ўчириш мумкин емас, Яъни жадвал ўчиришдан олдин тозаланган б ўлиши керак. Жадвални ўчириш командаси куйидаги к ўринишга ега:

DROP TABLE < < table name >;

Масалан: **DROP TABLE Salepeople;**

Жадвални яратилгандан с ўнг ўзгартириш.

Жадвални ўзгартириш учун ALTER TABLE командасидан фойдаланилади. Бу команда жадвалга Янги устунлар к ўшиш, устунларни ўчириш, устунлар катталигини ўзгартириш, камда чекланишларни к ўшиш ва олиб ташлаш имкониятларига ега. Бу команда АНСИ стандарти кисми емас, шунинг учун XAP хил тизимларда XAP хил имкониятларга ега.

Жадвалга устун к ўшиш учун команданинг типик синтаксиси:

**ALTER TABLE <табле наме> ADD <column name>
 <data type> <size>;**

Масалан:

ALTER TABLE Salepeople ADD Пконе ЧАР(7);1));

Назорат саволлари

- 1 MY SQL нинг имкониятлари ?
- 2 MYSQL да кандай типла ишлатилади ?
- 3 MYSQL нинг асосий функциялари?

Адабтётлар

1. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация, сопровождение. Теория и практика, 2-е изд. : Пер. с англ. : Уч. пос. – М.: Изд. дом "Вильямс", 2000. – 1120 с.
2. Роб П. Системы баз данных: проектирование, реализация и управление (5-е издание) издательство "БХВ - Санкт-Петербург" ·1200 стр, 2003 г. ·.
3. Попов И. И., Максимов Н. В., Голицына О. Л. Базы данных. издательство "Форум" · 352 стр ,2004 г. ·
4. Диго С.М. Базы данных Проектирование и использование . издательство "Финансы и статистика" · 592 стр, 2005 г.
5. Деут К. Введение в системы баз данных 8-изд. "Вильямс" ·1328 стр, 2005 г.
6. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.М. 1998
7. Астахова И.Ф., Толстобров А.П. SQL в примерах и задачах. Учебное пособие. Новое знание, 176 стр, 2002 г.

8. Полякова. Л.Н. Основы SQL. Курс лекции. Учебное пособие. издательство "ИНТУИТ.РУ" · 368 стр, 2004 г. ·
9. Бен Форта Освоу самостоятельно SQL. 10 минут на урок (3-е издание) издательство "Вильямс" · 288 стр, 2005 г. ·
10. Аллен К. 101 Oracle PL/SQL. Издательство: ЛОРИ. 2001.
11. Бьюли А., Мишра С. Секреты Oracle SQL. Издательство: Символ -Плюс , 368 стр., 2003.
12. Максим Кузнецов, Игорь Симдянов, Сергеу Голышев. PHP 5 на примерах. Серия: На примерах. Издательство: БХВ-Петербург, 2005 г., 576 стр.
13. Леон Аткинсон., Зеев Сураски PHP 5. Библиотека профессионала., Core PHP Programming., Серия: Библиотека профессионала., Издательство: Вильямс, 2005 г., 944 стр.
14. Дмитриу Котеров, Алексеу Костарев. PHP 5., Серия: В подлиннике Издательство: БХВ-Петербург, 2005 г., 1120 стр.
15. Максим Кузнецов, Игорь Симдянов, Сергеу Голышев. PHP 5. Практика создания Web-саутов (+ CD-ROM). Серия: Про. Профессиональное программирование., Издательство: БХВ-Петербург, 2005 г., 948 стр.
16. Дмитриу Котеров, Алексеу Костарев., PHP 5. Наиболее полное руководство Серия: В подлиннике Издательство: БХВ-Петербург, 2005 г., 1120 стр.
17. Анатолиу Мотев Уроки MySQL (+CD-ROM) Серия: Самоучитель Издательство: БХВ-Петербург, 2006 г., 208 стр.