

**ЎЗБЕКИСТОН АЛОҚА ВА  
АХБОРОТЛАШТИРИШ АГЕНТЛИГИ**

**ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ  
УРГАНЧ ФИЛИАЛИ**

**«АХБОРОТ ТЕХНОЛОГИЯЛАРИ»  
КАФЕДРАСИ**

**“WEB ДАСТУРЛАШ” ФАНИДАН  
ЛАБАРАТОРИЯ МАШҒУЛОТЛАРИНИ БАЖАРИШГА  
УСЛУБИЙ КЎРСАТМА**

УРГАНЧ-2009

Ушбу услубий кўрсатма Интернетда дастурлаш масалаларини ечишда шунингдек WEB сайтлар яратиш, маълумотлар базалалари билан ишлаш масалаларини қамраб олади Бу фанни ўқитишдан мақсад талабаларга таълим стандартида талаб қилинган билимлар, кўникмалар ва тажрибалар даражасини таъминлашдир.

<b>Тузувчилар:</b>	
Жафаров С.К.	Ахборот технологиялари кафедраси ўқитувчиси

<b>Такризчилар:</b>	

# Лаборатория иши №1

Мавзу: HTML гиперматнли тили ёрдамида Web-саҳифа яратиш

Ишдан мақсад: HTML гиперматнли тили ёрдамида Web-саҳифа яратишни ўрганиш

## НАЗАРИЙ ҚИСМ

Интернет - сайёрамиз миқёсидаги компьютер тармоқларининг бирлашмасидир. Агар космик станция аъзоларининг E-mail дан фойдаланишини инобатга олсак интернетнинг сайёрамиз сарҳадларидан чиққанини ҳам кўришимиз мумкин.

Бутун дунё ўргимчак тўри WWW (World Wide Web) - бу кўп сонли ўзаро бир-бирига боғланган ҳужжатлардир. Web саҳифада бошқа Web саҳифалар билан боғловчи чекли гипермуружаат бўлади. Гипермуружаат механизмини бир неча сўз билан изоҳлаш қийин, лекин камида бир марта Web саҳифага кириб ишлаган одам буни дарҳол тушунади.

Гипермуружаат бутунжаҳон ўргимчак тўридан ҳужжатнинг URL деб номланувчи манзилини кидиришда ишлатилади. Биз биламизки Web саҳифанинг асосий моҳияти ахборотни тасвирлаш ва уни ҳар бир фойдаланувчи учун очиб беришдир. Бунда бир қанча функционал чекланишлар мавжуд, яъни биз олдиндан Web саҳифани кўрувчининг компютери қандай, мониторинг имконияти қандай ва кўриш ойнасининг қандай ўлчамда ўрнатилганлигини билмаймиз. Биз ҳаттоки фойдаланувчининг қандай операцион муҳитда ишлаётганини ҳам билмаймиз. Web саҳифа эса INTEL машинада ҳам Макентошда ҳам бир хил кўринишда бўлиши керак. Фойдаланувчи ўз операцион муҳитида қандай ёзув шрифтлар ўрнатганлиги ёки унинг видеокартаси қанча рангни кўтара олиши ҳам маълум эмас. Юқоридаги маълумотларнинг йўқлиги ёки етарли эмаслиги умуман ахборот алмашиш тилининг пайдо бўлишига тўсиқ бўлиши керак эди, лекин бундай бўлмади.

Гап шундаки 1986 йили халқаро стандартлаштириш ташкилоти (ISO) томонидан қоғоздаги ҳужжатни экранда тасвирлаш имконини берувчи ҳамма вариантларни эътиборга олган SGML (Standart Generalized Markup Language) тили яратилди. Ҳамма имкониятларни эътиборга олган кучли система ишлаб чиқилди.

Web саҳифа учун бу яхши йўл деб ҳисобланган эди. Лекин бу тил қонун қоидалари изоҳи юзлаб саҳифаларни эгаллади. Бундай тил асосида яратилган саҳифани тасвирловчи дастурни ишлаб чиқиш учун жуда кўп вақт кетган бўларди, шунинг учун Интернет эҳтиёжи учун SGML тилининг маълум бир қисми танлаб олинди ва HTML (Hyper Text Markup Language) деб номланди. HTML

тилида ёзилган Web саҳифа файлида ахборотнинг қандай тасвирланиши ҳақидаги йўл йўриқлар кетма – кет жойлашган бўлади. Бу файл оддий матнли файл бўлиб уни махсус браузер датстурларсиз ўқиш қийин. График расмларни эса умумман кўриб бўлмайди чунки унинг ўрнида браузер дастурга бирор расм кераклигини кўрсатувчи ТЭГ туради. Агар биз ўз Web саҳифамизни яратмоқчи бўлсак, албатта HTML тилини яхши билишимиз керак бўлади.

HTML нинг конструкцияси ТЭГ лар дейилади. Браузер ТЭГ ларни оддий матнлардан фарқлаши учун улар бурчак қавсларга олинадилар. ТЭГ тасвирлаш жараёни ҳатти ҳаракатларининг бошланишини билдиради. Агар бу ҳаракат бутун ҳужжатга таллуқли бўлса, бундай тэг ўзининг ёпилувчи жуфтига эга бўлмайди. Жуфт тэглarning иккинчиси биринчисининг ҳаракатини яқунлайди. Масалан, ҳар бир Web саҳифа `<html>` тэги билан бошланиб `</html>` тэги билан ёпилиши керак. Эътибор берган бўлсангиз ёпилувчи тэг очилувчидан « / » белгиси билан фарқ қилади. Тэг номлари катта ёки кичик ҳарфлар билан ёзилиши мумкин, буларни браузер бир хил қабул қилади. HTML тилида бошқа компьютер тилларидаги каби изоҳ бериш имконияти мавжуд. Изоҳ қуйидаги «<- ->» ва «- ->» белгилар орсига ёзилади.

Масалан:

```
<-- Бу изоҳ -->
```

Ҳар қандай Web саҳифа иккита қисмдан ташкил топади. Булар сарлавҳа қисми ва асосий қисм. Сарлавҳа қисмида Web саҳифа ҳақидаги маълумот жойлашади, асосий қисмда эса Web саҳифанинг мазмуни билан тасвирланиш қоидалари жойлашади. Сарлавҳа қисми қуйидаги очилувчи `<head>` ва ёпилувчи `</head>` тэглари орасида жойлашади. Асосий қисм эса `<body>` ва `</body>` тэглари орасида жойлашади. Одатда сарлавҳа қисми олдидан қўлланилаётган HTML стандартлари ҳақида маълумот ёзилади. Ҳар қандай Web саҳифанинг умумий кўриниши қуйидагича бўлади:

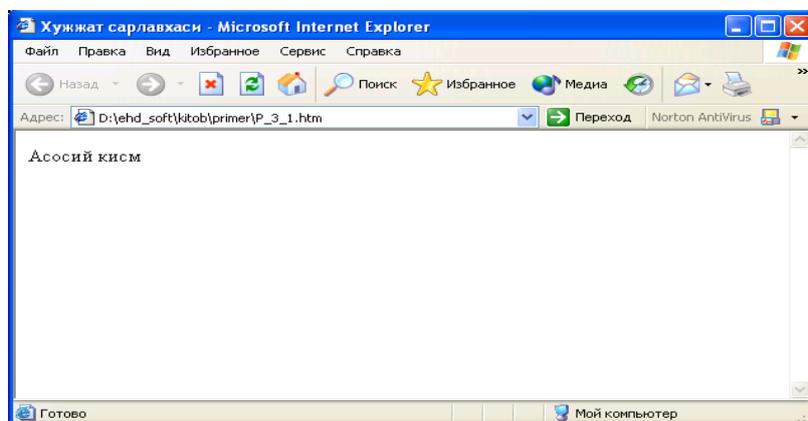
### **Лаборатория ишини бажариш тартиби**

**Қуйидаги кодни ёзиб лаб\_1.html да саклаймиз**

**Қарабсизки сизда HTML ҳужжатнинг асосий тузилиши пайдо булади**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
```

```
<html>
  <head>
    <title>
      Хужжат сарлавҳаси
    </title>
  </head>
  <body>
    Асосий қисм
  </body>
</html>
```



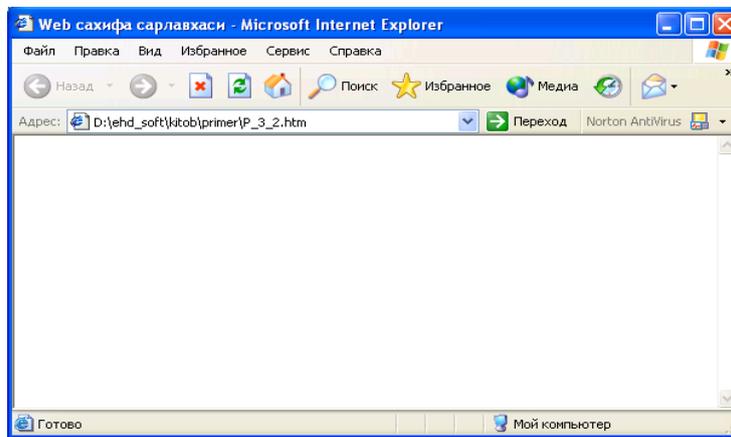
Расм 3\_1.

Биринчи *<!DOCTYPE>* тэги ўзининг параметрлари билан браузерга ушбу Вебсаҳифани қайси HTML версияда ёзилганлиги ҳақида маълумот беради.

Web саҳифа ишга тушурилганда браузернинг энг юқори сатрида юкланаётган ҳужжат мазмунини англатувчи қисқача ёзув туради. Бу ёзувни ҳосил қилиш учун қуйидаги очилувчи *<title>* ва ёпилувчи *</title>* тэгларидан фодаланамиз.

Мисол 3\_2:

```
<html>
<head>
  <title>Web саҳифа сарлавҳаси</title>
</head>
<body>
</body>
</html>
```



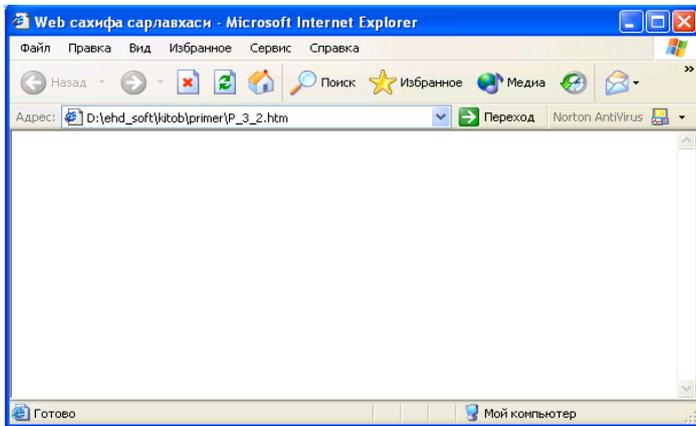
Расм 3\_2.

Web sahifaning asosiy qismi *<body>* va *</body>* taglari orasida joylashadi. Bu oddiy matn bulishi mumkin. Brauzer bu matnни туғридан туғри интерпретация қилиб экранда тасвирлайди. Бизга дастлабки Web sahifamizни яратиш учун oddiy «Блокнот» matn муҳаррири кифоя. Қуйида кўрсатилган мисолни matn муҳарририда ёзиб, уни хотирага ёзишда кенгайтмасини html ёки htm деб киритишимиз керак.

Мисол 3\_3:

```
<html>
  <head>
    <title>Менинг биринчи Web sahifam </title>
  </head>
  <body>
    Менинг бу sahifamга кирувчиларга алангали салом
  </body>
</html>
```

Бу файлни ишга тушириш учун сичқонча кўрсаткичини шу файл устига келтириб чап тугмасини икки марта босиш керак. Натижада экранда қуйдаги кўринишдаги натижа ҳосил бўлади:



Расм 3.3

`<body>` тэги бир қанча қўшимча параметрларга эга. Бу параметрлар тэгнинг очилувчи қисмида жойлашади. Параметрлар икки қисмдан иборат бўлади: параметр номи ва параметр қиймати. Масалан *bgcolor* параметри тасвирланаётган Web саҳифа фонининг рангини белгилайди.

Масалан: `<body bgcolor = "green">`

Параметрларнинг сатрли қийматлари қўштирноқ ичида ёзилади. Биз қуйида `<body>` тэгининг параметрлари билан танишамиз.

- *Background* - фон сифатида бирор бир график тасвирдан фойдаланиш. Параметр қиймати сифатида график тасвир жойлашган манзил (URL) берилади.
- *Text* - тасвирланаётган матн ранги.
- *Link* - Web саҳифадаги матнли гипермуружат ранги.
- *Vlink*-фойдаланувчи томонидан олдин муружат қилинган гипермуружаат ранги.
- *Alink* - фойдаланувчи томонидан танланган гипермуружаат ранги.
- *Lang* – Web саҳифа матни ёзилган тилни аниқлаш.

### 3.2. Метамаълумотлар

Энди биз метамаълумотлар билан танишиб чиқамиз. Web саҳифаларда мета маълумотларини ҳосил қилиш учун `<meta>` тэги ишлатилади, унинг умумий кўриниши қуйидагича:

```
<meta name="ўзгаручи номи" content="ўзгарувчи қиймати">
```

Агарда биз Web саҳифадаги **автор** ҳақида маълумот ёзмокчи бўсак уни қуйидаги кўринишда ёзиш мумкин:

```
<meta name="Auther" content="Бу менику!!!">
```

Мета маълумотлар асосан Internet да жойлашган қидириш машиналари учун зарур. Қидириш машиналари Web саҳифалар ҳақидаги маълумотларни қаердан олади деган савол пайдо бўлади. Худди шу маълумотларини қидириш машиналари метаўзгарувчилардан олади. (Web саҳифа қайси соҳага тегишли, унда қандай маълумотлар борлиги) *<meta>* тэгида *keymards* ва *description* ўзгарувчиси бор. *Keymards* ўзгарувчили Web саҳифадаги калитли сўзлар рўйхатини ўзида сақлайди. *description* ўзгарувчи эса Web саҳифанинг қисқача маълумотини ўзида сақлайди. Масалан, бизнинг Web саҳифамиз компьютер вируслари ҳақида тайёрланган бўлсин у ҳолда HTML ҳужжатда мета маълумотларни қуйидагича ёзиш мумкин:

Мисол 3\_4.

```
<html>
  <head>
    <title>компьютер вируслари </title>
    <meta name="keywords" lang="ru" content=" вирус,
      компьютер, антивирус,...">
    <meta name="description" content="Web саҳифадаги компьютер
      вирусларига бағишланган.
  </head>
  <body>
    компьютер вируслари бу ....
  </body>
</html>
```



Расм 3\_4.

Браузерлар фойдаланувчи томонидан очиб кўрилган Web саҳифаларини кэш хотирада сақлаб қолади. Агар фойдаланувчи яна шу саҳифаларга мурожаат қилса, Web браузер олдиндан кэш хотирада мавжуд бўлган (яна янги саҳифани интернетдан олмасдан) нусхасини олиб кўрсатади. Бу жараён фойдаланувчининг вақтини ва иқтисодини тежайди. Энди Web саҳифа қачон янгиланади деган савол пайдо бўлади. Бу саволга метамаълумотлардаги *expres* ўзгарувчилардан жавоб олиш мумкин. Бу ўзгарувчида Web саҳифанинг яроқлилик муддати кўрсатилади. Агар кэш хотирадаги web саҳифа яроқлилик муддати ўтган бўлса, браузер тармоқдан Web саҳифани қайтадан ўқиб олади.

Мисол:

```
<meta http-equiv="Expres" content="Tue,uf Aug 2002 14:56:27 Gmt">
```

Web саҳифаларда маълумотлар тез ўзгариши мумкин, масалан чатларда ва биржа саҳифаларида маълумотлар тез ўзгаради. Бундай ҳолларда *refresh* ўзгарувчисидан фойдаланилади ва қийматлари секундларда берилади.

Масалан:

```
<meta http-equiv="Refresh" content=10>
```

Бу ёзувдан кейин Web саҳифа ҳар 10 секунддан кейин автоматик тарзда ўзи қайта юкланади.

**Лаборатория ишни бажариш учун қуйидаги  
html тилининг асосий тэглари билишингиз лозим**

### **Асосий тэглар.**

`<html></html>` Web саҳифани кўрувчи дастурга бу ҳужжатнинг HTML ҳужжат эканлигини кўрсатади.

`<head></head>` Ҳужжатда тасвирланмайдиган ҳар хил маълумотларнинг жойлашиш ўрнини кўрсатади. Бу ерда ҳужжатнинг номланиш тэги ва кидирув машиналари учун тэг жойлашади. Бундан ташқари JavaScript функциялари ҳам шу ерда жойлашади.

`<body></body>` ҳужжатнинг кўринувчи яъни асосий қисмини билдиради.

### **Сарлавҳа тэги.**

`<title></title>` браузер сарлавҳасига ҳужжат номини жойлаштиради. **Ҳужжат асосий қисмининг атрибутлари.**

`<body bgcolor=?>` RRGGBB кўринишида ранг қийматини қўллаб ҳужжат фони рангини ўрнатади. Мисол: FF0000 – қизил ранг.

`<body text=?>` Ранг қийматини RRGGBB кўринишида қўллаб ҳужжатдаги матн рангини ўрнатади. Мисол: 000000 – қора ранг.

`<body link=?>` Ранг қийматини RRGGBB кўринишида қўллаб ҳужжатдаги гипермуружаат рангини ўрнатади. Мисол: 00FF00 - яшил ранг.

`<body vlink=?>` Ранг қийматини RRGGBB кўринишида қўллаб ҳужжатдаги фойдаланилган гипермуружаат рангини ўрнатади.

Мисол: 333333 – кулранг ранг.

`<body alink=?>` Гипермуружаатнинг босилаётгандаги рангини ўрнатади.

### **Матнни форматлашда ишлатиладиган тэглар.**

`<pre></pre>` Олдиндан форматланган матнни тасвирлайди.

`<h1></h1>` ЭНГ КАТТА сарлавҳа ҳосил қилади.

`<h6></h6>` энг кичик сарлавҳа ҳосил қилади.

`<b></b>` қалин шрифт ҳосил қилади.

`<i></i>` қия шрифт ҳосил қилади.

`<tt></tt>` Ёзув машинкасиникига ўхшаш шрифт ҳосил қилади.

`<cite></cite>` цитата учун ишлатиладиган шрифт ҳосил қилади. Одатда қия

ёзилган шрифт ишлатилади.

`<em></em>` матндаги сўзни алоҳида ажратиб кўрсатади. (қия ёки қалин матн)

`<strong></strong>` матннинг муҳим қисмларини белгилашда ишлатилади. (қия ёки қалин матн).

`<font size=?></font>` 0 дан 7 гача бўлган ораликда матн ўлчамини ўрнатади.

`<font color=?></font>` Ранг қийматини RRGGBB кўринишида қўллаб ҳужжатдаги матн рангини ўрнатади.

### Гипермуурожаатлар.

`<a href="URL"></a>` Бошқа бир Web ҳужжатга ёки жорий ҳужжатнинг бирор бир қисмига гипермуурожаат ҳосил қилади.

`<a href="mailto:EMAIL"></a>` Электрон хат ёзиш учун ишлатиладиган дастурга муурожаатни ҳосил қилади.

`<a name="NAME"></a>` Ҳужжатдаги матн қисмини гипермуурожаат қилиш мақсадида белгилайди.

`<a href="#NAME"></a>` Жорий ҳужжатнинг қисмига гипермуурожаат ҳосил қилиш.

### Форматлаш.

`<p>` янги абзац ҳосил қилади.

`<p align=?>` *left*, *right*, ёки *center* қийматларни қўллаб абзацни текислаш.

`<br>` янги сатрдан бошлаш.

`<blockquote></blockquote>` Матннинг иккала томонидан бўш жой ажратади.

`<dl></dl>` аниқловчи рўйхат ҳосил қилади.

`<dt>` рўйхатдаги ҳар бир терминни аниқлайди.

`<dd>` рўйхат бандига изоҳ бериш.

`<ol></ol>` рақамли рўйхат ҳосил қилади.

`<li>` рўйхатдаги ҳар бир элементни аниқлайди ва тартиб рақам беради.

`<ul></ul>` рақамланмаган рўйхат беради.

`<li>` Рўйхатдаги ҳар бир элементни аниқлайди.

`<div align=?>` HTML ҳужжатларда матнли блоklarни форматлаш.

## График элементлар.

`` HTML хужжатга график тасвир қўшиш.

`` *left, right, center; bottom, top, middle* қийматларини қўллаб тасвирни хужжатнинг бирор томонига текислаш.

`` Тасвир атрофи рамкаси қалинлигини ўрнатади.

`<hr>` HTML хужжатга горизонтал чизик қўшиш.

`<hr size=?>` Чизикнинг қалинлигини ўрнатиш.

`<hr width=?>` Чизик энини пиксел ёки процентларда ўрнатиш.

`<hr noshade>` Чизик соясини йўқотиш.

`<hr color=?>` Чизикга маълум бир ранг бериш. Қиймати RRGGBB.

## Жадваллар.

`<table></table>` жадвал ҳосил қилиш.

`<tr></tr>` жадвалдаги сатрларни аниқлаш.

`<td></td>` жадвалдаги алоҳида ячейкани аниқлаш.

`<th></th>` жадвал сарлавҳасини аниқлаш.

### Жадвал атрибутлари

`<table border=#>` Жадвал рамкаси қалинлигини бериш.

`<table cellspacing=#>` Жадвал ячейкалари орасидаги масофани бериш.

`<table cellpadding=#>` жадвал қийматлари ва рамкаси орасидаги масофани бериш.

`<table width=#>` жадвал энини пикселларда ёки хужжат энига нисбатан фоизларда бериш.

`<tr align=?>` ёки `<td align=?>` *left, center, ёки right* қийматларини қўллаб жадвалда ячейкаларни текислаш.

`<tr valign=?>` ёки `<td valign=?>` *top, middle, ёки bottom* қийматларини қўллаб жадвал ячейкаларини вертикал бўйича текислаш.

`<td colspan=#>` Битта ячейкага бирлашган устунлар сонини кўрсатиш. (жимлик

бўйича =1)

`<td rowspan=#>` Битта ячейкага бирлашган сатрлар сонини кўрсатиш. (жимлик бўйича =1)

`<td nowrap>` браузерга жадвал ячейкасидаги сатрни кўчиришни тақиқлайди.

### Фреймлар.

`<frameset></frameset>` Ҳужжатда фрейм яратиш.

`<frameset rows="value,value">` Фреймнинг горизонтал бўйича ўлчамлар нисбати.

`<frameset cols="value,value">` Фреймнинг вертикал бўйича ўлчамлар нисбати.

`<frame>` Фреймда ҳосил қилинадиган объектни аниқлайди.

`<noframes></noframes>` Бу тэглар орасида ёзилган маълумот браузер томонидан фрейм технологияси тушунилмаган ҳолатда экранда пайдо бўлади.

### Фрейм атрибутлари.

`<frame src="URL">` Фреймда тасвирланиши керак бўлган HTML ҳужжатни аниқлаш.

`<frame name="name">` Фрейм номини аниқлаш.

`<frame marginwidth=#>` фреймнинг ўнг ва чап томонларидан бўш жой қолдириш.

`<frame marginheight=#>` фреймнинг юқори ва қуйи томонларидан бўш жой қолдириш.

`<frame scrolling=VALUE>` фойдаланучига фреймдаги ўтказиш йўлакчаси (линейка прокрутки)ни бошқариш имконини бериш.

`<frame noresize>` фрейм ўлчамларининг ўзгармаслигини таъминлаш.

### Формалар.

`<form></form>` форма ҳосил қилиш.

`<select multiple name="NAME" size=?></select>` Йўлакчали меню яратиш.

`<option>` менюнинг ҳар бир алоҳида элементини кўрсатиш.

`<select name="NAME"></select>` Йўлакчасиз меню яратиш.

`<textarea name="NAME" cols=40 rows=8></textarea>` Матн киритиш учун кўп сатрли майдон яратиш

`<input type="checkbox" name="NAME">` Байроқчалар (*checkboxes*) ҳосил қилиш.

`<input type="radio" name="NAME" value="x">` Улагич (*radio*) ҳосил қилиш.

`<input type="text" name="foo" size=20>` матн киритиш учун сатр ҳосил қилиш.

`<input type="submit" value="NAME">` "Қабул қилиш" тугмасини ҳосил қилиш.

`<input type="image" border=0 name="NAME" src="name.gif">` "Қабул қилиш" тугмаси учун тасвирдан фойдаланиш.

`<input type="reset">` "Бекор қилиш" тугмасини ҳосил қилиш.

## 1-Лаборатория ишининг топшириқлари ҳар бир талаба битта фаннинг электрон қўлланмасини яратиш

### Variantlar

1	Интернет технологиялари
2	Йуналишга кириш
3	Информатика
4	Компьютер графикаси ва дизайни
5	Дастурлаш технологиялари
6	Маълумотлар базалари
7	Математик моделлаштириш
8	Internet мухитида дастурлаш
9	Компьютер тизимлари ва тармоклари
10	Объектга йуналтирилган дастурлаш тиллари
11	Интеллектуал тизимлар
12	Инсон машина интерфейси
13	Автоматлаштирилган системаларни куриш асослари
14	Компьютер тармоклари ва тармок ресурсларнинг архитектураси
15	Информацион хавфсизлик
16	Информацион тизимлар
17	Маълумотлар структураси ва алгоритмлар
18	Математик дастурлаш ва оптималлаш усуллари

19	Тизимли дастурий таъминот
20	Хисоблаш тизимларини ахборот асослари
21	Электрон дизайн
22	Операцион тизимлар ва офис иловалари
23	Телекоммуникация тизимлари ва тармоклари
24	TCP/IP тармоқлар, IP технологиялар
25	Корпоратив тармоқлар, маъмуриятлаштириш масаласи, узаро фоалият
26	Тармоқларнинг ташкили ва хавфсизлиги
27	WEB dasturlash
28	Java дастурлаш тили
29	PHP tilida dasturlash
30	

### Назорат саволлари

- 1) WWW нима ?
- 2) HTML нима ?
- 3) HTML нинг асосий структураси айтиб беринг ?
- 4) HTML нинг асосий тег лари қайсилари ?
- 5) Web сайтни яратиш воситалари ?

Мавзу: HTMLда формалар.

Ишдан мақсад: HTMLда формалар яратишни ўрганиш.

## НАЗАРИЙ ҚИСМ

HTML формалар, фойдаланувчилар томонидан, маълумотларни киритиш ва киритилган маълумотлардан кейинчалик фойдаланиш мақсадида ташкил қилинган. Формалар – матнли ойналар, тугмалар, байроқчалар, улагич(переключател)лар ва менюлар каби бошқариладиган элементлардан ташкил топиши мумкин. Формалар `<form>` ва `</form>` теглари орасида жойлашади.

Баъзи ҳолларда фойдаланувчи билан браузер ўртасида актив мулоқат қилишга тўғри келади, яъни маълумотларни киритишга ва таҳрир қилишга тўғри келади. Бундай ҳолларда биз юқорида айтганимиздек HTML ҳужжатнинг формалар деб аталувчи бўлимидан фойдаланамиз. Фойдаланувчи юқорида кўрсатилган формаларни маълумотлар билан тўлдиради ва уни қайта ишлаш учун серверга жўнатади. Сервер бу одатда Web-сервер ёки электрон почта сервери бўлиши мумкин.

Шундай қилиб ҳар қандай форманинг асосий қисми бўлиб унинг бошқариладиган элементлари ҳисобланади. Бу элементларнинг ҳар бири ўз номига эга ва бундан ташқари бу элементлар ўзининг бошланғич қийматига ҳамда жорий қийматига эга.

Элементнинг бошланғич қиймати унинг биринчи ҳосил қилинишида ёки унинг қайтадан ўрнатилишида (инициализация) экранда тасвирланади. Кейинчалик фойдаланувчи бу қийматларни ўзгартириши мумкин. Фойдаланувчи томонидан тўлдирилган форманинг қийматлари «ном=қиймат» жуфтлик кўринишида қайта ишланиш учун жўнатилади.

HTML формалар қуйидаги бошқариладиган элементлардан фойдаланади:

### **Тугмалар (*buttons*)**

Учта турдаги тагмалар мавжуд:

- Жўнатувчи тугмалар – бу тугмалар ёрдамида формалар серверга қайта ишлаш учун жўнатилади.

- Инкор этувчи тугмалар – бу тугма босилса форманинг ҳамма қийматлари қайтадан бошланғич ҳолатига ўрнатилади.
- Умумий тугмалар – бундай турдаги тугмани босганда шу тугмага боғланган процедура ишга тушади.

### **Байроқчалар (*checkboxes*)**

Байроқча – бу иккита (танланган ёки танланмаган) ҳолатга эга бўладиган элемент. Байроқчалар *input* элементи ёрдамида ҳосил қилинади.

### **Улагичлар (*radiobuttons*)**

Улагичлар ҳам байроқчаларга ўхшайди, аммо битта фарқи шундаки ҳамма элементлари битта номга эга бўлади ва шулардан биттаси танланади. Улагичлар ҳам *input* ёрдамида ташкил қилинади.

### **Менюлар (*menus*)**

Меню фойдаланувчига бир нечта вариантлардан бир ёки бир нечтасини танлаш имкониятини яратади. Меню *select* элементи ёрдамида *optgroup* ва *option* лар билан биргаликда ташкил қилинади.

### **Матн киритиш майдони (*text input*)**

Матнни фойдаланувчи томонидан киритилишини амалга оширувчи иккита элемент мавжуд. *input* элементи битта қаторли киритишни таъминлайди. *textarea* – кўпқаторли киритишни амалга ошириш учун ишлатилади.

### **Файл селектори (*file select*)**

Бу элемент ёрдамида биз серверга юборадиган файлни танлаймиз. Файл селектори ҳам *input* элементи ёрдамида амалга оширилади.

#### **<form> </form> тэглари**

Формалар ташкил қилиш *form* элементи ёрдамида амалга оширилади ва биз унинг умумий кўриниши билан танишиб чиқамиз.

Умумий кўриниши:

**Синтаксис :** <FORM> ... </FORM> (Блокли элемент)

**Атрибутлар:** *id, class, style, title, lang, dir, ходоуца*

*action* = URL (формаларни қайта ишлаш учун кўрсатилган URL манзил)

*method* = GET / POST (формалардаги маълумотларни жўнатиш учун HTTP методи)

*enctype* = файл тип(формага мурожат қилиш учун)

*accept* = файллар типлари (бўлиши мумкин бўлган файллар типлари)

*accept-charset* = кодлаштиришлар (бўлиши мумкин бўлган символларни кодлаш усуллари)

*name* = CDATA (сценария учун форма номи)

*target* = фрейм (натижаларни ҳосил қилиш учун фрейм)

*onsubmit* = сценария (формани жўнатиш)

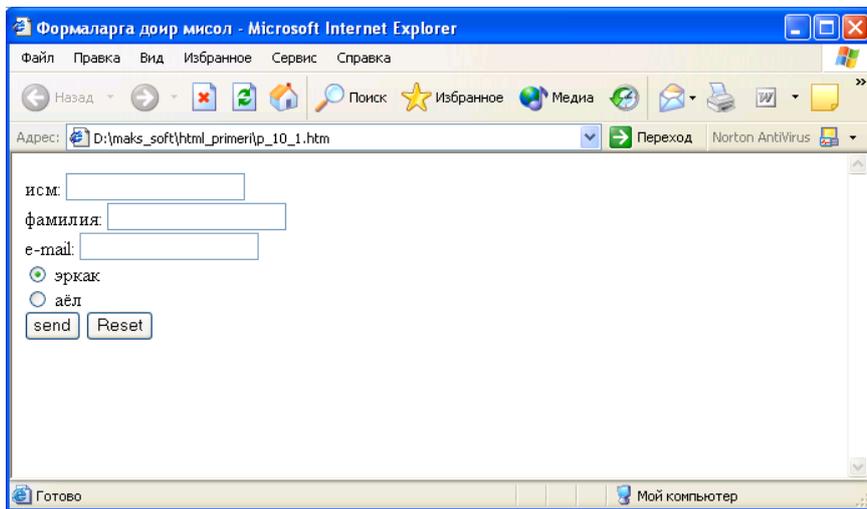
*onreset* = сценария (формани қайтадан ўрнатиш)

## Лаборатори ишини ишлаш бўйича

### 1-кўрсатма

**Input, form radio, бошқариш** тугмалари билан ишлашга доир

```
<html>
<head>
  <title>Формаларга доир мисол</title>
</head>
<body>
<form action="http://maks//cgi/person.exe" method="post">
<p>
<label for="firstname"> исм: </label>
<input type="text" id="firstname">
<br>
<label for="lastname"> фамилия: </label>
<input type="text" id="lastname">
<br>
<label for="email"> e-mail: </label>
<input type="text" id="mail">
<br>
<input type="radio" name="sex" value="male" checked="checked "> эркак <br>
<input type="radio" name="sex" value="femole"> аёл <br>
<input type="submit" value="send">
<input type="reset" value="Reset">
</p>
</form>
</body>
</html>
```



## Маълумотларни киритиш *INPUT* элементи

Умумий кўриниши:

**Синтаксис:** <INPUT> (Ёпиловчи тэги йўқ)

**Атрибуты:** id, class, style, title, lang, dir, ходиса

*type* = TEXT | PASSWORD | CHECKBOX | RADIO | SUBMIT | RESET |  
FILE | HIDDEN | IMAGE | BUTTON (Киритиш тури)

*name* = CDATA (Элемент номи)

*value* = CDATA (Элементнинг бошланғич қиймати)

*checked* = CHECKED (Байроқчалар ва улагичларнинг танланганлик ҳолати)

*readonly* = READONLY (Матн ва парол учун *фақат уқиш учун*)

*disabled* = DISABLED (Элемент тақиқланган)

*size* = CDATA (Элементнинг бошланғич кенглиги)

*maxlength* = сон (Матндаги максимал символлар сони)

*src* = URI (URI (Universal Resource Identifier) Тугманинг график шакли)

*alt* = текст (Тугма учун альтернатив матн)

*usemap* = URI (URI мурожаатларнинг мижоз картаси)

*ismap* = ISMAP (мурожаатларнинг сервер картаси)

*accesskey* = символ (Тез мурожаат учун тугма номи)

*tabindex* = сон (Tab тугмаси билан бориш учун тартиб рақами)

*accept* = файллар типлари (Мумкин бўлган файллар типлари)

*onfocus* = сценария (Элемент фокусланганда юз беради)

*onblur* = сценария (Элемент фокуси йўқолганда содир бўлади)

*onselect* = сценария (Элемент матни танланганда)

*onchange* = сценарий (Элемент қиймати ўзгарганда)

*text* – бирқаторли киритиш майдони. *value* – бу параметрга майдонинг бошланғич қиймати берилади. *size* – майдонинг ўлчами. *maxlength* – бир майдондаги мумкин бўлган символлар сони. *readonly* – маттни ўзгартиришни тақиқлайди.

*password* – паролни киритиш майдони, бу майдонинг матнни киритиш майдонидан фарқи шундаки бу ерда киритилаётган символлар \* билан алмаштирилади ва қайси белгилар киритилаётганлиги экранда кўринмайди. Аммо киритилган матн серверга ҳеч қандай кодланмасдан оддий матн кўринишида юборилади, шунинг учун бу элемент ёрдамида махфий маълумотларни (кредит карточка номерлари) юбориш тавсия қилинмайди.

*checkbox* – байроқча *Value* га танланган элемент қиймати юборилади.

*radio* – улагич. *Value* нинг қиймати танланган элемент қийматига тенг бўлади.

*reset* – форманинг бекор тугмаси. Бу тугма босилса формадаги ҳамма қийматлар бошланғич қийматга қайтарилади.

*submit* – форманинг жўнатиш тугмаси. Бу тугма босилса формадаги қийматлар серверга “*value=қиймат*” кўринишида жўнатилади. Агарда битта формада бар нечта шундай (*submit*) тугма бўлса унда уларга *name* атрибути ёрдамида ном бериш керак. Чунки жўнатилган маълумотларни қайта ишловчи дастур қайси тугма босилганини аниқлаб олиши лозим.

*image* – формани жўнатишнинг график тугмаси. Бу ерда *src* атрибути ёрдамида график тасвирнинг манзили берилади, *alt* ёрдамида эса тугманинг альтернатив матни ўрнатилади. График тугманинг сичқонча босилган жойининг координаталари серверга *name.x* ва *name.y* ўзгарувчилар ёрдамида жўнатилади, бу ерда *name* тугма номи.

*button* – умумий турдаги тугма. *value* атрибутига тугма матни берилади. *onclick* атрибутига тугма босилганда ишлаши керак бўлган сценарий берилади.

#### 10.4. BUTTON тугмаси

**Синтаксис:** <BUTTON>...</BUTTON> (Матнли элемент)

**Атрибутилар:** *id*, *class*, *style*, *title*, *lang*, *dir*, *ҳодиса*

*type* = SUBMIT | RESET | BUTTON (Киритиш тури)

*name* = CDATA (Элемент номи)

*value* = CDATA (Тугманинг бошланғич қиймати)

*disabled* = DISABLED (Элемент тақиқланган)

*accesskey* = символ (Тез муружаат учун тугма номи)

*tabindex* = *сон* (Tab тугмаси билан бориш учун тартиб рақами)

*onfocus* = сценария (Элемент фокусланганда юз беради)

*onblur* = сценария (Элемент фокуси йўқолганда содир бўлади)

Бундай элемент ёрдамида қосил қилинган тугмалар *input* элементи ёрдамида ҳосил қилинган тугмага ўхшайди лекин унинг ўз ташкил этувчилари бор. Бу ерда *name* атрибути ёрдамида тугма номи берилади. *value* билан унинг бошланғич қиймати, *type* ёрдамида эса тугма тури аниқланади.

*submit* – формани жўнатиш тугмаси (жимлик бўйича қабул қилинган).

*reset* – формани бекор қилиш.

*button* – умумий кўринишдаги тугма.

*disabled* – бу атрибут тугмани актив бўлишини тақиқлайди. Бу дегани шу элемент фокусга олинмайди.

*accesskey* – ёрдамида тугмага тезда муружаат қилиш учун клавиатурадаги бирор тугманинг *unicode* даги символи берилади.

Мисол: 10\_2.

```
<html>
```

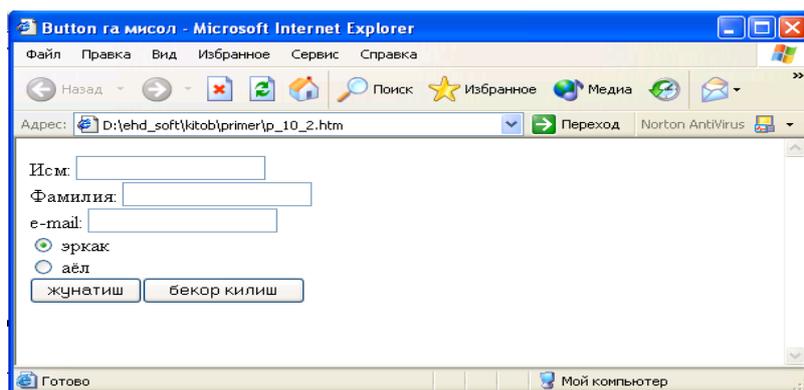
```
<head>
```

```
<title> Button га мисол</title>
```

```

</head>
<body>
<form action="http://maks//cgi/person.exe" method="post">
<p>
<label for="firstname"> исм: </label>
<input type="text" id="firstname">
<br>
<label for="lastname"> фамилия: </label>
<input type="text" id="lastname">
<br>
<label for="email"> e-mail: </label>
<input type="text" id="mail">
<br>
<input type="radio" name="sex" value="male" checked="checked " > эркак <br>
<input type="radio" name="sex" value="femole"> аёл <br>
<button name="submit" value="submit" type="submit">
Жўнатиш </ button>
<button name="reset" type="reset"> бекор қилиш </ button>
</p>
</form>
</body>
</html>

```



### Лаборатори ишини ишлаш бўйича

2-кўрсатма Менюлар ва Кўп сатрли матнли майдон. TEXTAREA элементи билан ишлаш тугмалари билан ишлашга доир

## Менюлар. SELECT, OPTGROUP ва OPTION

**Синтаксис:** <SELECT>...</SELECT> (Матнли элемент)

**Атрибутлар:** *id*, *class*, *style*, *title*, *lang*, *dir*, *ҳодиса*

*name* = CDATA (Элемент номи)

*size* = (Кўринадиган опциялар сони)

*multiple* = MULTIPLE (Куп опцияларни танлаш)

*disabled* = DISABLED (Элемент тақиқланган)

*tabindex* = *сон* (Tab тугмаси билан бориш учун тартиб рақами)

*onfocus* = сценария (Элемент фокусланганда юз беради)

*onblur* = сценария (Элемент фокуси йўқолганда содир бўлади)

*onchange* = сценария (Элемент қиймати ўзгарганда)

*select* элементи ёрдамида рўйхатдан бирор опцияни танлаш имкониятини яратади.

Одатда бу элемент ёрдамида очилувчи менюлар тузилади ва менюнинг опциялари (пунктлари) *optgroup* ва *option* лар ёрдамида аниқланади. *name* атрибути ёрдамида элементнинг номи берилади, *size* атрибути эса экранда ҳосил қилинадиган опциялар сонини аниқлайди, агар опциялар *size* да кўрсатилган сондан катта бўлса, у ҳолда браузер ўтказиш йўлакчаси (прокрутка)ни ҳосил қилади.

Жимлик бўйича фойдаланувчи рўйхатда кўрсатилган опциялардан фақат биттасини танлаши мумкин, агар бир вақтнинг ўзида бир нечта опцияни танлаш керак бўлса, у ҳолда *multiple* атрибути билан буни амалга ошириш мумкин. Бу ҳолда ҳар бир танланган опция *name=value* кўринишида алоҳида жўнатилади. *disabled* атрибути ёрдамида биз элементни активлаштирмаймиз, аммо ҳосил қилиб қўйишимиз мумкин. *tabindex* нинг қиймати 0 дан 32767 гача бўлган бутун сонлар бўлиши мумкин. Бу қиймат билан биз Web саҳифа бўйлаб Tab тугмаси ёрдамида ҳаракатланганда фокуснинг ҳосил бўлиш номерини аниқлаймиз. Агар бирор элементнинг *Tabindex* қиймати 0 га тенг ёки берилмаган бўлса, у ҳолда бу элемент бошқа элементлар *Tabindex* нинг қиймати аниқлангандан (мусбат сонлар билан) кейин фокусланади. Агар бир нечта элементларнинг *Tabindex* ларининг қийматлари бир хил бўлса у ҳолда биринчи учрагани фокусланади.

*onfocus* - элемент фокусланганда ҳосил бўладиган ҳодиса

*onblur* - элемент фокусини йўқотганда юз берадиган ҳодиса

*onchange* - элементнинг қиймати ўзгарганда юз берадиган ҳодиса

**Синтаксис:** <OPTGROUP>...</OPTGROUP> (Матнли элемент)

**Атрибутлар:** *id*, *class*, *style*, *title*, *lang*, *dir*, *ҳодиса*

*label* = матн (опциялар группасининг белгиси)

*disabled* = DISABLED (Элемент тақиқланган)

**Синтаксис:** <OPTION>...</OPTION> (Матнли элемент)

**Атрибутлар:** *id*, *class*, *style*, *title*, *lang*, *dir*, *ҳодиса*

*value* = CDATA (опция номи)

*label* = матн (опция белгиси)

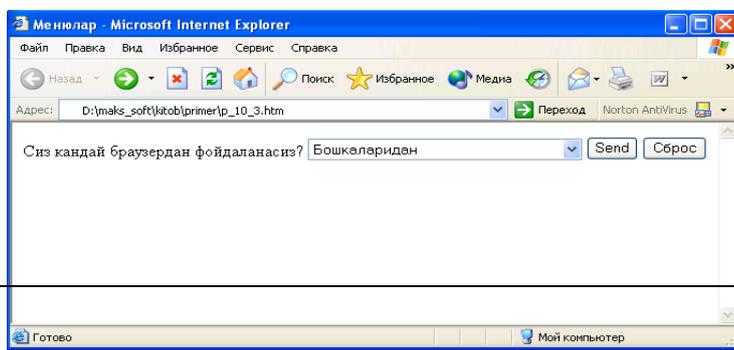
*selected* = SELECTED (опция танланган)  
*disabled* = DISABLED (элемент тақиқланган)

Мисол: 10\_3.

```
<html>
<head>
<title>Менюлар </title>
</head>
<body>
<FORM action="http://maks/cgi-bin/Qunduz/php" method="post">
  <P>Сиз қандай браузердан фойдаланасиз?
  <SELECT name="browser">
    <OPTGROUP label="Netscape Navigator">
      <OPTION label="4.x ёки юқори">Netscape Navigator4.x ёки юқори</OPTION>
      <OPTION label="3.x">Netscape Navigator 3.x</OPTION>
      <OPTION label="2.x">Netscape Navigator 2.x</OPTION>
      <OPTION label="1.x">Netscape Navigator 1.x</OPTION>
    </OPTGROUP>
    <OPTGROUP label="Microsoft Internet Explorer">
      <OPTION label="4.x ёки юқори">Internet Explorer4.x ёки юқори</OPTION>
      <OPTION label="3.x">Internet Explorer 3.x</OPTION>
      <OPTION label="2.x">Internet Explorer 2.x</OPTION>
      <OPTION label="1.x">Internet Explorer 1.x</OPTION>
    </OPTGROUP>
    <OPTGROUP label="Opera">
      <OPTION label="3.x ёки юқори">Opera 3.x ёки юқори</OPTION>
      <OPTION label="2.x">Opera 2.x</OPTION>
    </OPTGROUP>
    <OPTION selected="selected">Бошқаларидан</OPTION>
  </SELECT>
  <INPUT type="submit" value="Send"> <INPUT type="reset">
</P>
</FORM>
```

</body>

</html>



## Кўп сатрли матнли майдон. TEXTAREA элементи.

**Синтаксис:** <textarea> ...</textarea> (Матнли элемент)

**Атрибутлар:** *id*, *class*, *style*, *title*, *lang*, *dir*, *ҳодуца*

*name* = CDATA (Элемент номи)

*rows* = *сон* (Кўринадиган сатрлар сони)

*cols* = *сон* (Кўринадиган устунлар сони)

*readonly* = READONLY (Фақат ўқиш учун)

*disabled* = DISABLED (Элемент тақиқланган)

*accesskey* = *символ* (Тез мурожаат учун тугма номи)

*tabindex* = *сон* (Tab тугмаси билан бориш учун тартиб рақами)

*onfocus* = *сценария* (Элемент фокусланганда юз беради)

*onblur* = *сценария* (Элемент фокуси йўқолганда содир бўлади)

*onselect* = *сценария* (Матн элементи танланганда)

*onchange* = *сценария* (Элемент қиймати ўзгарганда)

*textarea* ёрдамида биз формаларга кўп сатрли матнларни кирита оламиз. Форма серверга жўнатилгандан *name=value* кўринишида қийматлар тўпламидан иборат бўлади. *value* бу фойдаланувчи томонидан киритилган қиймат. *rows* ва *cols* атрибутлар мос равишда матн киритиладиган ойнанинг сатрлар ва устунлар сонини билдиради. Агар киритилган матн ойнага сиғмаса браузер автоматик равишда (прокрутка) йўлакча ҳосил қилади. Амалиётда киритиладиган матннинг ўлчами 32 кв ёки 64 кв дан ошмаслиги керак. Шарт бўлмаган атрибут *readonly* майдондаги матнни ўзгартиришни тақиқлайди. *Disabled* атрибути эса элементни фаол (актив) ҳолатини ўчиради. Элементга тезда мурожаат қилиш учун *accesskey* атрибутига бирор тугманинг *unicode* даги символини бериш керак. *tabindex* ёрдамида эса Web саҳифа бўйлаб Tab тугмаси ёрдамида ҳаракатланганда нечанчи ўринда фокусланиш номери берилади. Бунинг қиймати 0 дан 32767 гача бўлган бутун сонлар бўлиши мумкин.

### Бошқариладиган элементлар гуруҳи.

*fieldset* элементи формадаги элементлар гуруҳини аниқлайди. Бу элемент ёрдамида формани бир нечта бўлақларга ажратиш мумкин. Бу эса фойдаланувчига формани тўлдиришда қулайлик яратади. Элемент очилувчи <*fieldset*> ва ёпилувчи </*fieldset*> тегларига эга.

*fieldset* элементининг ичида жойлашадиган *legend* теги мавжуд бу тег билан элементлар гуруҳининг сарлавҳаси берилади.

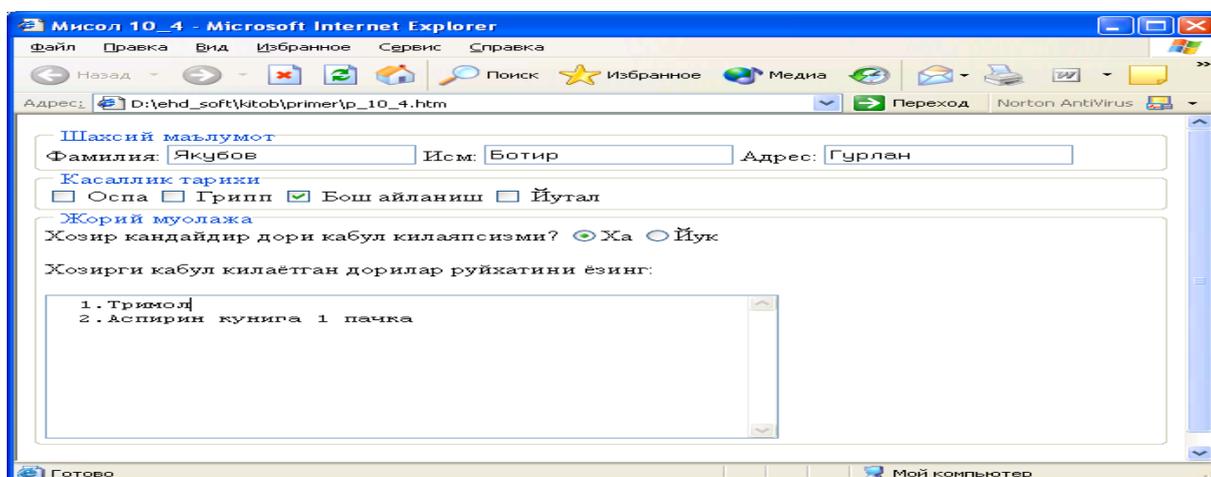
Мисол: 10\_4.

```
<html><head><title>Мисол 10_4</title></head>
<body><FORM action=" http://maks/cgi-bin/Qunduz/php " method="post">
<P>
<FIELDSET> <LEGEND>Шахсий маълумот</LEGEND>
Фамилия: <INPUT name="personal_lastname" type="text" tabindex="1">
Исм: <INPUT name="personal_firstname" type="text" tabindex="2">
Манзил: <INPUT name="personal_address" type="text" tabindex="3">
```

```

</FIELDSET> <FIELDSET> <LEGEND>Касаллик тарихи</LEGEND>
<INPUT name="history_illness"
  type="checkbox" value="Smallpox" tabindex="20"> Оспа
<INPUT name="history_illness" type="checkbox" value="Mumps" tabindex="21">
Грипп
<INPUT name="history_illness" type="checkbox" value="Dizziness" tabindex="22">
Бош айланиш <INPUT name="history_illness" type="checkbox"
  value="Sneezing" tabindex="23"> Йўтал </FIELDSET>
<FIELDSET>
<LEGEND>Жорий муолажа</LEGEND>
Ҳозир қандайдир дори қабул қилаяпсизми?
<INPUT name="medication_now" type="radio" value="Yes" tabindex="35">Ҳа
<INPUT name="medication_now" type="radio" value="No" tabindex="35">Йўқ
<p> Ҳозирги қабул қилаётган дорилар рўйхатини ёзинг:</p>
<TEXTAREA name="current_medication"
  rows="10" cols="50" tabindex="40"> </TEXTAREA> </FIELDSET> </P>
</FORM>
</body></html>

```



## Топшириқлар

### Қуйидаги вариантларнинг формасини яратиш

1. telefonga sms yuborish forumnini yaratish
2. e-mailga xat yuborish forumnini yaratish

3. Paynet to`lovi forumnini yaratish
4. passport ma`lumotlarini kiritish uchun forumnini yaratish
5. abituriyent ma`lumotlarini kiritish forumnini yaratish
6. Telefon markalari haqidagi ma`lumotlarni kiritish forumnini yaratish
7. Avtomobillar markalari haqidagi ma`lumotlarni kiritish forumnini yaratish
8. Kompyuter sotish jarayonining formasini yaratish
9. Abutturent qabul qilish formasini yaratish
10. Ishga qabul qilish so`rovnomasining formasini yaratish
11. Ishga taqsimlash formasini yaratish
12. Avtomobillar ma`lumotlarini kiritish formasini yaratish
13. Aviaqassa formasni yaratish
14. Kasalxonalarda kasallik varaqasini yuritish formasni yaratish
15. e-mail yaratish formasini yaratish

### НАЗОРАТ САВОЛЛАРИ

- 1) Форма нима ?
- 2) Форманинг бошқариш элементлари ?
- 3) Форманинг асосий тегларини айтиб беринг ?
- 4) Маълумотларини киритиш элементи қайси ?
- 5) BUTTON тугмаси вазифаси ?
- 6) TEXTAREA тегининг вазифаси ?
- 7) INPUT тегининг вазифаси ?
- 8) action нинг вазифаси ?
- 9) method нима ?
- 10) Элементлар ходисаларини айтиб Беринг ?

## Лаборатория иши №-3

Мавзу : HTMLда гипер мурожаатлар ва фреймлар.

Ишдан мақсад: HTMLда гипер мурожаатлар ва фреймлар билан ишлаш кўникмасини хосил қилиш.

### НАЗАРИЙ ҚИСМ

Web саҳифаларга гипермурожаат WWW нинг асосий хусусиятларидан биридир. Исталган бир ҳужжатдан бошқа бир WWW ҳужжатга HTML нинг махсус тэги ёрдамида мурожаат бўлиши мумкин. Web га саёҳат пайтида сиз Web саҳифаларда гиперматнли мурожаатларга тўқнашган бўлсангиз керак. Бу матн фрагментлари кўк шрифтда ва таги чизилган бўлади. Шу тариқа тасвирланган матнлар мурожаат матнлари дейилади. Агар сиз уни сичқонча билан туртсангиз у автомат равишда бошқа Web – саҳифага мурожаат қилади.

Ҳар қандай URL манзилида 3 қисм мавжуд: протокол, интернет узели, файл номи (ўнга йўл кўрсатилади, шарт бўлса).

*Протокол* – бу 2 та компьютернинг бир-бири билан алоқасининг қоидаси, биз WWW нинг стандарт протоколи http (HyperText Transfer protocol) ҳақида гаплашамиз. Яна сиз интернет узели ва файл номини ҳам кўрсатишингиз керак.

Масалан:

*<http://www.ursu.uzpak.uz/create>*

Буни тушунтирадиган бўлсак, браузер кўрсатилган *ursu.uzpak.uz* манзилини (http) аниқ бир уланиш усули ёрдамида интернетга улайди ва *create* директориясидаги Web-саҳифани топади. Ҳар бир интернетдаги ҳужжат ва файл алоҳида ресурсларнинг универсал кўрсаткичи деб номланувчи манзилга эга (uniform resource locator - URL). URL электрон почта манзилини эслатиб туради. Интернет – компьютерлар URL ни қандай жўнатишни, уни қандай топишни ва нима билан улашни ўзлари тушунади. Web-саҳифа қаерда жойлашишидан катъий назар хох у Янги Зелландияда бўлсин ёки Японияда бўлсин WWW улар билан бир хил ишлайди. Браузер ҳар бир URL ни ишга солиб сизга интернетдаги ҳужжатни топиб беради ва автоматик равишда тақдим қилади.

Дунё бўйлаб Web узелларни кўриб чиқмоқчи бўлган шахсга URL ҳақида ва

керакли домен калитни қандай топишни ўйлашга заррача эҳтиёж йўқ.

### Мурожаатлар анатомияси ва маҳаллий Web-саҳифаларга мурожаат.

Web саҳифага мурожаатларни жойлаштириш икки турда бўлади (бир-бирига ўхшаш). Дастлаб маҳаллий ҳужжатларга мурожаат қилишни кўриб чиқамиз. Маҳаллий ҳужжат бу шундай ҳужжатки сизнинг уй саҳифангиз интернетнинг қайси жойида жойлашган бўлса у ҳам шу ерда жойлашган бўлади. Маҳаллий ҳужжатларга мурожаат қилиш осон ва бунда сиз тўлиқ URL ни билишингиз шарт эмас, фақат файлга йўл ва унинг номини берсангиз кифоя. Сўнгра сиз WWW нинг ҳохлаган жойидаги ҳужжатларга мурожаат қилишни ўрганасиз. Бу турдаги мурожаатда сиздан тўлиқ URL талаб қилинади.

Фараз қилайлик сизда битта ҳужжат учун маълумот кўп ва сиз уни бир неча HTML файлларга бўлишга қарор килдингиз. У ҳолда асосий Web-саҳифада бу ҳужжатнинг ҳар бирига тўлиқ ва осон ўтувчи мурожаатларни жойлаштириш керак бўлади.

Ҳатто сиз асосий Web-саҳифангизни мундарижа кўринишида қилишингиз мумкин ва бунда алоҳида саҳифаларга мурожаат қилинади.

Дастлаб `<a href>` теги киритилади. `<a>` тег ўзаг тег деб номланади. У Web-саҳифаларга мурожаат учун ишлатилади. `href` калит сўзи браузерга сиз мурожаат қилаётганингиз ҳақида хабар беради. `</a>` ёпилувчи тегини эсдан чиқармай ёзиб кетиш керак. `misol.htm` файлига мурожаатларни яратиш учун куйидаги кодни киритишингиз керак:

```
<a href="misol.htm"> Мисолни кўриш </a>
```

Шундай қилиб «Мисолни кўриш» ибораси экранда мурожаат матни кўринишида пайдо бўлади.

Бу матнга сичқонча билан туртсангиз, `misol.htm` файлини очилади.

Файлга йўлни кўрсатишни эсдан чиқармаганингизни текшириб кўринг. Масалан, `tiger.htm` файли `Exhibits` директориясида сақланган бўлса, `href` калит сўзида `exhibits/tigers`. `<a href="exhibits/tigers.htm">` О тиграк

`</a>` **WWW нинг хохлаган жойига мурожаат.**

Бошқа бир Web узелга мурожаат қилиш учун ҳужжатнинг тўлиқ URLи талаб қилинади. Лекин мурожаатнинг бу тури ҳам шундай тарзда, яъни `<a href>`

ва `</a>` каби бўлади. URL ҳамиша `http://` билан бошланади. Шу тарзда браузер ҳужжатни сизнинг Web узелингиздан эмас WWW дан қидириб бошлайди. URL нинг қолган қисмлари сиз излаётган ҳужжатнинг номи ва интернетдаги йўли, узелидан иборат бўлади:

```
<a href="http://www.yahoo.com/arts/performinarts/circuses/"> yahoo олами  
руйхати </a>
```

### Мурожаат ранги.

World Wide Web даги мурожаат матнининг рангини ўзгартиришингиз мумкин. Сизнинг Web саҳифангизнинг матнини рангини `body` теги ёрдамида ўрнатиш мумкин: `<body next=red>`

Шу тарзда гипермурожаат матнининг рангини ҳам бошқариш мумкин. Қуйида 3 та янги калит сўз келтирилган:

`link` - `link` калит сўзи `Text` каби ишлатилади. `link=blue` ўрнатилган бўлса (ёки бошқа бир ранг), фойдаланувчи ҳали ишлатмаган (сичқонча билан туртмаган) мурожаатлар кўк рангда бўлади.

```
<body text=red link=blue>
```

`vlink` - Айрим ҳолларда бир неча мурожаатларнинг ранги бошқаларидан ажралиб турганини сезган бўлишингиз мумкин. Бу мурожаат сиз томонингиздан фаоллаштирилганлигидан далолат беради. Фойдаланувчи мурожаат қилган мурожаатларнинг рангини (Visited line) `vlink` калит сўзи билан ўзгартириш мумкин. Мурожаатнинг рангини очик кўк рангга ўзгартириш учун қуйидаги кодда ёзиш керак:

```
<body text= line link=aqua vlink=silver>
```

`alink` - Мурожаатга сичқонча билан туртган пайтдаги рангини ўзгартириш мумкин. Мурожаатнинг фаол ранги `alink` калит сўзи ёрдамида берилади. Кимдир сизнинг саҳифангизга кирганида мурожаат ранги `link` билан берилган рангда бўлади. Лекин қачонки сиз уни сичқонча билан туртсангиз қисқа вақтгача `alink` билан берилган рангга ўтади.

Қуйида фаол мурожаатнинг рангини очик сариқ рангга ўзгартирамиз:

```
<body text= line link=aqua vlink=silver alink=yellow>
```

### Тасвирларни мурожаат сифатида ишлатиш.

WWW-саҳифаларда мурожаат воситаси бўлиб фақат матн эмас, балки тасвирлар ҳам ишлатилади. Бу шуни билдирадики, қачонки сиз тасвирларни сичқонча билан туртсангиз экранда бошқа Web-ҳужжат пайдо бўлади.

Бошқа HTML - ҳужжатга тасвир ёрдамида мурожаат қилиш учун худди матн сингари бўлади. Фақат бунда матн ўрнига тасвирни беришингиз керак. Жимлик ҳолатида Web-браузерлар тасвир мурожаатни кўк рамка билан ўраб олади. Тасвирнинг ҳохлаган жойига сичқонча билан туртиш орқали мос ҳужжатга ўтилади. Мурожаат сифатида тасвирларнинг ҳохлаган тури, расмлар ва пиктограммалардан фойдаланиш мумкин. (фон безакларидан ташқари).

Маслаҳат тасвирлар мурожаат сифатида тез-тез ишлатилади. Иложи борича тушунарли тасвирлардан фойдаланинг, шунда фойдаланувчилар мурожаат уларни қандай манзилга олиб боришни тушунадилар.

Масалан, агар сиз уй саҳифангизда севимли кўшиқларингиз рўйхатига мурожаат қилмоқчи бўлсангиз, мусиқага доир пиктограммадан фойдаланинг. Тасвир мурожаатларидан фойдаланганда альтернатив матн бериш зарур. Тасвир кўринмай қолганда фойдаланувчилар шу орқали мурожаатдан фойдаланишлари мумкин. Айрим фойдаланувчиларда тасвирни юклаш кўп вақтни талаб қилади. Улар учун альтернатив матн жуда зарур, ана шунда улар нима кўришларини билиб оладилар. Ихтиёрий бир тасвирга мурожаат чиқариш учун қуйидагича кодни киритамиз:

```
<a href="http://www.mep.com/quel">
```

```
 </a>
```

Бу матн ўрнига `<img>` тегини киритганимизга эътиборингизни қаратинг. Шу тариқа график мурожаат бериш мумкин.

### **Фреймлар ҳақида умумий маълумот.**

Бизга маълумки бир вақтнинг ўзида браузер ойнасига иккита HTML ҳужжатни юклай олмаймиз. Агар биз ишлатаётган Web саҳифаларимизнинг барчасида бир хил меню бандлари мавжуд бўлса ҳар сафар Web саҳифани юклаганимизда бир хил маълумотни қайта-қайта юклашга тўғри келади. Бу маълумотлар унча катта бўлмаслиги мумкин, лекин уни юклаш маълум бир секундларни олади. Шунинг учун Web саҳифаларнинг ўзгармайдиган

элементларини ҳар сафар юкламасдан ўзгаришсиз қолдириш керак. Бу муаммоларни ечиш имкониятлари бор. Биз битта ойнани бир нечта тўғри тўртбурчакларга бўлиб уларнинг ҳар бирига битта HTML ҳужжат юклашимиз мумкин. Бу тўғри тўртбурчак соҳаларни биз фреймлар деб атаймиз. HTML да фреймларнинг икки хил кўриниши мавжуд бўлиб булар оддий ва сурилувчи фреймлардир. Фреймли структурага эга бўлган ҳужжатлар `<frameset>` ва `</frameset>` тэглари ёрдамида яратилади. Бу икки тэг орасида ҳосил қилинаётган алоҳида фреймларга оид маълумотлар эълон қилинади. Алоҳида фреймлар `<frame>` ва `</frame>` тэглари ёрдамида яратилади. Агар сизнинг браузерингиз фреймларни тушунмаса у ҳолда экранда `<noframes>` ва `</noframes>` тэглари орасида ёзилган маълумот тасвирланади.

`<frameset>` тегининг асосий вазифаси браузер ойнасини бир нечта бўлақларга бўлиш. Бунинг учун асосан вертикал ва горизантал бўйича фреймлар ўлчами ва сонини кўрсатиб турувчи *rows* ва *cols* параметрлари ишлатилади.

- *cols* параметри браузер ойнасида горизантал бўйича фреймлар сони ва ўлчамини кўрсатади. Параметр қиймати сифатида вергуллар билан ажратилган фреймлар ўлчамлари рўйхати келтирилади. ўлчамлар одатдагидек сонларда ёки процентларда берилади. Жимлик бўйича 100% қиймат ишлатилади.

- *rows* параметри браузер ойнасида вертикал бўйича фреймлар сони ва ўлчамини кўрсатади. Параметрга қиймат бериш *cols* параметри билан бир хил бўлади.

## Лаборатория ишни бажаришга доир 1-кўрсатма

Мисол: 9\_1

```
<html>
```

```
<head><title>Фреймлар</title></head>
```

```
<frameset cols="150,*">
```

```
<frame src="9_1_1.htm">
```

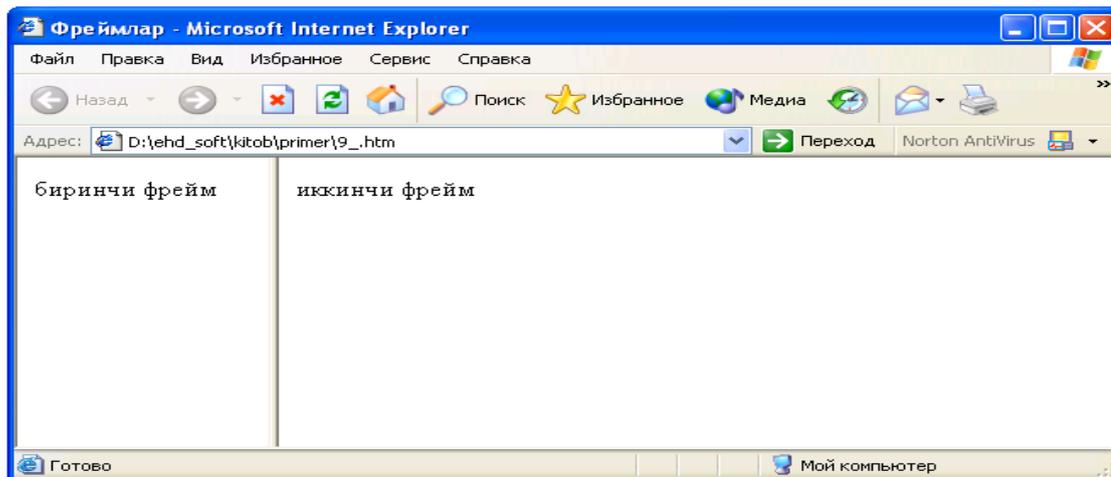
```
<frame src="9_1_2.htm">
```

```
<noframes><p>афсуски сизнинг браузерингиз фреймларни тушинмайди.
```

Браузернинг янги версияларидан фойдаланинг</p>

</noframes>

</frameset>



</html>

Диққат билан HTML кодни ва унинг экранда чиқарган натижасини таҳлил қилиш натижасидан кўринадики, бу ерда `<body>` ва `</body>` тэглари ишлатилмаган. Бу тэглр орасида экранда ҳосил қилниши керак бўлган барча асосий маълумотлар ёзилади, `<frameset>` тэги бўлса браузерга ойнани фреймларга ажратиш ҳақида маълумот беради. Юқоридаги мисолда биз `cols` параметрини иккита қиймати билан ишлатдик. Бу қийматда биринчи фрейм учун 150 пиксел жой иккинчи фрейм учун эса қолган жойнинг ҳаммаси ишлатилиши кўрсатилган. Мисолда кўрганимиздек `<frame>` тэглари ҳар бир фреймни алоҳида эълон қилади ва уларнинг хоссаларини кўрсатади. Мисолимизда биз қиймат сифатида html хужжатнинг манзилини қабул килувчи `src` параметрини ишлатдик. Бундан ташқари биз `<noframes>` ва `</noframes>` тэгларини ҳам ишлатдик. Бу тэглр орасида ёзилган маълумот браузер томонидан фрейм технологияси тушунилмаган ҳолатда экранда пайдо бўлади. Берилган ўлчамда бўлинган фреймлар чегарасини фойдалануви томонидан ўзгартириш имконияти мавжуд. Бунинг учун *splitter* деб номланувчи чегара чизиғидан сичқонча кўрсаткичи билан ушлаб суриш кифоя. Асосий браузер ойнасини битта устунда жойлашган иккита фрейм ва қолган жойни эгалловчи учинчи фрейм кўринишида бўлишимиз учун куйдаги html кодни ёзишимиз керак.

## Лаборатория ишни бажаришга доир 2-кўрсатма

### Frame тэги ва унинг параметрлари.

`<frame>` тэги алоҳида фреймлар хоссаларини ўрнатиш учун ишлатилади. Энди биз `<frame>` тэгида ишлатиладиган параметрлар билан танишиб чиқамиз.

- *name* параметри фреймнинг номини кўрсатишда ишлатилади, буни *id* параметри билан алмаштирмаслик керак. *name* параметри ёрдамида берилган ном ўзак фреймдан ташқарида жойлашган бошқа бир ҳужжатни юклаганда тэглarda гипермуружаат сифатида ишлатилади.

- *src* параметри фреймда тасвирланиши керак бўлган HTML ҳужжатнинг URL ини беришда ишлатилади.

- *frameborder* параметри фрейм чегара хошияларининг бўлиш ёки бўлмаслигини кўрсатади. Параметр қиймати 0 ёки 1 сонлари бўлиши мумкин. Жимлик бўйича параметр 1 қиймат қабул қилади ва фрейм чегара хошияси кўринувчи бўлади. Агар қиймат 0 фрейм чегараси хошияси кўринмайди.

- *marginwidth* параметри фреймнинг горизонтал бўйича ўлчамини пикселларда беради.

- *marginheight* параметри фреймнинг вертикал бўйича ўлчамини пикселларда беради.

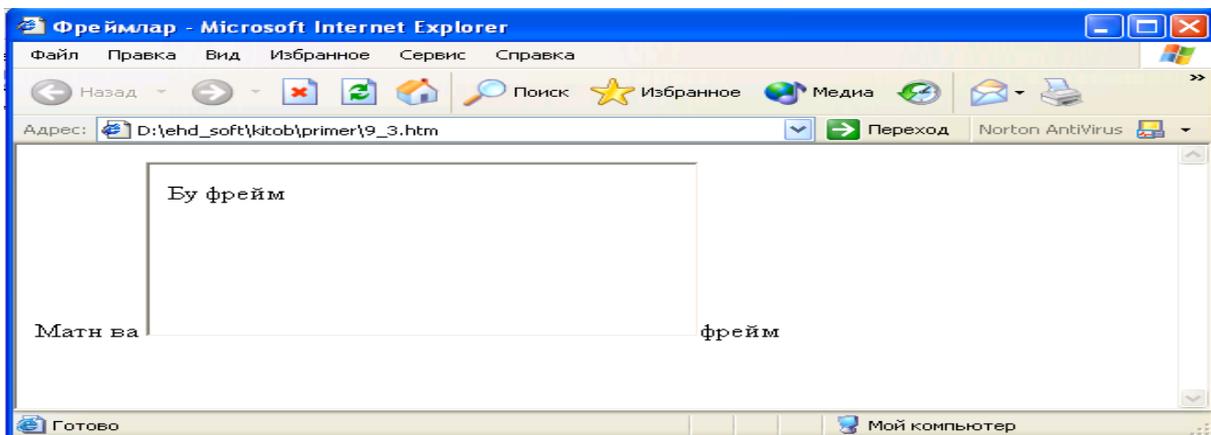
- *noresize* параметри фрейм ўлчамларининг ўзгармаслигини таъминлайди. Агар биз бу параметрни қўлласак фойдаланувчи томонидан фрейм чегараларини ўзгартириш имкони бўлмайди. Бу параметрда қиймат йўқ.

- *scrolling* параметри фойдаланувчига фреймдаги ўтказиш йўлакчаси (линейка прокрутки)ни бошқариш имконини беради. Қиймат сифатида қуйидаги *auto*, *yes*, *no* калит сўзларидан биттасини қабул қилади. Жимлик бўйича *auto* қиймати қабул қилинган бўлади, бу эса фреймдаги маълумотлар тасвирланиш соҳасига жойлашмаган вақтда ўтказиш йўлакчаси пайдо бўлишини билдиради. *yes* қиймати эса фреймдаги маълумот тасвирланиш соҳасига сиғиши ёки сиғмаслигидан қатъий назар ўтказиш йўлакчаси доимо чиқиб туришини билдиради. *no* қиймати эса фреймда ўтказиш йўлакчасининг чиқишини умуман тақиқлайди.

Биз юқоридаги мисолларимизда фреймлар браузер ойнасини алоҳида бўлақларга тўлиқ бўлганини кўрдик. Лекин фреймларнинг шундай тури ҳам мавжудки, уларни HTML ҳужжатга худди график тасвир жойлаштиргандай жойлаштириш мумкин. Бундай турдаги фреймларни HTML ҳужжатга жойлаш учун `<iframe>` тэги ишлатилади. График объект жойлаштириш тэгидан фарқли равишда бу тэг ўзининг ёпилувчи `</iframe>` тэгига эга.

Мисол 9\_3:

```
<html>
<head><title>Фреймлар</title>
</head>
<body>
<p>Матн ва <iframe src="9_3.htm"></iframe>фрейм</p>
</body>
</html>
```



## 1-Лаборатория ишининг топшириқлари ҳар бир талаба битта фаннинг электрон қўлланмасини **Fremlar** ёрдамида яратиш

### Variantlar

1	Интернет технологиялари
2	Йуналишга кириш
3	Информатика
4	Компьютер графикаси ва дизайни
5	Дастурлаш технологиялари
6	Маълумотлар базалари

7	Математик моделлаштириш
8	Internet мухитида дастурлаш
9	Компьютер тизимлари ва тармоклари
10	Объектга йуналтирилган дастурлаш тиллари
11	Интеллектуал тизимлар
12	Инсон машина интерфейси
13	Автоматлаштирилган системаларни куриш асослари
14	Компьютер тармоклари ва тармок ресурсларнинг архитектураси
15	Информацион хавфсизлик
16	Информацион тизимлар
17	Маълумотлар структураси ва алгоритмлар
18	Математик дастурлаш ва оптималлаш усуллари
19	Тизимли дастурий таъминот
20	Хисоблаш тизимларини ахборот асослари
21	Электрон дизайн
22	Операцион тизимлар ва офис иловалари
23	Телекоммуникация тизимлари ва тармоклари
24	TCP/IP тармоқлар, IP технологиялар
25	Корпоратив тармоқлар, маъмуриятлаштириш масаласи, узаро фоалият
26	Тармоқларнинг ташкили ва хавфсизлиги
27	WEB dasturlash
28	Java дастурлаш тили
29	PHP tilida dasturlash
30	

### НАЗОРАТ САВОЛЛАРИ

- 1) Протокол нима ?
- 2) Мурожаат ранги ?
- 3) WWW нинг хоҳлаган жойига мурожаат теги ?
- 4) *vlink* нинг вазифаси ?
- 5) `<a href>` ва `</a>` нинг вазифаси ?
- 6) *link* калит сўзининг вазифаси ?
- 7) Тасвирларни мурожаат сифатида ишлатиш ?
- 8) Фрейм нима ?
- 9) Фреймнинг асосий тегларини ?
- 10) *frameset* тегининг асосий вазифаси ?
- 11) *cols* нинг вазифаси ?
- 12) *rows* нинг вазифаси ?

## Лаборатория иши № 4

**Мавзу:** JavaScript ёрдамида соҳани тўғри тўлдирилганлигини текшириш

**Ишдан мақсад:** JavaScript ёрдамида соҳани тўғри тўлдирилганлигини текширишни ўрганиш

### Назарий қисм

Java script да формалар. Формалар интернетда кенг ишлатилади. Формага киритилган маълумотлар тез – тез қайта 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 қандай булишини курсатмоқчи эдик, кейин эса интернетга маълумот жунатишнинг қандай имкониятлари борлигини курсатамиз. Дастлаб биз оддий скрипт яратайлик. Айтайлик html-саҳифа 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 ҳеч нарса киритмаган булсангиз хатолик юз бериш ҳақида маълумот оласиз. 1 – элементдаги исталган маълумот тугрилик жихатдан текширилади. Албатта агар фойдаланувчи бошқа номни киритган булса у буларни қафолатламайди. Браузер хатоларни сонларга ҳам қаршилик 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 @ белгини узиди ҳам бу шартга тугри келади. интернетдаги ҳар бир электрон почта манзилида @ белги қатнашади. Шундай экан бу белгига текшириш бу ерда жуда уринли. Скрипт бу 2 та элемент билан қандай ишланди ва унда текшириш қай қилинади булади. Бу қуйидагича 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 майдонига 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 “ok” буйруги деб аталади. If буйрукги 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).

## **Лабаратория Ишни бажаришга 1-кўрсатма**

### **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 булмайти. Шу боис биз скриптимишни 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>
```

## **Лаборатория ишни бажариш бўйича**

### **2-кўрсатма**

#### **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 хоҳласангиз у холда сиз CGI интерфейсидан фойдаланишингиз керак. У сизга маълумотларни автоматик равишда кайта номлашга имкон беради. Масалан, сервер миждозлардан айримлари кира олиш имконияти булган маълумотлар базасини яратиш мумкин. Бошка мисол- yahoo 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-кувватловчи одамлар уларга курсатилган маълумотларни уқиш ва кидиришни кутиб утиришга хождат колмайди. Буларнинг барчасини сервернинг узи бажаради. Java S бундай имкониятларни бажаришга йул куймайди.

Java S маълумотларни сервердаги ыирор бир файлга ёзиб олиш имконияти булмагани сабабли у ёрдамида укув такризлари китобини яратаолмайсиз. Буни факат CGI интерфейси оркали бажариш мумкин.

Албатта, сиз фойдаланувчилар электрон почта оркали юборган маълумотлар учун такриз китоби яратишингиз мумкин. Лекин бу холда сиз электрон почта оркали олинган такризларни кулда битта файлга киритишингиз керак. Агар сиз хар куни 1000 такриз олишни назарда тутмасангиз шундай килишингиз мумкин. Шунга мос скрипт оддий HTML матндан иборат булади ва хеч кандай Java S дастурлаштириш керак эмас. Албатта битта холни хисобга олмаганда, яъни агар сизга маълумотларни юборишдан олдин текшириш керак булса бу ерда хакикатан хам Java S керак булади. Факат биз хамма ерда хам ишламайдиган mailto буйругини киритамиз, у факат Microsoft Explorer 3.0 да куллаб-кувватланмайди.

```
<form method=post action="mailto:your.address@goes.here"
  enctype="text/plain">
```

Нравится ли Вам эта страница?

Вовсе нет.<br>

СНЕСКО>Напрасная трата времени.<br>

Самый плохой сайт в Сети.<br>

</form>

Enctype="text/plain" параметри маттни худди узини хеч кодланган кисмларсиз юбориш учун ишлатилади. Бу сезиларли даражада бундай почтани укишни соддалаштиради.

Агар сиз формани сетга узатилмасдан текширишни истасангиз, у холда 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() {  
    Docuent.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.

**Топшириқ қуйидаги формаларга киритилган маълумотларни бўшлиққа, сон ва харфни фарқлаш дастурини тузинг**

### **Вариантар**

1. telefonga sms yuborish forumnini yaratish
2. e-mailga xat yuborish forumnini yaratish
3. Paynet to`lovi forumnini yaratish
4. passport ma`lumotlarini kiritish uchun forumnini yaratish
5. abiturient ma`lumotlarini kiritish forumnini yaratish
6. Telefon markalari haqidagi ma`lumotlarni kiritish forumnini yaratish
7. Avtomobillar markalari haqidagi ma`lumotlarni kiritish forumnini yaratish
8. Kompyuter sotish jarayonining formasini yaratish
9. Abutturent qabul qilish formasini yaratish
10. Ishga qabul qilish so`rovnomasining formasini yaratish
11. Ishga taqsimlash formasini yaratish
12. Avtomabillar ma`lumotlarini kiritish formasini yaratish
13. Aviaqassa formasni yaratish
14. Kasalxonalarda kasallik varaqasini yuritish formasni yaratish
15. e-mail yaratish formasini yaratish

Назорат саволари

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

## Лаборатория иши №-5

**Мавзу : *JavaScriptдан фойдаланган ҳолда объектлар яратиши***

**Ишдан мақсад : *JavaScriptдан фойдаланган ҳолда объектлар яратишини ўрганиши***

### НАЗАРИЙ ҚИСМ

**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 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 орқали 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 қилиш кераклигини хабар беради.

Берилаган мисол бир нечта янги хусусиятларга эга, уларни тартиб буйича куриб чиқамиз. Биринчи янги хусусият бу OnClick=”alert(‘Y<sub>0</sub>’)” булиб у <input> тегига тегишли. Айтиб утганимиздек, бу тугмани босганингизда нима содир булишини аниқлайди. Шу тарика, агар Click ходисаси юз берадиган булса, компьютер alert(Y<sub>0</sub>) чакиригини бажариши керак. Бунинг узи JavaScript тилидаги кодга мисол булади (Шунга эътибор қаратингки, бу ҳолатда биз хатто <script> тегида фойдаланмадик).

Alert функцияси тушувчи дарчаларни ҳосил қилиш имконини беради. Уни чакиришда сиз қавслар ичига бирор сатр беришингиз керак. Бизнинг ҳолатда бу ‘Y<sub>0</sub>’. Ва бу худди шу матн тушаётган дарчада пайдо булади. Шу тарика, укувчи тугмачани шикирлатган пайтда, бизнинг скрипт ‘Y<sub>0</sub>’ матнли дарча ҳосил қилади.

Берилган мисолда бир канча чалкашликлар тугилиши мумкин: биз document.write() буйругида жуфт куштирнок, alert() конструкциясида эса бирлик куштирнок ишлатдик. Нега? Қупчилик ҳолларда 2 та ҳолдан ҳам фойдаланиш мумкин. Бирок, сунги мисолда биз OnClick=”alert (‘Y<sub>0</sub>’)”да бирлик ва 2 лик куштирнок ишлатдик. Агар биз OnClick=”alert (‘Y<sub>0</sub>’)” деб ишлатганда компьютер будай скриптга тушунмаган булар эди. Ва кайси конструкция OnClick функциясига тегишли, кайсиси 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 куйидагича ёзишингиз мумкин: `OnClick='alert ("Ү0")'`.

## ФУНКЦИЯЛАР.

JavaScript тилида тузилган купчилик дастурларда функциялардан фойдаланамиз. Шунинг учун хозир бу тилнинг мухим элементи хакида гапириш жуда мухим. Функциялар купчилик холларда узида бир неча буйрукларни боглаш усулини номоён килади. Келинг мисол учун бирор бир маттни уч марта бирданига чон килувчи скрипт ёзамиз. Дастлаб оддий йулни курайлик:

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

Ва бу скрипт кушимча маттни босиб чикаради:

*Менинг сахифамга хуш келибсиз!*

*Бу JavaScript!*

*Бу JavaScript!*

*Бу JavaScript!*

Агар дастлабки кодига карайдиган булсак, керакли натижани олиш учун коднинг аник бир кисми 3 марта такрорланган. Ахир бу самаралими?

Йук биз ушбу масалани осонирок ечишимиз мумкин:

```
< html >
< script language= "JavaScript">
< ! - - hide
function myfunction () {
document. Write( `Менинг сахифамга 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 буйрук бирлаштирилди ва улар функция чакирилиши билан биргаликда бажарилади. Хакикатдан хам бизнинг мисолимизда функция 3 марта чакирилган ва буни функциянинг узи аниклагандан сунг myfunction() сатрини 3 марта ёзганимизни куришимиз мумкин. Ва уз навбатида бу шуни билдирадики, функциянинг таркибидаги нарсалар (буйруклар, { } 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 курамиз.

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

```
<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. Сизга 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 ҳакозаларни саклаш учун ишлатилади. Скриптинг 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 булаемиз.

## 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 услуга 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 курамиз:

```
<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 қандай ташкил этилганлигини билишимиз шарт. Қурииб турибдики иерархик тузилишга кура ҳар бир объект уз номига эга. Мас равишда агар сиз html саҳифадаги 1-расмга мурожат қилмоқчи булсангиз объектлар иерархиясидаги уз йуналишингизни танлаб олишингиз керак ва энг юқори чуққидан 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 қандай матн киритилганини билмоқчи булсангиз датслаб қандай қилиб объектга рухсат олишни аниқлаб олишингиз керак ва яна объектлар иерархиясининг энг

юкорисидан бошлаймиз. Сунгра `elements[0]` объектга йул курсатамиз ва кетма-кетлик билан барча объектларни номини ёзамиз ва натижада шу нарса аникланадики 1-майдонга рухсатни шуларни ёзиш оркали 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 хақидаги 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' саломи 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 мурожаат 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 содалаштиради, 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 ҳам эгасиз.

## Лаборатория ишни бажариш учун кўрсатма

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 мухим 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 хам худди шу адресга мос тушади. Шундай булсада биз учун жуда мухим 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 >
```

Топшириклар

#### Назорат саволлари

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

## Лаборатория иши № 6

**Mavzu:** My Sql da ma`lumotlar bazasi tashkil etish

**Ishdan maqsad:** My Sql da ma`lumotlar bazasi tashkil etish

### NAZARIY QISM

Quyida qanday qilib klient dasturi mysql ga ulanishni ko`ramiz. Bu dastur yordamida MySQL-serverga ulanish, SQL-so`rovlarni bajarish vash u so`rovlar natijalarini ko`rib chiqish mumkin. Bu qsimni o`rganish uchun kompyuteringizda utilita mysql o`rnatilgag va MySQL serveri bilan boo`langan bo`lishi kerak.

MySQL serveriga mysql dasturi yordamida bog`lanish uchun foydalanuvchi nomini va odatda parol kiritish lozim. Agar server va klient har xil mashinalarda joylashgan bo`lsa, MySQL serveri ishga tushirilgan xost nomini ko`rsatish lozim:

```
shell> mysql -h host -u user -p
```

SHundan so`ng ekranda quyidagi so`rov paydo bo`ladi Enter password:, va sizga o`z parolingizni kiritishingiz kerak bo`ladi. Agar ulanish to`g`ri amalga oshgan bo`lsa, ekranda quyidagi ma`lumot va komanda satri belgisi paydo bo`ladi mysql>:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 459 to server version:
```

```
Type 'help' for help.
```

```
mysql>
```

Quyidagi mysql> belgining paydo bo`lishi mysql dasturi ishga tayyorligini bildiradi.

Serverdan ixtiyoriy paytda QUIT komandasini terib uzilish mumkin:

```
mysql> QUIT
```

```
Be
```

#### ***Izox***

*Odatda MySQL lokal mashinaga Yangi o`rnatilgan bo`lsa, murojaat parol va xost kiritilmasdan, komana qatorigamysql komandasini kiritish yo`li Bilan amalga oshiriladi.*

Serverga ulangandan so`ng komandalar sintaksisini o`rganish uchun bir necha sodda so`rovlar berishingiz mumkin. Hali hech qanday ma`lumotlar bazasi tanlanmagani uchun quyida keltirilgan so`zrovlar umumsiy harakterga ega.

#### ***MySQL ma`lumotlar bazasida ishlatiladigan ma`lumotlar tiplari***

Butun sonlar

Ma`lumotlar tipini ko`rsatish umumiy formasi:

*prefiks*INT [UNSIGNED]

Shart bo'lmagan bayroq UNSIGNED iorasiz sonlar (o' ga teng yoki katta) sonlar saqlash uchun maydon yaratishni bildiradi.

TINYINT	Diapazoni -128 dan 127 gacha sonlarni saqlashi mumkin
SMALLINT	Diapazoni -32 768 dan 32 767 gacha sonlarni saqlashi mumkin
MEDIUMINT	Diapazoni -8 388 608 dan 8 388 607 gacha sonlarni saqlashi mumkin
INT	Diapazoni -2 147 483 648 dan 2 147 483 647 gacha sonlarni saqlashi mumkin
BIGINT	Diapazoni -9 223 372 036 854 775 808 dan 9 223 372 036 854 775 807 gacha sonlarni saqlashi mumkin

#### Kasr sonlar

MySQL da butun sonlar bir necha turga ajratilganidek, kasr sonlar ham bir necha turga ajratiladi.

Umumiy xolda ular quyidagicha yoziladi:

*TipNomi*[(length, decimals)] [UNSIGNED]

Bu erda – kasr uzatilishda joylashadigan belgi joylari soni (maydon kengligi). *decimals* – o'nli nuqtadan so'ng hisobga olinuvchi raqamlar soni. UNSIGNED – ishorasiz sonlarni beradi.

FLOAT	Aniqligi uncha katta bo'lmagan suzuvchi nuqtali son.
DOUBLE	Ikkilik aniqlikka ega bo'lgan suzuvchi nuqtali son.
REAL	DOUBLE uchun sinonim.
DECIMAL	Satrlar shaklida saqlanuvchi kasr son.
NUMERIC	DECIMAL uchun sinonim.

#### Satrlar

Satrlar simvollar massivlaridan iborat. Odatda *SELECT* so'rovi bo'yicha matnli maydonlar bo'yicha izlashda simvollar registri hisobga olinmaydi, Ya'ni "Vasya" va "VASYA" satrlari bir xil hisoblanadi. Agar ma'lumotlar bazasi matni joylashtirish va

o'qishda avtomatik qayta kodlashga sozlangan bo'lsa, bu maydonlar siz ko'rsatgan kodlashda saqlanadi.

Oldiniga *length* dan oshmagan simvollar saqlovchi satrlar tiplari Bilan tanishamiz., *length* prinadlejit diapazonu ot 1 dan to255 gacha bo'lgan diapazonda yotadi.

VARCHAR (length) [BINARY]

Bu tipdagi maydonga biror qiymat kiritilganda undan oxirini ko'rsatuvchi simvollar avtomatik ravishda qirqib olinadi. Agar BINARY bayrog'i ko'rsatilgan bo'lsa, SELECT so'rovda satr registri hisobga olgan xolda solishtiriladi.

VARCHAR 255 dan ortiq bo'lmagan simvollarni saqlashi mumkin.

TINYTEXT 255 dan ortiq bo'lmagan simvollarni saqlashi mumkin.

TEXT 65 535 dan ortiq bo'lmagan simvollarni saqlashi mumkin.

MEDIUMTEXT 16 777 215 dan ortiq bo'lmagan simvollarni saqlashi mumkin.

LONGTEXT 4 294 967 295 dan ortiq bo'lmagan simvollarni saqlashi mumkin.

Ko'pincha TEXT tpi qo'llanadi, lekin ma'lumotlar 65 536 simvoldan oshmasligiga ishonmasangiz, LONGTEXT tipidan foydalaning.

#### Binar ma'lumotlar

Binar ma'lumotlar – TEXT formatidagi ma'lumotlarning o'zi, lekin ularda izlashda simvollar registri hisobga olinadi.

TINYBLOB 255 dan oshmagan simvollarni saqlashi mumkin.

BLOB 65 535 dan oshmagan simvollarni saqlashi mumkin.

MEDIUMBLOB 16 777 215 dan oshmagan simvollarni saqlashi mumkin.

LONGBLOB 4 294 967 295 dan oshmagan simvollarni saqlashi mumkin.

BLOB-ma'lumotlar avtomatik qayta kodlanmaydi, agar o'rnatilgan ulanish Bilan ishlaganda drxol qayta kodlash imkoniyati o'rnatilgan bo'lsa.

Sana va vaqt

MySQL sana va vaqtni har xil formatlarda saqlash uchun mo'ljallangan maydonlar bir necha tiplarini qo'llaydi.

DATE GGGG-MM-DD formatdagi sana

TIME CHCH:MM:SS formatdagi vaqt

DATETIME GGGG-MM-DD CHCH:MM:SS formatdagi sana va vaqt

MP TIMESTA olishda u timestamp formatida emas, GGGGMMDDCHCHMMSS formatda aks etadi, bu esa PHP dan undan foydalanish qiymatini ancha kamaytiradi.

### *Ma'lumotlar bazasini yaratish MySQL (CREATE DATABASE)*

Ma'lumotlar bazasi CREATE DATABASE komandasi yordamida yaratiladi.

Komanda sintaksisi :

CREATE DATABASE *database\_name*

□ *database\_name* – Ma'lumotlar bazasiga beriladigan nom.

Keyingi misolda db\_test ma'lumotlar bazasini yaratamiz:

```
CREATE DATABASE db_test
```

PHP da ma'lumotlar bazasini yaratish:

```
$sql="CREATE DATABASE db_test";
```

```
mysql_query($sql);
```

### *MySQL ma'lumotlar bazasini o'chirish (DROP DATABASE)*

Ma'lumotlar bazasini o'chirish uchun **DROP DATABASE** komandasidan foydalaniladi.

Sintaksis:

**DROP DATABASE** *database\_name*

Bu erda

□ *database\_name* – o'chirish kerak bo'lgan ma'lumotlar bazasi nomi.

Quyidagi misolda db\_test ma'lumotlar bazasi o'chiriladi:

```
DROP DATABASE db_test
```

PHPda ma'lumotlar bazasini o'chirish:

```
$sql="DROP DATABASE db_test";
```

```
mysql_query($sql);
```

## **USE**

Jadvallar bilan ishlash uchun MySQL ga qaysi baza bilan ishlash haqida ma'lumot berish kerak. Bu **USE** komandasi yordamida amalga oshiriladi:

```
USE db_name;
```

Bu erda `db_name` – tanlangan ma'lumotlar bazasi nomi. Yaratilgan `db_test` bazasini tanlaymiz:

```
mysql> CREATE DATABASE db_test;  
Database changed;
```

### **MySQL ma'lumotlar bazasida jadval yaratish (CREATE TABLE)**

**Jadval yaratish CREATE TABLE komandasi orqali amalga oshiriladi.**

```
CREATE TABLE table_name(column_name1 type, column_name2 type,...)
```

- `table_name` – Yangi jadval nomi;
- `column_name` – yaratilayotgan jadval ustunlari (maydonlari), nomlari.
- `type` – ustun tipi.

Do'stlaringiz telefon nomerlari jadvalini yaratish kerak bo'lsin.

Bizning jadvalimiz uch ustundan iborat bo'ladi: Do'stingiz ismi sharifi, adres va telefoni

```
CREATE TABLE tel_numb(fio text, address text, tel text)
```

PHP da bu quyidagi ko'rinishga ega bo'ladi:

```
$sql="CREATE TABLE tel_numb(fio text, address text, tel text)";  
mysql_query($sql);
```

Ma'lumotlar turlariga mos ustunlar bilan bajarish mumkini bo'lgan (yoki taqiqlangan) operatsiyalarni ko'rsatuvchi modifikatorlarni ulash mumkin.

*not null* – Maydon noma'lum qiymatga ega bo'la olmasligini, Ya'ni jadvalga Yangi yozuv qo'shishda maydon albatta initsializatsiya qilinishi kerakligini (agar ko'ozda tutilgan qiymat berilmagan bo'lsa) ko'rsatadi.

Masalan, bizning telefonlar jadvalimizda do'stimiz ismi sharifi(maydon fio) va telefoni (maydon tel) maydonlari noma'lum qiymatga ega bo'la olmasligini ko'rsatish kerak:

```
CREATE TABLE tel_numb(fio text NOT NULL, address text, tel text NOT NULL)
```

*primary key* - Maydon birlamchi kalitligini, Ya'ni ilova qilish mumkin bo'lgan yozuv identifikatori ekanligini aks etadi.

```
CREATE TABLE tel_numb(fio text, address text, tel text, PRIMARY KEY (fio))
```

*auto\_increment* – Maydonga Yangi yozuv qo'shishda maydon unikal qiymat qabul qiladi va jadvalda hech qachon bir xil nomerli maydonlar mavjud bo'lmaydi.

```
CREATE TABLE tel_numb(fio text AUTO_INCREMENT, address text, tel text)
```

*default* – maydon uchun ko'zda tutilgan qiymatni aniqlaydi. Agar joylanayotgan yozuvda bu maydon uchun qiymat ko'rsatilmagan bo'lsa, shu

qiymat kiritiladi.

```
CREATE TABLE tel_numb(fio text, address text DEFAULT 'Ne ukazan', tel text)
```

### ***SHOW Komandalar***

Ma'lumotlar bazasi muvaffaqiyatli yaratiganini tekshirish uchun, kompyuteringizda qanday ma'lumotlar bazasi mavjudligini ko'rsatuvchi `SHOW DATABASES`, komandasini bajarish mumkin:

```
mysql> SHOW DATABASES;
```

Hamma jadvallar muvaffaqiyatli yaratilganiga ishonch xosil qilish uchun, `SHOW TABLES` komandasini bajaramiz.

Tanlangan jadval hamma ustunlari ro'yxatini quyidagi so'rov yordamida chiqarish mumkin:

```
mysql> SHOW FIELDS FROM tel_numb;
```

### ***DESCRIBE***

`DESCRIBE` Komandasi yaratilgan jadvallar strukturasi ko'rsatadi va quyidagi sintaksisga ega:

```
DESCRIBE table_name
```

Bu erda `table_name` – strukturasi so'ralayotgan jadval.

*DESCRIBE* Komanda SQL standartiga kirmaydi va MySQL ichki komandasidir.

Keling quyidagi SQL- so'rov bajarib `forums`, jadvali strukturasi ko'ramiz:

```
mysql> DESCRIBE tel_numb;
```

### ***MySQL ma'lumotlar bazasidan jadvalni o'chirish (DROP TABLE)***

Jadvalni **O'chirish** uchun **DROP TABLE** komandasidan foydalaniladi

```
DROP TABLE table_name
```

□ `table_name` – o'chirilayotgan jadval nomi.

```
DROP TABLE tel_numb
```

PHP da bu quyidagi ko'rinishga ega bo'ladi:

```
$sql="DROP TABLE tel_numb";
```

```
mysql_query($sql);
```

### ***Jadval xossalari o'zgartirish: Jadvalni qayta nomlash (ALTER TABLE RENAME)***

Jadvalga Yangi nom berish quyidagi konstruktsiya yordamida amalga oshirilishi mumkin:

```
ALTER TABLE table_name_old RENAME table_name_new
```

bu erda

- *table\_name\_old* - jadval eski nomi;
- *table\_name\_new* – jadval Yangi nomi.

Misol uchun *search* jadvali nomini *search\_en* nomiga o'zgartirish kerak bo'lsin:

```
$sql="ALTER TABLE search RENAME search_en";  
mysql_query($sql);
```

#### *Jadvallar xossalarini o'zgartirish: Ustunlar qo'shish (ALTER TABLE ADD)*

Yangi ustun qo'shishni quyidagi konstruktsiya yordamida amalga oshirish mumkin:

```
ALTER TABLE table_name ADD field_name paramtrs  
gde
```

- *table\_name* – Yangi ustun qo'shiladigan jadval nomi;
- *field\_name* – qo'shilayotgan ustun nomi;
- paramtrs** – qo'ilyotgan ustunni tasvirlovchi parametrlar.

Ma'lumotlar tipini ko'rsatish majburiy parametrdir.

Masalan, *my\_friends* nomli jadvalga *adress\_2* nomli matn qiymatlarga ega ustun qo'shishimiz kerak bo'lsin:

```
$sql="ALTER TABLE my_friends ADD adress_2 TEXT";  
mysql_query($sql);
```

Ko'zda tutilgan bo'yicha Yangi ustun jadval oxiriga qo'shiladi. Agar ustun jadval boshiga qo'shilishi kerak bo'lsa, qo'shilayotgan ustun parametrlaridan so'ng **FIRST** kalit so'zini yozish kerak:

```
$sql="ALTER TABLE my_friends ADD adress_2 TEXT FIRST";  
mysql_query($sql);
```

Agar ustun jadval boshi yoki oxiri emas, balkit ma'lum ustundan keyin qo'yilishi lozim bo'lsa quyidagi kalit so'zdan foydalanish lozi **AFTER** *ustun nomi*, *shu ustundan so'ng Yangi ustun qo'shiladi*:

```
$sql="ALTER TABLE my_friends ADD adress_2 TEXT AFTER adress_1";  
mysql_query($sql);
```

Bu misolda Yangi *adress\_2* ustuni *adress\_1* ustunidan keyin qo'yiladi.

Agar jadvalga bir emas bir necha ustun qo'shish kerak bo'lsa har bir ustun uchun **ADD** *field\_name* paramtrs vergul orqali yozish kerak:

```
$sql="ALTER TABLE my_friends ADD adress_2 TEXT,  
ADD adress_3 TEXT, ADD adress_4 TEXT";  
mysql_query($sql);
```

Agar jadvalga ikki ustun qo'shish lozim bo'lsa, quyidagicha amalga oshirish

mumkin:

```
$sql="ALTER TABLE my_friends ADD address_2 TEXT AFTER address_1,  
ADD address_3 TEXT AFTER address_2";  
mysql_query($sql);
```

Ya'ni birinchi qo'shilayotgan ustunni *address\_1* dan so'ng, ikkinchisini birinчисidan so'ng.

#### ***Jadval xossalarini o'zgartirish: Ustun xossalarini o'zgartirish (ALTER TABLE CHANGE)***

Bir yoki bir necha ustunlar xossalarini quyidagi konstruktsiya yordamida o'zgartirish mumkin:

```
ALTER TABLE table_name CHANGE field_name_old field_name_new  
parametrs  
gde
```

- table\_name* – o'zgartirilayotgan ustun joylashgan jadval nomi;
- field\_name\_old* – o'zgartirilayotgan ustun nomi;
- field\_name\_new* – o'zgartirilayotgan ustun Yangi nomi (agar ustun nomi o'zgartirilmasa *field\_name\_old* ga teng);
- parametrs – ustun Yangi parametrlari.

Keyingi misolda *field\_1* tipini matn sifatida o'zgartiramiz:

```
$sql="ALTER TABLE my_table CHANGE field_1 field_1 TEXT";  
mysql_query($sql);
```

Agar qo'shimcha ustun nomini *field\_2* deb o'zgartirish kerak bo'lsa:

```
$sql="ALTER TABLE my_table CHANGE field_1 field_2 TEXT";  
mysql_query($sql);
```

Agar bir necha ustun xossalarini birdaniga o'zgartirish lozim bo'lsa, **CHANGE** *field\_name\_old field\_name\_new* parametrs konstruktsiyani vergul bilan har bir ustun uchun qaytaramiz:

```
$sql="ALTER TABLE my_table CHANGE field_1 field_2 TEXT,  
CHANGE field_3 field_3 TEXT";  
mysql_query($sql);
```

#### ***Jadval xossalarini o'zgartirish: Ustunlarni o'chirish (ALTER TABLE DROP)***

Ustunni quyidagi konstruktsiya yordamida o'chirish mumkin:

```
ALTER TABLE table_name DROP field_name
```

bu erda

- table\_name* – ustunni o'chirilayotgan jadval nomi;
- field\_name* – o'chirilayotgan ustun nomi.

```
$sql="ALTER TABLE search DROP id_num";
```

```
mysql_query($sql);
```

Agar biz birdaniga bir necha maydonlarni o‘chirmoqchi bo‘lsak, **DROP** *field\_name* konstruktsiyani vergul bilan har bir ustun uchun qaytaramiz:

```
$sql="ALTER TABLE search DROP id_1, DROP id_2, DROP id_3";
```

```
mysql_query($sql);
```

### ***Jadvalga satrlar qo‘shish (INSERT INTO)***

**YOzuvlarni joylash** uchun **INSERT INTO** komandasidan foydalaniladi.

```
INSERT INTO table_name(field_name1, field_name2,...) values('content1',  
'content2',...)
```

Bu komanda *table\_name* jadvaliga *field\_nameN* maydonlariga *contentN* qiymat o‘rnatilgan yozuv qo‘shadi.

Masalan, agar biz adreslar va telefonlar (FIO, adres, telefon) yaratmoqchi bo‘lsak, quyidagi kodni yozishimiz kerak:

```
CREATE TABLE tel_numb(fio text, address text, tel text)
```

*tel\_numb* jadvaliga qiymatlarni quyidagicha joylash mumkin:

```
INSERT INTO tel_numb(fio, address, tel)
```

```
values('Vailiy Ivanov', 'ul.Gorkogo, d.18', '23-23-23')
```

Joylash komandasida ko‘rsatilmagan maydonlar "aniqlanmagan" qiymatlar oladi (aniqlanmagan qiymat - bu bo‘sh satr emas, balki MySQL ga, shu maydonningnet hech qanday qiymati yo‘qligini bildiruvchi belgidir).

Agar jadval yaratilayotganda maydon NOT NULL bayrog‘i bilan belgilangan bo‘lsa va u yozuv joylashda qiymat olmasa, MySQL xato haqida ma’lumot qaytaradi.

Jadvalga binar ma’lumotlarni (apostrof va sleshlarni o‘z ichiga olgan satrlarni) joylashda ba’zi simvollar teskari sleshtar bilan, Ya’ni \, ' simvol iva nolkodga ega simvolbilan ximoyalangan bo‘lishi kerak.

### ***Jadvaldan satrlarni o‘chirish (DELETE FROM)***

**YOzuvni o‘chirish** uchun **DELETE FROM** komandasi ishlatiladi

**DELETE FROM** *table\_name* **WHERE** (*ifoda*)

Bu komanda *table\_name* jadvalidan *ifoda* bajarilgan hamma yozuvlarni o‘chiradi.

*ifoda* - bu oddiy mantiqiy ifoda.

Masalan FIO, adres va telefonni o‘z ichiga olgan jadvaldan yozuvni o‘chirish:

```
DELETE FROM tel_numb WHERE (fio='Vasiliy Ivanov')
```

yoki, bir necha parametr bo‘yicha o‘chirish kerak bo‘lsa

```
DELETE FROM tel_numb WHERE (fio='Vasiliy Ivanov' && tel='23-45-45')
```

Ifodalarda maydonlarning nomlari, konstantalar va operatorlardan tashqari, soddalari hisoblanuvchi qismlar kelishi mumkin, masalan: (id<10+4\*5).

Bizda mexmonlar kitobi ma'lumotlarni saqlash uchun MySQL ma'lumotlar bazasidan foydalansin.

Jadval (nomli *db\_guest*), qoldirilgan ma'lumotlarni o'z ichiga olib, quyidagi tarkibga ega:

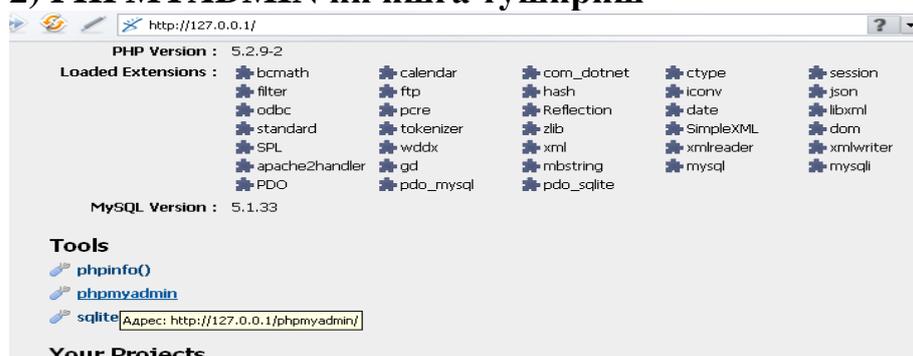
- id* - maydon yozuidentifikatsion unikal nomeri;
- name* – ma'lumot qoldirgan foydalanuvchi nomeri;
- mail* – foydalanuvchi e-mayli;
- url* - foydalanuvchi URL li;
- content* – ma'lumotning o'zi.

## Лаборатория ишни бажаришга кўрсатма Талаба МБни яратиш бўйича

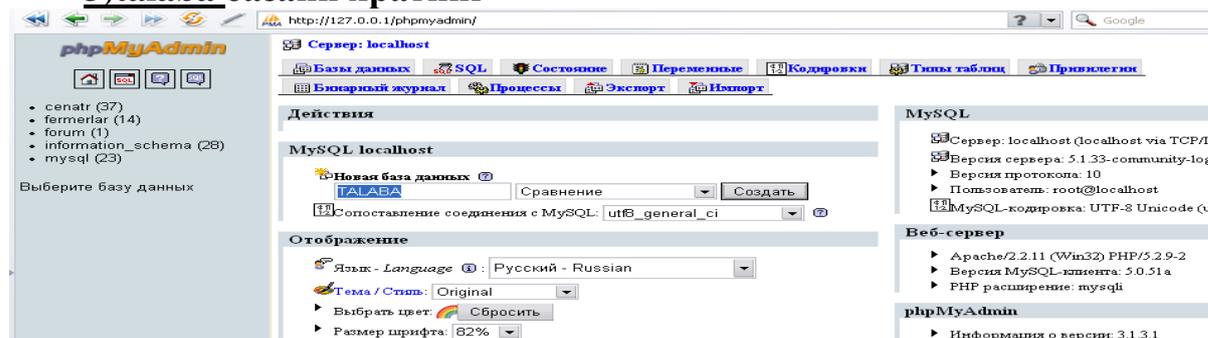
### 1) Wamp серверни START қилиш керак



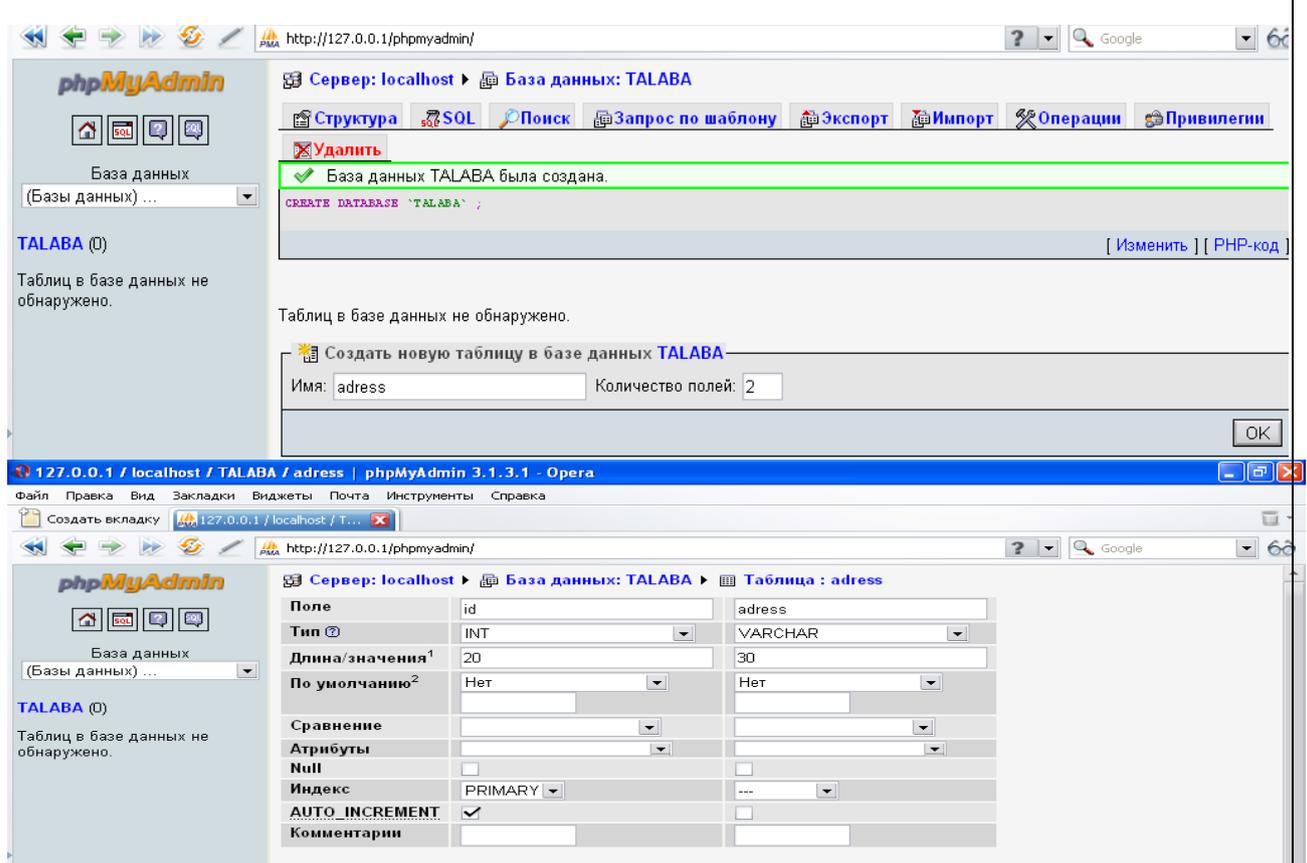
### 2) PHPMYADMIN ни ишга тушириш



### 3) talaba базани яратинг



### 4) Яратилган talaba базасига жадваллар яратинг



## Топширик Барилган вараиннинг маълумотлар базасини яратинг

### Варианлар

- 1) компьютер дўкони
- 2) Дастурий таъминотлар мажмуининг
- 3) Махалла фуқароларининг МБ
- 4) Интернет карточка билан савдо қилувчи дўкон МБ
- 5) мехнонхона МБ
- 6) талаба рейтингини аниқлашнинг МБ
- 7) кадрлар бўлими МБ
- 8) Маниторинг бўлими МБ
- 9) Интернет магазин МБ
- 10) Ўзбек Таомларни тайёрлашнинг МБ
- 11) Барча олий таълимлар ҳақидаги ахборот МБ
- 12) Барча касб-хунар таълимлар ҳақидаги ахборот МБ
- 13) Интернет Аптека МБ
- 14) Фермер МБ
- 15) мева дўкони МБ

### Nazorat savollari

- 1) 1 MY SQL ning imkoniyatlari ?
- 2) MYSQL da qanday tiplar ishlatiladi?
- 3) *MYSQL* ning asosiy funksiyalari?
- 4) *SELECT* operatorining vazifasi ?
- 5) *INSERT* operatorining tuzilish qanday ?
- 6) *DROP* operatorining vazifasi ?
- 7) *CREATE* operatorining vazifasi nimada ?
- 8) *CREATE* operatorining vazifasi nimada ?

## Лаборатория иши № 7

**Мавзу** PHP билан MY SQL маълумотлар базаси боғлаш

**Ишдан мақсад** : PHP билан MY SQL маълумотлар базаси боғлашни ўрганиш

### НАЗАРИЙ ҚИСМ

PHP тилига 1995 йил асос солинган. Шу даврда Лердарор номли дастурчи Perl/CGI курунишидаги сценарийни тузди. Бу сценарийни у узининг саҳифасига кировчилар сонини аниклаш мақсадида тузди. Шу дастур PHP тилининг яратилишига туртки булди.

1997 йилдан бошлаб у PHP деб атала бошлади. PHP тилини Nuper Text Processor PHP техник нуктаи назаридан караганда сервер томонидан туриб ишлайдиган сценарийлар тилидир. PHP тили дастурчиларга узининг кулайлиги, ишончилиги билан манзур булган. PHP тилининг асосий устунлигидан бири уни тутридан-тутри html нинг ичига жойлаштира олишимиздир.

Куйидаги мисолда PHP билан html тэглари аралашиб келади:

```
<html>
<title><?php print "Бу PHP"; ?></title>
<body>
  <?php
    print "<h3> Яна PHP </h3>";
  ?>
  <h4>Бу HTML</h4>
  <?php
    print "<h1>Яна PHP";
    print "<h6>2-сатр</h6>";
  ?>
</body>
</html>
```

PHP тилидаги дастурлар ишлаши учун компьютерга Apache номли web сервер урнатилиши керак. Одатда Apache, PHP, MySQL учаласи биргаликда ишлатилади. Бу ерда MySQL маълумотлар базасини бошкариш учун ишлатиладиган сервер

дастури.

## PHP га утиш

Бу жараен escaping to PHP деб юритилади.

PHP га утишнинг 4 та варианты бор:

- 1) Стандарт тэглр;
- 2) Киска тэглр;
- 3) Script тэги;
- 4) ASP стилидаги тэглр.

Купчилик холларда дастурчилар стандарт тэгдан фойдаланади. У куйидагича:

```
<?php  
    print "Бу PHP";  
?>
```

Куриниб турганидек PHP нинг бошланиши <?php билан очилади, ?> билан епилади.

Киска тэглр куйидагича езилади:

```
<?  
    print "Бу PHP";  
?>
```

Лекин стандарт тэглрдан фойдаланган максадга мувофик. Чунки серверда киска тэглр ажратилган булиши мумкин. Шунинг учун турли хил хатолик булади.

Агарда html редакторлар php кодни хато деб кабул килса, у холда куйидаги script тэги ердамида php код кушилади. Бу холат камдан-кам холларда юз беради.

```
<script language="php">  
    <?php  
        print "Бу PHP";  
    ?>  
</script>
```

## Узгарувчилар

PHP да узгарувчилар \$ (доллар) белгиси билан бошланади. Бу белгидан кейин лотин алифбосининг харфлари, кейин араб ракамлари ва "\_" белгилар кетма-

кетлиги булиши мумкин. Лекин \$ белгисидан кейин бирданига ракам билан бошлаш мумкин эмас. Масалан, \$b, \$a\_3,\$abbos. Хато узгарувчилар: \$20, \$, A, \$92.

PHP да ўзгарувчиларнинг номлари катта, кичиклиги фаркланади. Масалан, \$A≠\$a булар алохида ўзгарувчилар.

Ўзгарувчига қиймат бериш учун тенг "=" белгиси ишлатилади.

```
<?php
    $a=10;$s='мактаб';
    echo $a;
?>
```

PHP да ўзгарувчилар қабул қилган қийматларининг типи бўйича аниқланади. PHP кучли типланмас дастурлаш тиллари синфига киради. Масалан, Perl? Basic лар каби.

Масалан, қуйидаги дастурни ёзсақ;

```
<?php
    echo `100`+1;
?>
```

Дастур ишлаганда натижа 101 бўлади. PHP транслятори сатрли катталикни автоматик равишда сонга айлантириб сон билан қўшади.

```
<?php
    $a='мактаб';
    $b='келдим';
    $A="$a,$b";// мактаб,келдим
    echo $A;
?>
```

Бу script бажарилгандан кейин мактаб,келдим ҳосил бўлади. Қўштирноқ ичида махсус бошқарувчи белгилар ҳам берилиши мумкин.

\r-enter;

\n-янги сатр;

\"-"-чизикча белгиси;

\\$-доллар белгиси;

\\-\ белгиси.

Юқоридаги мисолни бошқачароқ кўринишда ёзамиз:

```
<?php
    $a='мактаб';
    $b='келдим';
    $A=$a.';'.$b.'!';// мактаб,келдим!
    echo $A;
?>
```

Олдинги дастурнинг нтижаси билан бир хил. Фақат сатрларни қўшиш учун махсус “.” белгиси ишлатилган. Кўпчилмк холларда шу белгини шилатиш мақсадга мувофиқ. Агар биз

```
$A="$ab";
```

деб ёзсак, мактабб сўзи чиқади.

Агарда бир неча ўзгарувчиларга бир хил қиймат бериш керак бўлса, қуйидагича ёзиш мумкин:

```
$a=$c=10.35;
```

### Шартли операторлар

Дастурлаш тилларидаги каби PHP да ҳам шартли жараёнларни дастурлаш учун шартли операторлар мавжуд. Буни қуйидаги мисолда кўрамиз:

```
<?php
    $a=25;$b=5*5;
    if ($a==$b)
        echo '$a ва $b тенг';
    else
        echo '$a ва $b тенг эмас';
?>
```

Бу ерда if else кўринишидаги шартли оператор ишлатилган. Унинг умумий кўриниши қуйидагича:

```
if (шарт)
```

```
ифода1;  
else  
ифода2;
```

if дан кейин тенглик текширилади. Бунда “==” тенг маъносида, “!=” тенг эмаслигини билдиради.

Шарт текшириш оператори куйидаги кўринишда ҳам ишлатилиши мумкин:

```
<?php  
$a=25;$b=5*5;  
if ($a==$b)  
    echo '$a ва $b';  
?>
```

Агар шартнинг акс ҳолда қисми қатнашмаса юқоридагича ишлатилади.

Бирор шарт бажарилганда бита ифода эмас, бир нечта оператор бажариладиган бўлса, {} кавслардан фойдаланилади

## Лабаратория ишини бажариш бўйича кўрсатма

Ма’умотлар базаси ҳамма yozuvlarini chiqaruvchi va qolldirilgan ma’lumotlarga o‘zgartirish kiritishga imkon beruvchi stsenariy (PHP da) yozamiz:

**guest\_update.php** fayli listingi

```
<html>  
<head>  
<title>Mexmonlar kitobidagi yozuvlarni Yangilash.</title>  
</head>  
<body>  
<?  
// Ma’lumotlarbazasiga ulanamiz  
mysql_connect("localhost", "root", "");  
mysql_select_db("test");  
  
// Agar yozuvni o‘zgartirish tugmasini bosgan bo‘lsak  
if(@$_submit_update) {  
    // Tanlangan yozuvni Yangilaymiz  
    $sql="update db_guest set name='$name', mail='$mail', url='$url', content='$content'  
        where (id='$update)";  
    mysql_query($sql);  
};
```

```

// O'zgaruvchiga <I>$result</I> butun qoldirilgan yozuvlar bazasini yozamiz
$sql="select * from db_guest";
$result=mysql_query($sql);
// Mexmonlar kitobida yozuvlar sonini aniqlaymiz
$rows=mysql_num_rows($result);
echo "<table border=0 align=center>";
echo "<tr><td align=center><B>Imya</B></td>";
echo "<td align=center><B>E-mayl</B></td>";
echo "<td align=center><B>URL</B></td>";
echo "<td align=center><B>Ma'lumot</B></td>";
echo "<td align=center><B>O'zgartirish</B></td></tr>";
for($i=0;$i<$rows;$i++) {
    // kursorni kerakli pozitsiyaga o'rnatamiz
    mysql_data_seek($result,$i);
    // Assotsiativn massivga <I>$arr_guest</I>
    // mexmonlar kitobi maydonlari qiymatlarini yozamiz
    $arr_guest=mysql_fetch_array($result);
    echo "<form method=get action='guest_update.php'>";
    echo "<input type=hidden name='update' value='".$arr_guest['id']."'>";
    echo "<tr>";
    echo "<td><input type=text name='name' value='".$arr_guest['name']."'></td>";
    echo "<td><input type=text name='mail' value='".$arr_guest['mail']."'></td>";
    echo "<td><input type=text name='url' value='".$arr_guest['url']."'></td>";
    echo "<td><textarea name='content'>".$arr_guest['content']."</textarea></td>";
    echo "<td><input type=submit name='submit_update' value='Izmenit'></td>";
    echo "</tr>";
    echo "</form>";
};
echo "</table>";
echo "</form>";
?>
</body>
</html>

```

### *Jadvalda yozuvlarni izlash (SELECT)*

**Jadvallardan izlash** uchun **SELECT** komandasidan foydalaniladi

**SELECT \* FROM *table\_name* WHERE (*ifoda*) [order by *field\_name* [desc][asc]]**

Masalan, bizga mexmonlar kitobidagi ma'lumotlar saqlanuvchi jadvalda ma'lum foydalanuvchi qoldirgan hamma ma'lumotlarni topish kerak bo'lsin.

```
// Foydalanuvchi nomi
```

```

$user="Admin";
// name – mexmonlar kitobida ma'lumot qoldirgan foydalanuvchilar nomlari
// saqlanuvchi jadvaldagi maydonning nomi,
// db_guest – mexmonlar kitobidagi ma'lumotlar saqlanuvchi jadval nomi,
$sql="select * from db_guest where (name='$user')";
$result=mysql_query($sql);
// SHartga mos keluvchi yozuvlar sonini aniqlaymiz
$rows=mysql_num_rows($result);
echo "$user mexmonlar kitobida $rows yozuvlarni qoldirdi.";

```

Agar izlanayotgan matn butun maydonning hammasini emas bir qismini egallasa nima qilish kerak (masalan, umuiy matn massivida soʻz yoki jumla izlanganda)?

Masalan quyidagi komandadan foydalanish mumkin:

```

// $search – izlangan matnni oʻz ichiga oladi
$sql="select * form db_guest where (locate('$search',content)>0)";
$result=mysql_query($sql);

```

Maʼnosi quyidagicha: agar \$search satrlarining content maydoniga 0 dan katta boʻlsa (Yaʼni ular umuman mavjud boʻlsa), yozuv topilgan yozuvlarga qoʻshiladi.

Shuni aytish kerakki koʻrsatilgan usul matnni registrni hisobga olgan xolda izlaydi. Registrni hisobga olmasdan izlash uchun quyidagi komandadan foydalanish:

```

// $search - izlangan matnni oʻz ichiga oladi
$sql="select * form db_guest where (locate(lower('$search'),lower(content))>0)";
$result=mysql_query($sql);

```

Yaʼni jadvaldagi izlanayotgan satr va yozuv yuqori registrga oʻtkaziladi, soʻngra izlash amalga oshiriladi.

Bizda mexmonlar kitobining maʼlumotlar bazasi mavjud boʻlsin va biz bu maʼlumotlarni qoldirilgan vaqti boʻyicha tartiblashimiz kerak boʻlsin.

Buning uchun baza maydonlaridan biri maʼlumot yozish vaqtini oʻz ichiga olishi kerak.

Biror ustun boʻyicha tartiblash **order by** konstruktsiyasi yordamida amalga oshiriladi.

Bizning misolimizda "Yangiroq " maʼlumotlar yuqoriga joylashadi:

```

// time – yozish vaqtini oʻz ichiga olgan ustun
// format "UNIX timestamp"
$sql="SELECT * FROM db_guest ORDER BY data DESC";
$result=mysql_query($sql);

```

Maʼlumotlar kamayish boʻyicha tartiblangan.

Agar maʼlumotlarni oʻsish boʻyicha tartiblashlozim boʻlsa, **DESC** kalit soʻzi oʻrniga **ASC** qoʻllanadi:

```

$sql="SELECT * FROM db_guest ORDER BY data ASC";
$result=mysql_query($sql);

```

Ma'lumotlarni bir necha ustunlar bo'yicha tartiblash mumkin, Buning uchun ularning nomlari vergul bilan ajratib ko'rsatilishi kerak.

Biror ustun bo'yicha tartiblanayotganda, shu ustunda bir necha bir xil qiymatlar mavjud bo'lsa kerak bo'ladi.

```
$sql="SELECT * FROM db_name ORDER BY field1,field2 ASC";
```

```
$result=mysql_query($sql);
```

Ma'lumotlar avval birinchi ustun *field1* bo'yicha tartiblanadi. So'ngra agar birinchi ustunda bir necha bir xil qiymatlar bo'lsa, ikkinchi ustun bo'yicha qo'shimcha tartiblash bajariladi( birinchi ustundagi bir qiymatli gurux ichida)yu

Agar izlashda hamma topilgan yozuvlar emas, ma'lum guruxni chiqarish lozim bo'lsa

**LIMIT** parametridan foydalaniladi.

Bu parametrdan ikki qiymat ko'rsatiladi:

**LIMIT start,length**

*start* - ukazyvaet, s kakoy pozitsii nujno vydavayt naydenные zapisi

*length* - kol-vo zapisey

Masalan, mehmonlar kitobi ma'lumotlar 20 dan 45 gacha yozuvlarni olish kerak bo'lsin (Ya'ni 25 ta yozuv chiqarish uchun):

```
$sql="select * from db_guest limit 20,25";
```

```
$result=mysql_query($sql);
```

## **Топширик Барилган вараиннинг маълумотлар базасини яратинг ва уни PHP,HTML билан боғлаб МБ даги маълумотларни тахрирланг**

### **Варианлар**

- 1) компьютер дўкони
- 2) Дастурий таъминотлар мажмуининг
- 3) Махалла фуқароларининг МБ 4) Интернет карточка билан савдо қилувчи дўкон МБ
- 5) мехнонхона МБ
- 6) талаба рейтингини аниқлашнинг МБ
- 7) кадрлар бўлими МБ
- 8) Маниторинг бўлими МБ
- 9) Интернет магазин МБ
- 10)Ўзбект Таомларни тайёрлашнинг МБ
- 11) Барча олий таълимлар хақидаги ахборот МБ
- 12) Барча касб-хунар таълимлар хақидаги ахборот МБ
- 13) Интернет Аптека МБ
- 14) Фермер МБ
- 15) мева дўкони МБ

## Назорат саволлари

- 1) PHP нима
- 2) Қандай web серверларни биласиз
- 3) MYSQL нима
- 4) HTML да форма компоненталаридан ыайсиларини биласиз
- 5) Form тегнинг қўшимча параметрлари
- 6) Select буйруғининг вазифаси
- 7) Insert буйруғининг вазифаси
- 8) PHP да ECHO буйруғининг вазифаси
- 9) PHP да шарт операторлари
- 10) PHP да танлаш операторлари

## Фойдаланилган Адабиётлар

1. Ш. Пауэрс Динамический HTML "Лори", 1999. 384 с.
2. Вайк А., Джиллиам Дж. JavaScript: Полное руководство: Перевод с английского. "Вильямс", 2004. 719 с.
3. Лещев Д. Создание интерактивного web-сайта: Учебный курс. "Питер", 2003.
4. Дронов В. JavaScript в Web-дизайне. СПб-Петербург, 2002. 880 с.
5. Джейсон Кренфорд Тиге DHTML и CSS для Internet. ИТ Пресс, 2005. 520 с.
6. Гаевский А.Ю., Романовский В.А. Самоучитель по созданию Web-страниц
7. Матросов, Сергеев, Чаунин. HTML 4.0 в подлиннике. ВHV-СПб, 2000, 672 с.
8. Уилтон П. JAVASCRIPT. Основы. Символ-плюс. 2002. 1056 с.
9. Кингли-Хью Э., Кингли-Хью К. JAVASCRIPT 1.5: Учебный курс. Питер. 1-е издание. 2002.
10. Бранденбау. JAVASCRIPT. Сборник рецептов для профессионалов. СПб. 2001.
11. Рейнбоу В. Компьютерная графика СПб-Питер, 2003. 768 с.
12. Викрам Васвани. Полнью справочник по MySQL. MySQL: The Complete Reference. Издательство: Вильямс, 2006 г., 528 стр.
13. Люк Веллинг, Лора Томсон MySQL. Учебное пособие MySQL Tutorial Перевод: Мягкая обложка Издательство: Вильямс, 2005 г., 304 стр.

14. Поль Дюбуа MySQL MySQL Серия: Landmark Другие издания:  
Твердыу переплет Аналоги: Твердыу переплет Издательство:  
Вильямс, 2004 г., 1056 стр.