



1

Mazkur o'quv-uslubiy majmua Oliy va o'rta maxsus ta'lif vazirligining 2020 yil 14-avgustdagи 418-sonli buyrug'i bilan tasdiqlangan fan dasturi asosida tayyorlandi.

Tuzuvchilar:	L.N.Xudoyorov – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi dotsenti Sh.T.Ulasheva – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi katta o'qituvchisi O.U.Begulov – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti M.A.Ro'ziqulova – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti A.Baxriddinov – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti K.Ablaqulov – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti
Taqrizchi(lar):	R.N.Abdullayev – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi katta o'qituvchisi F.Yu.Shodiyev – QDU “Algoritmlar va dasturlash texnologiyalari” kafedrasi katta o'qituvchisi

O'quv-uslubiy majmua Muahmmad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Qarshi filiali uslubiy kengashining 2021 yil _____ dagi ____-sonli qarori bilan tasdiqqa tavsiya qilingan.

MUNDARIJA

I.	NAZARIY MASHG'ULOTLAR MATERIALLARI.....	4
II.	LABORATORIYA MASHG'ULOTI MATERIALLARI.....	171
III.	MUSTAQIL TA'LIM MAVZULARI.....	234
IV.	GLOSSARIY.....	231
V.	ILOVALAR	237

I. NAZARIY MASHG'ULOTLAR MATERIALLARI

1-ma’ruza. WEB dasturlashga kirish. Asosiy tushunchalar. WWW rivojlanish bosqichlari.

Reja:

1. *Asosiy tushunchalar.*
2. *WWW rivojlanish bosqichlari.*

1. Asosiy tushunchalar. Bugungi kunda Internetning ommaviyligi haqida gapirish o’rinsiz. Internet hayotimizning bir bo’lagiga aylandi, biz uning xizmatlaridan har kuni foydalanishga odatlandik. Hozirda ixtiyoriy inson web-texnologiyalarning inson hayotining ta’lim, kommersiya, siyosat, ko’ngil ochar , ... bo’laklariga kirib borganligini tasavvur eta oladi va uning guvohi va foydalanuvchisiga aylanmoqda.

Internet turli xil insonlarni yagona maqsad bilan birlashishiga sabab bo’lmoqda. Hamma Internet tarmog’idan biror turdagи axborot olishga harakat qiladi. Shunday vaqtlar keladiki, hujjatni Internetda chop etish malakasi yozuv mashinasidan foydalanish kabi har bir, hatto o’rta ma’lumotga ega bo’lgan insonning qo’lidan keladi.

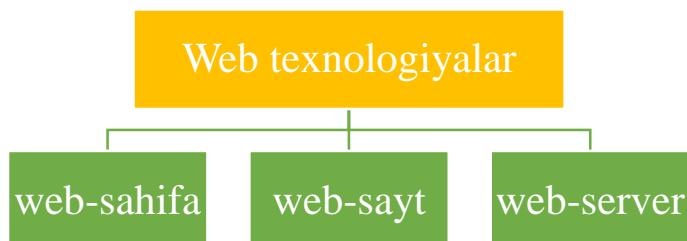
Dastlabki web-sahifalar juda sodda tuzilishga ega bo’lib, ular matnni formatlash va giperko’rsatkichlardan tarkib topgan edi. Web texnologiyalar rivojlanishi natijasida Web sahifalar tarkibida Plug-in dasturlar joylashtirila boshlandi, natijada Web sahifalarga interfaol xususiyati berildi. Web texnologiyalarning rivojlanishining oxirgi natijalaridan biri bu skript tillaridir (Script Languages). Ularni ishlatalishdan maqsad Web serverining ishini yengillashtirish, har-xil ishlar uchun Web serverini bezovta qilmasdan, bunday masalalarni foydalanuvchi kompyuterining o’zida yaratishdir. Web texnologiyasining oxirgi erishgan yutuqlaridan biri dinamik Web sahifalardir. Dinamik Web sahifalar CGI dasturlar bilan bevosita bog’liq bo’lib, CGI dasturlar serverda joylashgan va server imkoniyatlarini ishlaturuvchi dasturlardir. Ular serverga kelgan so’rovlarni qayta ishlaydi va qayta ishlash natijasida yangi Web sahifa hosil bo’ladi.

Asosiy tushunchalar

Web-sahifa – o’zining unikal adresiga ega bo’lgan va maxsus ko’rish dasturi yordamida (brauzer) ko’riluvchi hujjatdir. Unga matn, grafika, ovoz, video yoki animatsiya ma'lumotlar birlashmasi - multimediali hujjatlar, boshqa hujjatlarga gipermurojaatlar kirishi mumkin.

Web-sayt – bir qancha web-sahifalarning mantiqiy birlashmasi.

Web-server – tarmoqqa ulangan kompyuter yoki undagi dastur hisoblanib, umumiy resurslarni klientga taqdim etish yoki ularni boshqarish vazifalarini bajaradi. Internet tarmog’ini foydalanuvchilarga tarmoq resurslaridan erkin foydalanish imkoniyatini beradigan web- serverlarsiz tasavvur etib bo’lmaydi. Bunday serverlarda Internetda taqdim etilgan axborotning katta qismi jamlangan.



2. WWW rivojlanish bosqichlari. Internet tarmog'ining ishlash prinsipi TCP/IP (Transmission Control Protocol/Internet Protocol - ma'lumotlarni uzatish qaydnomasi/ Internet qaydnomasi) kompyuter tarmog'ida ma'lumotlarni uzatish qaydnomalari majmuining nomidir.

TCP (Transmission Control Protocol). Qabul qiluvchi va uzatuvchi kompyuterlarning mantiqiy bog'lanishga asoslangan ma'lumotlar uzatishini qo'llab - quvvatlovchi qaydnomasi.

IP (Internet Protocol)- Ma'lumotlar uzatishni ta'minlaydi

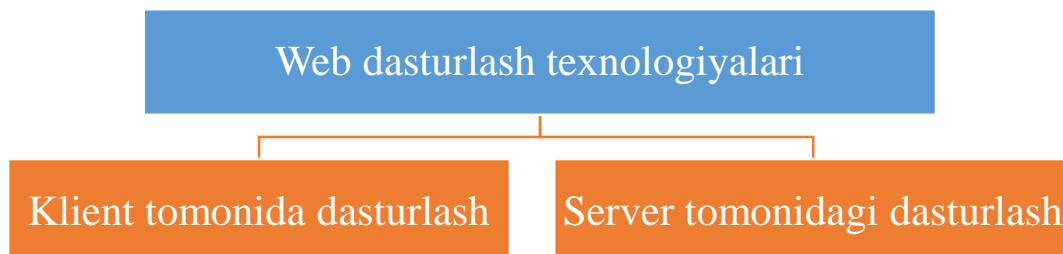
Biz Internet tarmog'idagi Web-sahifalarni ko'rishimiz uchun WWW (World Wide Web) deb ataluvchi servisdan foydalanamiz.

World Wide Web (WWW, Butun dunyo o'rgimchak to'ri) – bu klient-server texnologiyasi asosida tashkil etilgan, keng tarqagan Internet xizmatidir.

WWW (World Wide Web) – bu qanaqadir Internetdan ajratilgan ma'lum bir joy emas, kompyuter aloqa o'rnatadigan biror nima ham emas. Butunjahon o'rgimchak to'rini Internet doirasidagi xizmat deyish to'g'riroq. Web-serverlar deb ataluvchi ma'lum protokollardan, kompyuterlardan foydalanish orqali (chunki ular tarmoqqa ulangan va server dasturiy ta'minotiga ega) Internet xizmati yo'lga qo'yiladi.

Kompyuter web-server bo'lishi uchun Internetga ulangan va server dasturiy ta'minoti (DT) ga ega bo'lishi etarli. Bu DT bilan Windows, Mac OS, Unix kabi operatsion sistemalar ta'minlay oladi. Web-server har doim Internetda "o'tiradi" va talab qilingan tomonga kerakli informatsiyani jo'natadi.

"Klient-server" texnologiyasi



Klient tomonidagi ssenariylar foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni to'g'riligini serverga murojaat qilmasdan tekshiradi. Ko'p hollarda bu ssenariylar JavaScript va VBScript tillarida yoziladi.

Server tomonida bajarilishi kerak bo'lgan ssenariylar odatda sayt papkasining ichidagi maxsus papkaga joylashtiriladi. Foydalanuvchi so'roviga asosan server bu ssenariyni bajaradi. Bajarilgan ssenariy natijasi web-serverga uzatiladi va undan so'ng klientga uzatiladi. Server tomonidagi ssenariylarni tashkil etish uchun odatda Perl, ASP, PHP, JSP va SSI kabi til va texnologiyalardan foydalaniladi.

Nazorat savollari

1. Web sahifa nima?
2. Web sayt nima?
3. Web server nima?

2-ma'ruza. HTML tiliga kirish. HTML xujjatining umumiy strukturasi va uning asosiy teglari

Reja:

1. HTML ga kirish. HTML-hujjatlar tuzilishi.
2. HTML ning asosiy teglari, atributlari. Sarlavha teglari.
3. Rasm va jadvallar bilan ishslash teglari.
4. Havolalar va ro'yxatlar bilan ishslash teglari.
5. HTML formalar.

Kalit so'zlar: Matn muxarirlari, kod muxarirlari, WYSIWYG texnologiyasi, HTML, HEAD, BODY

Ishdan maqsad: HTML asosiy tushunchalarini o'rganish, uning sarlavha va tana qismiga tegishli bo'lgan asosiy teglar bilan tanishish, talabalarga gipermatnli hujjat tashkil etish asoslari, manbalari, HTMLhujjat haqida tushunchalar berish va HTML-hujjat yarata olish ko'nikmalarini shakllantirishni tashkil etish;

1. HTML kodlarni yozishda qo'llaniladigan matn muharrirlari haqida ma'lumot

Windows muxitidagi matn muxarirlari: Notepad, TextPad, UltraEdit, EdutPlus. Ko'rsatilgan xamma matn muxarirlari yordamida HTML kodlarni yozishda qo'llash mumkin.

Ko'pgina matn muxarirlarida HTML kodlarni xatto programmalashtirish tillarida kodlarni yozish juda kulay. Ba'zilari avtomatik ravishda operatorlar, funksiyalarni tanish va ularni xar-xil ranglarda tasvirlash imkoniyatiga ega. Ba'zi matn muxarirlarida HTML xujjatni web brauzerda sinab ko'rish tugmasi mavjud.

HTML xujjatni yaratishga mo'ljallangan maxsus programmalar (HTML muxarirlar) xam mavjud: FrontPage, Adobe GoLive, Macromedia Dreamweaver, Nestcape Composer. Muxarirlar 2 turga bo'linadi:

- ✓ kod muxarirlari;
- ✓ WYSIWYG texnologiyasi (What You See Is What You Get – nimani ko'rsang o'shani olasan) asosida ishlaydigan muxarirlari. Bu muxarirlar yordamida foydalanuvchi HTML komandasini va elementlarini yozmaydi, oddiy matn muxarirlaridek matn yozadi, tasvirlarni kerakli joyga joylashtiradi va x.k. xolos.

WYSIWYG-muharriri

Vizual HTML muharrirlarni 2 asos bo'yicha etirof qilinadi:

1. Adobe tarkibidagi GoLive;
2. Macromedia tarkibidagi Dreamweaver;

Adobe GoLive CS oynasi ko'rinishi quyidagicha:



Macromedia Dreamweaver MX 2004 oynasi ko'rinishi quyidagicha:



2. HTML tili haqida qisqacha ma'lumot

HTML (ing. Hypertext Markup Language — gipermatnli belgilash tili) - bu SGMLga (Standard Generalized Markup Language — standart umumlashtirilgan belgilash tili) asoslangan va xalqaro ISO 8879 standartiga mos keluvchi til, xalqaro to'rda ishlataladi.

HTML tili taxminan 1991-1992 yillarda Yevropa Yadroviy Tadqiqotlar Markazida ishlovchi britaniyalik mutaxassis Tim Bernars Li tomonidan ishlab chiqilgan. Dastlab bu til mutaxassislar uchun xujjat tayyorlash vositasi sifatida yaratilgan. **HTML** tilining soddaligi (SGMLga nisbatan) va yuqori formatlash imkoniyatlarining mavjudligi uni foydalanuvchilar orasida tez tarqalishiga sabab bo'ldi. Bundan tashqari unda gipermatnlardan foydalanish mumkin edi. Tilning rivojlanishi bilan unga qo'shimcha multimedia (rasm,tovush, animatsiya va boshqalar) imkoniyatlari qo'shildi.

Brauserlar - maxsus **HTML** tilida yaratilgan xujjatlarni o'quvchi kompyuter dasturi. Aynan brouserlar **HTML** tilida yaratilgan xujjatlarni formatlangan xolda ko'rish imkoniyatini beradi. Xozirda eng mashxur brouserlar bu Internet Explorer, Firefox, Opera va xokazolardir.

Imkoniyatlari

HTML quyidagi formatlash imkoniyatlariga ega:

- Matn qismining mantiqiy rolini belgilash (matn sarlavhasi, paragraf, ro'yxat va hokazo).
- Gipermatnlar yaratish. Bu ayniqsa juda qulay bo'lib o'zaro bog'langan hujjat sahifalari orasida navigatsiya qilishni yengillashtiradi.
- Matnning rangi, qalinligi va boshqa shrift ko'satkichlarini belgilash.
- Maxsus belgilar qo'yish. ASCII kodirovkasida ko'satilmagan belgilar **HTML** vositalari bilan qo'yish mumkin. Masalan grek alfaviti belgilari α, ψ, ζ , matematik belgilar $\int, \infty, \sqrt{ }, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}$, moliya belgilari $\text{€}, \text{£}, \text{¥}, \text{©}, \text{®}, \text{™}$ va hokazolar.
- Foydalanuvchi kiritishi uchun maydonlar yaratish.
- Multimedia fayllarini ochish. □ Boshqa imkoniyatlar.

3. HTML hujjat tuzilishi

HTML (Hyper Text Markup Language) – belgili til bo'lib, ya'ni bu tilda yozilgan kod o'z ichiga mahsus ramzlarni mujassamlashtiradi. Bunday ramzlar hujjat ko'rinishini faqatgina boshqarib, o'zi esa ko'rinxaydi. HTMLda bu ramzlarni teg (teg – yorlik, belgi) deb ataladi. HTMLda hamma teglar ramzchegaralovchilar (< , >) bilan belgilanadi. Ular orasiga teg identifikatori (nomi, masalan V) yoki uning atributlari yoziladi. Yagona istisno bu murakkab chegaralovchilar (<!--va -->) yordamida belgilanuvchi sharxlovchi teglardir. Aksariyat teglar jufti bilan ishlataladi. Ochuvchi tegning jufti yopuvchi teg. Ikkala juft teg faqatgina yopuvchi teg oldidan «slesh» (“/”) belgisi qo'yilishini hisobga olmaganda, deyarli bir xil yoziladi. Juft teqlarning asosiy farqi shundaki, yopuvchi teg parametrlardan foydalanmaydi. Juft teg yana konteyner deb ham ataladi. Juft teglar orasiga kiruvchi barcha elementlar teg konteyneri tarkibi deyiladi. Yopuvchi tegda zarur bulmagan bir qator teglar mavjud. Ba'zida yopuvchi teglar tushirib qoldirilsa ham zamonaviy brauzerlar aksariyat hollarda hujjatni to'g'ri formatlaydi, biroq buni amalda qo'llash tavsiya etilmaydi. Masalan, rasm qo'yish tegi , keyingi qatorga o'tish
, baza shriftini ko'rsatish <BASEFONT> va boshqalar o'zining , </BR> va hokazo yopuvchi juftlarisiz yozilishi mumkin. Noto'g'ri yozilgan tegni yoki uning parametri brauzer tomonidan rad qilinadi. (bu brauzer tanimaydigan teqlarga ham taalluqli). Masalan, <NOFRAME> teg-konteyneri faqatgina freymlarni taniydigan brauzer tomonidan hisobga olinadi. Uni tanimaydigan brauzer <NOFRAME> tegini tushunmaydi. Teglar parametr va atributlarga ega bo'lishi mumkin. Parametrlar yig'indisi har-bir tegda individualdir. Parametrlar quyidagi qoida asosida yoziladi:- Teg nomidan so'ng probellar bilan ajratilgan parametrlar kelishi mumkin;- Parametrlar ixtiyoriy tartibda keladi;- Parametrlar o'zining nomidan keyin keluvchi «» - «qo'shtirnoq» ichida beriladi. - Parametr qiymatida ba'zan yozuv registri muxim. Shuni esda tutish lozimki, hamma teglar o'zining individual parametriga ega bo'lishiga qaramay, shunday bir qator parametrlar mavjudki, ularni <BODY> bo'limining barcha teglarida ishlatalish mumkin. Bu parametrlar CLASS, ID, LANG, LANGUAGE, STYLE va TITLElardir. HTML-hujjatini yozishni boshlashda ishlataladigan birinchi teg bu <HTML> tegidir. U har doim hujjat yozuvining boshida bo'lishi lozim. Yakunlovchi teg esa </HTML> shakliga ega bo'lishi kerak. Bu teglar, ular orasida joylashgan yozuvning hammasi

butun bir HTMLhujjatini anglatishi bildiradi. Aslida esa hujjat oddiy matnli ASCII-faylidir. Bu teglarsiz brauzer hujjati formatini aniqlab, tarjima qila olmaydi. Ko'pincha bu teg parametrga ega emas. HTML 4.0 versiyasiga qadar VERSION parametri mavjud edi. HTML 4.0da esa VERSION o'rniغا <!DOCTYPE> parametri paydo bo'ldi. <HTML> va </HTML> orasida 2 bo'limdan tashkil topishi mumkin bo'lgan hujjatning o'zi joylashadi. Mazkur hujjatning birinchi bo'limi sarlavhalar bo'limi (<HEARD> va </HEARD>), ikkinchi bo'lim esa hujjat tana qismidir (<BODY> va </BODY>), uni hujjat tanasi ham deb yuritamiz. Freym tuzilishi hujjatlar uchun <BODY> bo'limining o'rniغا <FRAMESET> bo'limidan foydalilanildi.

4. Hujjatning HEAD bo'limi

```
<HTML>
  <HEAD>
    *
    сарлавҳа қисми
    *
  </HEAD>

  <BODY>
    *
    тана қисми
    *
  </BODY>

</HTML>
```

HEAD bo'limi sarlavha hisoblanadi va u majburiy teg emas, biroq mukammal tuzilgan sarlavha juda ham foydali bo'lishi mumkin. Sarlavha qismining maqsadi hujjatni tarjima qilayotgan

dastur uchun mos axborotni etkazib berishdan iborat.

Hujjat nomini

ko'rsatuvchi <TITLE>

tegidan tashhari bu bo'limning qolgan barcha teglari ekranda aks ettirilmaydi. Odatda <HEAD> tegi darhol <HTML> tegidan keyin keladi. <TITLE> tegi sarlavhaning tegidir, va hujjatga nom berish uchun hizmat qiladi. hujjat nomi <TITLE> va </TITLE> teglar orasidagi matn qatoridan iborat. Bu nom barauzer oynasining sarlavhasida paydo bo'ladi (bunda sarlavha nomi 60 belgidan ko'p bo'lmasligi lozim). O'zgartirilmagan qolda bu matn hujjatga «zakladka» (bookmark) berilganda ishlatiladi. hujjat nomi uning tarkibini qisqacha ta'riflashi lozim. Bunda umumiylar ma'noga ega bo'lgan nomlar (masalan, Homepage, Index va boshqalar)ni ishlatmaslik lozim. Hujjat ochilayotganda birinchi bo'lib uning nomi aks ettirilishi, so'ngra esa hujjat asosiy tarkibi ko'p vaqt olib, kengayib ketishi mumkin bo'lgan formatlash bilan birga yuklanishini hisobga olgan qolda, foydalanuvchi xech bo'lmaganda ushbu axborot qatorini o'qiy olishi uchun hujjatning nomi berilishi lozim.

5. Hujjatning BODY bo'limi

Ushbu bo'linma hujjatning tarkibiy qismini o'z ichiga oladi. Bo'linma <BODY> tegidan boshlanib </BODY> tegida tugaydi. Biroq ushbu teglar qat'iy mavjud bo'lishi shart emas, chunki brauzerlar matnga qarab hujjat tarkibiy qismining ibtidosini aniqlashi mumkin. <BODY> tegining bir qator parametrleri mavjud bo'lib, ularning birortasi ham majburiy emas.

<BODY> tegi parametrлари:

ALINK – faol murojaat (ssilka)ning rangini belgilaydi.

BACKGROUND – fondagi tasvir sifatida foydalaniluvchi tasvirning

URL-manzilini belgilaydi.

BOTTOMMARGIN – hujjatning quyi chegaralarini piksellarda belgilaydi.

BGCOLOR – hujjat fonining ranglarini belgilaydi.

BGPROPERTIES – agar FIXED qiymati o’rnatilmagan bo’lsa, fon tasviri aylantirilmaydi.

LEFTMARGIN – chap chegaralarni piksellarda belgilaydi.

LINK – xali ko’rib chiqilmagan ssilkaning ranggini belgilaydi.

RIGHTMARGIN – hujjat o’ng chegarasini piksellarda o’rnatadi.

SCROLL – brauzer darchalari xarakatlantirish (prokrutka) yo’laklarini o’rnatadi. TEXT – matn rangini aniqlaydi.

TOPMARGIN – yuqori chegarasini piksellarda o’rnatadi. VLINK – ishlatilgan murojaat rangini belgilaydi.

BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN va TOPMARGIN parametrleri matn chegarasi va darcha chetlari orasidagi masofani piksellarda belgilaydi. (Faqat HTML 4.0 versiyasidan boshlab IE brauzerlari bu parametrлarni taniy oladi) BGPROPERTIES parametri faqatgina bitta FIXED qiymatiga ega. HTML dagi ranglar o’n otilik sanoq tizimida (RGB), yoki ranglar nomi yordamida berilishi mumkin. Ranglar bazasi 3 ta rangga – qizil (R), yashil (G) va ko’k (B) ranglarga asoslangan bo’lib, u RGB deb belgilanadi. har-bir rang uchun 00 dan FF gacha bo’lgan o’n otilik sanoq tizimidagi qiymat beriladi, bu esa 0 dan 255 gacha bo’lgan diapazonga to’g’ri keladi. So’ngra bu qiymatlar bir songa birlashtiriladi va ularning oldiga “#” belgisi qo’yiladi. Masalan, #800080 siyohrangni bildiradi.

Misollar:

<BODY TEXT = “#000000”> yoki <BODY TEXT q=black>

<BODY BGCOLOR =“#ffffff”> yoki <BODY BGCOLOR = WHITE>

<BODY LINK q “#ff0000”> yoki <BODY LINK = RED>

<BODY LINK = “#ooFFFF” ALINK = “#800080”> yoki <BODY VLINK = Aqua ALINK =

PURPLE>

hamma barauzerlar o’n otilik sanoq tizimidagi standart ranlarni taniydi. Bular quyidagilardir:

Black	#000000	Maroon	#800000
Silver	#C0C0C0	Red	#FF0000
Grey	#808080	Purple	#800080
White	#FFFFFF	Green	#008000
Fuchsia	#FF00FF	Navy	#000080
Lime	#00FF00	Blue	#0000FF
Olive	#808000	Teal	#008080
Yellow	#FFFF00	Aqua	#00FFFF

Quyida ko’rastilgan parametrler asosida misol ko’rib o’tamiz:

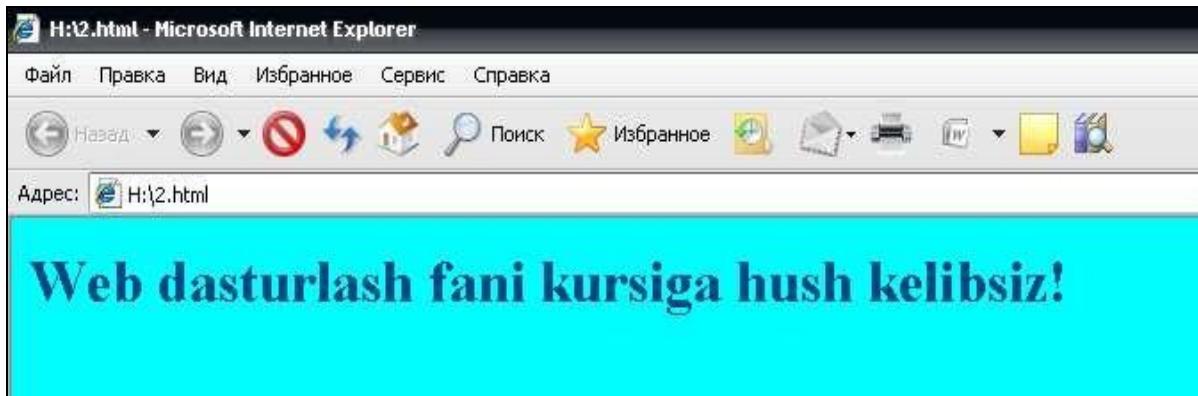
```
<html>
```

```
<BODY BGCOLOR = AQUA TEXT = “#848484” LINK = RED VLINK = PURPLE ALINK = GREEN>
```

```
<h1> Web dasturlash fani kursiga hush kelibsiz!</h1>
```

```
</html>
```

Bu dasturni ishga tushirish asosida quyidagi oynaga ega bo'lamiz:



Agar BGCOLOR parametri rangni nomi yoki uning tarkibiy qismlarini o'n otilik sanoq tizimidagi kodda keltirish vazifasi yordamida fon ranggini chiqarish uchun ishlatsila, BACKGROUND tasvir yordamida sahifaga fon berishda foydalaniladi. Tasvir sifatida GIF yoki JPG formatidagi grafik fayllar ishlataladi. HTML-hujjat fonidagi tasvir doimo butun sahifani to'ldirib turadi. Agar tasvir o'lchami darcha o'lchamidan kichik bo'lsa, u mozayka tamoyiliga asosan ko'paytiriladi. Odatda fon tasviri sifatida tarmoq orqali yuklash uchun uncha ko'p vaqt ketmaydigan kichik tasvir tanlab olinadi, yoki fon sifatida shaffof relief logotipi tasviridan foydalaniladi.

Misol:

```
<BODY BACKGROUND = texture.gif BGCOLOR = gray>.
```

Sahifa yaratilishida doimo fon rangini berish tavsiya qilinadi. Agar fon tasviri ham berilayotgan bo'lsa, fon va tasvir ranglari bir-biriga yaqin bo'lgani ma'qul. Misol:

```
<BODY TEXT = BLUE LINK = RED VLINK = BLUE ALINK = PINK  
BACKGROUND="" HYPERLINK
```

```
"http://www.foo.com/jkorppela/HTML3.2/wave.gif" >.
```

Misol:

```
<HTML><HEAD><TITLE> - Sahifa fonini berish misoli </TITLE></HEAD>  
<BODY BGCOLOR = YELLOW TEXT = BLACK LINK = RED VLINK = PURPLE  
ALINK = GREEN>  
</BODY></HTML>
```

6. HTML shablonni yaratish

Notepad ni ochamiz. HTML faylni yaratishni boshlaymiz. Sarlavxa yozish uchun:

```
<head>  
</head> teglari  
yoziladi.
```

Brauzer bu teglar o'rtasidagi matnni sarlavxa deb tushunadi va Brauzerning eng tepe qismiga shu matnni yozadi. Endi saxifaning tanasini xosil qilamiz:

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

Bu teglar o'rtasiga xamma matn va tasvirlarni joylashtirish kerak. HTML shablon bo'lishi uchun HTML saxifaning sarlavxa va tanasini o'z ichiga olgan quyidagi zarur teglar etishmayapti:

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

Demak HTML shablon quyidagi ko'rinishga ega bo'ldi:

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

va teglari o'rtasiga kuyidagi teglarni joylashtirish mumkin:

<title> ... </title> - xujjat nomi. Masalan:

```
<title> web saxifa </title>
```

<meta /> – ma'lumotlar xaqidagi ma'lumot. Ya'ni bu erda asosiy terminlar yoziladi. Qidiruv sistemalari ishlaganda aynan shu terminlar bo'yicha qidirish ishini olib boradi, saytlarni topadi. Masalan:

```
<meta name="keywords" content="TATU, Informatika, Institut, Fakultet, Student"/>  
content 50-200 ta so'zni o'z ichiga olishi mumkin. name="("keywords",  
"autor", "copyright", "description") – maxfiy qidirishda qo'llaniladi, HTTP serverga dostup berish  
uchun http-equiv qo'llaniladi.
```

Nazorat savollari:

1. Windows muxitidagi matn muxarirlari haqida tushuncha bering?
2. Matn muxarirlari turlarini aytинг?
3. HTML tegi vazifasini tushuntirib bering?
4. Teglar haqida ma'lumot bering?
5. HTML tilida teglar qanday belgilanadi?
6. HEAD tegi va uning ichidagi teglar xaqida gapiring?
7. <TITLE> va </TITLE> teglar orasidagi sarlavha nomini belgilash qoidalarini ko'rsating?
8. Parametrlar qiymati qanday ko'rinishda beriladi?
9. BODY tegi uning parametrlari haqida ma'lumot bering?
10. HTML shablonlarni qanday saqlanadi?

3-ma'ruza. HTML5 yangi standartlari va ularning vazifalari

Reja:

1. HTML5 yangi standartlar.
2. HTML5 da yangi teglar va atributlar.
3. Audio, video va grafika bilan ishlash.
4. Vizual effektlar.

Kalit so'zlar: Formalar (<FORM>), freymalar (<FRAME>), ob'ektlar, <INPUT>, ularning parametrlari, <SELECT>,<TEXTAREA>

Ishdan maqsad: Web-dasturlash fani bo'yicha HTML tili asosidagi formalar va freymalar bilan ishlash jarayonida talabalarda bilim, ishlash ko'nikmasi va malakalarini shakllantirish va bu jarayonni amalga oshirishda talabalarga ma'lumot berishni tashkil qilish.

1. HTML tilida formalar

Formalar Web dasturlashda foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni tartibga solish uchun qo'llaniladi. Forma elementlari to'ldirilgandan keyin undagi ma'lumotlar serverdagi ma'lumotlarni qayta ishlovchi dasturga yuboriladi. Ko'p sonli jo'natilayotgan ma'lumotlar jo'natish tugmasi bosilgandan so'ng serverda joylashgan Common Gateway Interface (CGI) yordamida yoki maxsus fayl orqali qayta ishlanadi. Shu tariqa foydalanuvchi Internet orqali Web-server bilan birgalikda ishlaydi.

HTML tili web-saxifalar tarkibiga matnli soxalar, menyular, tugmalar kabi interfeys elementlar qo'yish imkoniyatini beradi. Bu ma'lumotlar juda oddiy (elektron adres) yoki yetarlicha murakkab bo'lishi mumkin.

Masalan, foydalanuvchi onlayn xolatida internet dasturlari bilan ishlashi mumkin (kasalxona sayti orqali vrach chaqirishi yoki ishxonasining sayti yordamida avtomobilni prokatga olishi mumkin).

Elektron formalarni yaratishni maqsadi quyidagicha: u foydalanuvchidan informatsiya so'rashi va undan informatsiya olishi mumkin. Bunday holatlarda ma'lumotlarning bir qismini foydalanuvchi o'ziga tegishli bo'lgan ma'lumotni tanlashi mumkin bo'lgan tayyor menyular tashkil etishi mumkin. So'rovlar quyidagi turlarda bo'lishi mumkin:

Javoblari maxsus matnli soxalarga terilishi kerak bo'lgan savollar (masalan, ismi, familiyasi, elektron adresi, ...).

Javoblari maxsus menyulardagi yoki ro'yxatdagi javoblardan tanlashi kerak bo'lgan savollar (masalan qaysi davlatga tegishli ekanligi, qaysi mavzular qiziqtirishi, ...).

Pereklyuchatellar orqali javoblardan yagonasini tanlashga mo'ljallangan savollar (masalan, erkak/ayol, xa/yo'q...).

Formalar HTML tilining kiritish oynasi, tugma, pereklyuchatel kabi elementlar naborlaridan foydalangan xolda yaratiladi. Foydalanuvchi tomonidan yaratilayotgan xar bir element o'zining nomiga ega bo'lib, keyinchalik oldindan aniqlangan biror o'zgaruvchiga beriladi. Masalan, matn kiritish maydoniga

shahar nomini berilsa va foydalanuvchi maydonni Jizzax deb to'ldirsa shuni bildiradiki, shahar o'zgaruvchisiga Jizzax qiymati berildi.

O'zgaruvchilar va ularning qiymatlari serverlarga uzatiladi va o'z navbatida server «skriptlar» deb ataluvchi kichik programmalarga murojaat qiladi. Skriptlar olingan ma'lumotlarni qabul qilib, va ularni qayta ishlaydi.

Natijada web-saxifada foydalanuvchining ma'lumoti qabul qilinganligi to'g'risida oddiygina ma'lumot(masalan, "raxmat") chiqarib qo'yish mumkin.

HTML formalari bilan ishlash uchun skriptlar bilan ishlay olish kerak (ular CGI skriptlar deyiladi va ularni turli programmalashtirish tillarida tuzish mumkin. CGI—Common Gateway Interface).

2. Formaning berilishi —FORM elementlari

FORM elementi hujjatni ma'lum bir formaga soladi va forma elementlari teglarini boshqa teglardan ajratib turadi. <FORM> bir nechta <INPUT> yoki shu kabi boshqa teglar ketma ketligidan tashkil topadi. Ular <FORM> va </FORM> teglari orasiga joylashtiriladi. Formada usuldan (method), formaga kiritilgan ma'lumotlarni qayta ishlash uchun holatlar (action) mavjud. Usul (GET yoki POST) formaga kiritilgan ma'lumotlar qay tarzda serverga jo'natilish usulini belgilasa, holat esa serverdagи qaysi dasturga yuborilish URL (Uniform Resource Location) manzilini ifodalaydi. <FORM METHOD="post" ACTION="<URL>">

Quyida FORM elementi parametrlari bilan tanishib chiqamiz:

Formaning boshqaruv elementlari —<INPUT> tegi

Boshqarish elementining yana bir ko'rinishi bu – menyulardir. <Select> elementi yordamida xar xil menyularni xosil qilish mumkin. Bu xолatlarda foydalanuvchiga bir nechta variantlar berilib, undan o'zining javobini tanlash imkoniyati beriladi. Name atributi zaruriy atribut xisoblanadi. Size atributi yordamida esa ekranga bir vaqtida chiqadigan variantlar sonini berish mumkin. Bik .Yu <select> bu konteynordir. Qiymatlar <option> yordamida kiritilib, <select> va </select> lar orasiga joylashtiriladi. Agar foydalanuvchi menyuning biror-bir punktini tanlasa qiymat <select> tegida nomlangan o'zgaruvchiga beriladi. Bik .Yu Selected= "selected" atributi biror qiymatni avtomatik tanlash imkoniyatini beradi (znachenie po umolchaniyu):

```
<select name="javob1">
  <option value="first"> Birinchi </option>
  <option value="monthly"> Xar oyda <></option>
  <option value="weekly"> Xar xuftada </option>
  <option selected="selected" value="daily"> Xar kunda </option>
```

Yuqorida aytilganidek, size atributi yordamida ekranga chiqariladigan javoblar sonini kiritish mumkin: <select name="javob1" size="6">

Birinchi misolda javoblar bir qatorda ifodalanadi va sichqon tugmasi bosilganda qolgan javoblar ko'rindi. Ikkinci misolda size da necha son ko'rsatilgan bo'lsa shuncha javob varianti ko'rsatiladi va qolgan javoblarni prokrutkalarni siljitsish yordamida ko'rish mumkin.

Agar javoblardan bir nechtasini bir vaqtida tanlash zarur bo'ladigan bo'lsa <select> ning multiple= "multiple" atributi yordamga keladi. Bu xолatda size da nechta javoblar ko'rsatilgan bo'lsa shulardan bir nechtasini tanlash imkoniyati bo'ladi.

Ushbu teg formaning qaysi nuqtasiga ma'lumot kiritilishini belgilaydi. U foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni formaga keltiradi. Bular matn kiritish maydoni,

ro'yxatlar, rasmlar yoki tugmalar bo'lishi mumkin. Maydon tipi TYPE atributi yordamida aniqlanadi.

TYPE=text atributi

Agar foydalanuvchi uncha katta bo'limgan matn kirtsqa (bir yoki bir nechta satr), <INPUT> tegidan foydalanadi va TYPE atributiga text qiymati o'zlashtiriladi. Standart holat uchun bu qiymatni berish muhim emas. Bundan tashqari maydonni nomlash va unga murojaat qilish uchun NAME atributi ham beriladi.

Sizning ismingiz <INPUT NAME=Name SIZE=35>

Foydalanish mumkin bo'lgan yana uchta qo'shimcha atributlar mavjud. Birinchisi MAXLENGTH deb ataladi, u foydalanuvchi kiritayotgan matn maydoni maksimum uzunligini belgilaydi. Standart bo'yicha bu qiymat chegaralanmagan. Ikkinci atribut SIZE hisoblanadi, u esa matn maydonini ko'rinish turuvchi qismini belgilaydi. Standart bo'yicha uning qiymati brauzerga bog'liq bo'ladi. Agar MAXLENGTH qiymati SIZE qiymatidan katta bo'lsa, brauzer ma'lumotni oynaga moslashtiradi. So'nga qo'shimcha atribut matn maydonini boshlang'ich qiymatini belgilovchi VALUE dir.

TYPE=checkbox atributi

HTML formada mustaqil belgilagich (bayroqcha) dan foydalanish uchun <INPUT> tegining atributiga TYPE=checkbox ni o'zlashtirish kerak. Formaga bog'liq ravishda foydalanuvchi bir yoki bir nechta belgilagichlarni belgilashi mumkin. Agar <INPUT> tegi atributi bilan CHECKBOX qiymati qo'llanilsa, u bilan birga NAME va VALUE atributlari ham qo'llanilishi kerak. NAME atributi ushbu ma'lumot kiritish ob'ektining nomini ifodalaydi. VALUE atributida ushbu maydonning qiymati ko'rsatiladi.

Rossiya<INPUT NAME="Davlat" TYPE=checkbox
VALUE="Rossiya">

Strani SNG<INPUT NAME="Davlat" TYPE=checkbox
VALUE="SNG">

Ba'zi hollarda ushbu maydon belgilangan qolda qo'llanilishi ham mumkin. Bunday hollarda <INPUT> tegida CHECKED atributi qo'llanilishi kerak.

TYPE=radio atributi

Ba'zan bir nechta qiymatlar orasidan birini tanlashga to'g'ri keladi. Bunday hollarda formada <INPUT> tegi bilan birga TYPE=radio atributi qo'llaniladi. Agar <INPUT> tegi atributi bilan ushbu qiymati qo'llanilsa, u bilan birga NAME va VALUE atributlari ham qo'llanilishi kerak. NAME atributi ushbu ma'lumot kiritish ob'ektining (tugma) nomini ifodalaydi. VALUE atributida ushbu maydon ning qiymati ko'rsatiladi..

Erkak jinsi <INPUT NAME="Jins" TYPE=radio
VALUE="Erkak">

Ayol jinsi <INPUT NAME="Jins" TYPE=radio
VALUE="Ayol">

TYPE=image atributi

Formaning tarkibiga harab ba'zan unda joylashgan rasmning ustiga sichqonchani bosish bilan undagi ma'lumotlarni jo'natishga to'g'ri kelib qoladi. Buning uchun <INPUT> tegi TYPE=image atributi bilan qo'llaniladi. Foydalanuvchi rasm ustiga sichqoncha cursorini bossa, aynan shu erdag'i ekran koordinatalarini brauzer saqlab qoladi. So'ng formaga kiritilgan

ma'lumotlarni "qayta ishlaydi". Agar <INPUT> tegi image atributi bilan qo'llanilsa, u bilan birga NAME va SRC atributlari ham qo'llanilishi kerak. NAME maydonning nomini belgilaydi. SRC atributi esa rasm joylashgan manbaning URI manzilini beradi. ALIGN atributi qo'shimcha hisoblanadi va u ham ba'zan tegi bilan qo'llaniladi.

```
<BR>Nuqtani tanlang <INPUT TYPE="image" NAME="point"
SRC="image.gif">
```

TYPE=password atributi

Agar formada parollardan foydalanish kerak bo'lib qolsa, TYPE atributi qiymatiga password (TYP=password) ni o'zlashtiriladi. Ushbu tipdan foydalanish kiritilayotgan ma'lumotni oshkor bo'limgan holda ko'rsatishni ta'min etadi. Shu sabab, kiritilgan ma'lumot ochiq kanal orqali jo'natiladi va ushbu ma'lumot tutib olinishi mumkin.

```
<BR>Nominingiz<INPUT NAME="login">Parol
<INPUT TYPE="password" NAME="So'z">
```

TYPE=reset atributi

Bazan foydalanuvchi formani to'ldirish vaqtida, ularni boshdan to'ldirishga to'g'ri keladi. Ushbu qolda Reset tugmasi mavjud bo'lib, bu tugmaning bosilishi formani dastlabki, kirish holatigi olib keladi (formani "tozalaydi"). Reset tugmasini tashqil qilish uchun <INPUT> tegi atributiga TYPE=reset o'zlashtiriladi. Agar formada reset atributi qo'llanilsa, <INPUT> tegiga VALUE atributini qo'shimcha qilish mumkin. Ushbu atribut tugmadagi yozuvni ifodelaydi.

```
<INPUT TYPE="reset" VALUE="Formani tozalash ">
```

TYPE=submit atributi

HTML forma da foydalanuvchi ma'lumot kiritish jarayonini yakunlash jarayoni mavjud. Buning uchun

<INPUT> tegining atributiga TYPE=submit qiymat o'zlashtiriladi. Agar formada <INPUT> tegi submit atributi bilan qo'llanilsa, unga qo'shimcha ravishda ikkita atributdan foydalanish mumkin: NAME va VALUE. NAME atributi maydonning nomini ifodelaydi. VALUE atributi — Submit tugmasi matnini ko'rsatadi.

```
<INPUT TYPE="submit" VALUE="Xabarni jo'natish ">
```

TYPE=hidden atributi

Yashirin maydon. INPUT tegini TYPE=hidden atributi bilan qo'llanilishi foydalanuvchiga ma'lum

bo'limgan NAME va VALUE atributlaridagi qiymatlarni jo'natishga imkon beradi.

```
<TEXTAREA> – ko'p satrli matn kiritishni tashqil etish tegi
```

Ba'zan formada ko'p satrli matnlarni kiritish talab etiladi. Buning uchun <TEXTAREA> tegi yordamida bir necha satrdan iborat bo'lgan matn maydoni tashqil etish mumkin. Ushbu teg uchta atribut bilan ishlataladi: COLS, NAME va ROWS.

- Atribut COLS

Maydonning ustunlari (belgilar soni) sonini belgilaydi.

- Atribut NAME

Maydonning nomini belgilaydi.

- Atribut ROWS

Maydonning ko'rinvchi satrlari sonini belgilaydi.

<TEXTAREA NAME=mavzu COLS=38 ROWS=3></TEXTAREA>

<SELECT>- formada ro'yxatdan foydalanish tegi

Agarda forma mukammal bo'lsa, gohida unda harakatlanuvchi ro'yxat ham qo'llaniladi. Buning uchun SELECT tegidan foydalaniladi. Ro'yxat bo'limlarini aniqlash uchun <OPTION>tegidan foydalaniladi. <SELECT> tegi muhim bo'lmasan uchta atributni qo'llab quvvatlaydi: MULTIPLE, NAME va SIZE.

- MULTIPLE atributi Bir vaqtning o'zida bir nechta variantni tanlash imkonini beradi.

- NAME atributi

Ob'ekt nomini ifodalaydi.

- SIZE atributi

Ro'yxatni ko'rinvchi satrlari sonini ifodalaydi. SIZE > 1 bo'lgan qolda brauzer oddiy ro'yxatni ko'rsatadi. Formada <OPTION> tegi faqat <SELECT> teglari orasida qo'llaniladi. Parametrleri: SELECTED va VALUE.

- SELECTED atributi

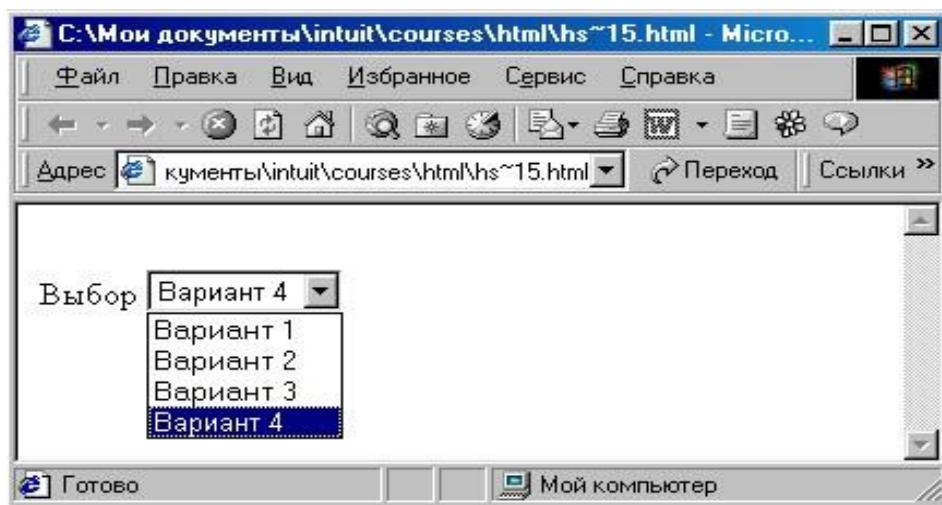
Dastlaki holatda ushbu element tanlangan ekanligini bildiradi.

- VALUE atributi

Ro'yxatga o'zlashtirilishi mumkin bo'lgan qiymatni ifodalaydi.

Tanlash

```
<SELECT NAME="Tanlash">
<OPTION>Variant 1
<OPTION>Variant 2
<OPTION VALUE="Variant 3">Variant 3
<OPTION SELECTED>Variant 4
</SELECT>
/
```



Qalqib chiquvchi menyu

Sayt strukturasini yaratishda asosan biz <div> tegidan yoki jadvallardan foydalanamiz. Saytning qismini <div> tegi orqali ajratamiz. Html5 da esa sayt strukturasi uchun alohida teglar yaratilgan. Bu teglar quyidagilar:

<HEADER>- sahifa nomi uchun teg.

```
<header>
<h1>This is the Title of the Content</h1>
<div class=<post-meta">
<p>By Author</a> <span class=<category">Filed in Web 2.0</span></p>
</div>
</header>
```

<FOOTER> - saytning eng pastki qismi uchun zarur bo'lgan teg.

Asosan bu tegning ichiga "copyright" kabi ma'lumotlar yoziladi.

```
<footer>
<div class=<tags">
<span class=<tags-title">Tags:</span> <a href="#" rel=<tag">Command
Prompt</a>, <a href="#" rel=<tag">Compass</a>, <a href="#" rel=<tag">CSS</a>, <a href="#" rel=<tag">Sass</a>, <a href="#" rel=<tag">Terminal</a>
</div>
```

```
<div class=<facebook-like">
<div>10 likes</div> <!-- let's pretend it to be the facebook like -->
</div>
```

</footer>

<NAV> - sayt navigatsiyasi uchun teg.

```
<nav>
<ul>
<li><a href="#overview">Overview</a></li>
<li><a href="#history">History</a></li>
<li><a href="#development">Development</a></li>
<ul>
</nav>
```

<ASIDE> - asosan o'ng yoki chap menyular yaratish uchun ishlatish mumkin bo'lgan teg.

<SECTION> - sahifadagi asosiy ma'lumotlarni chiqarish uchun.

<ARTICLE> - yangiliklar, kommentariylar, maqolalarning kontentlarini chiqarish uchun.

<article>

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus id felis et augue sagittis
euismod quis at sem. Nunc sit amet magna ac velit congue ultricies. Sed eros justo, lacinia in
fringilla sollicitudin, congue id massa. Nunc dignissim bibendum nibh, sed dictum massa pharetra
sit amet.</p>
```

</article>

<VIDEO> - saytga video joylashtirish uchun.

```
<video width=<600" height=<400" controls=<controls" poster=<video/poster.jpg">
<source src=<video/video.mp4" type='video/mp4; codecs=<avc1.42E01E, mp4a.40.2'">
```

Teg video ne podderjivaetsya vashim brauzerom.

```
<a href="video/video.mp4">Skachayte video</a>
```

</video>

<AUDIO> - saytga audio joylashtirish uchun.

```

<audio src=<>video/1.mp3" controls=<>controls"></audio>
<CANVAS> - bazadan qiymat olib diagramma shaklida chiqarish uchun.
<canvas id=<>smile" width=<>200" height=<>200">
<p>Vash brauzer ne podderjivaet risovanie.</p>
</canvas>
Bu teglarni kerakli joylarda to'g'ri ishlatalish sayt strukturasini tushunarli (tartibli)
bo'lismiga olib keladi.

```



```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>video</title>
</head>
<body>
    <video width="600" height="400" controls="controls" poster="video/poster.jpg">
        <source src="video/video.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
        <a href="video/video.mp4">Скачайте видео</a>
    </video>
    <br>
    Audio<br>
    <audio src="video/1.mp3" controls="controls"></audio>
</body>
</html>

```



Nazorat savollari:

1. HTML tilida formani vazifasini izohlab bering?
2. Freymlar qanday ishlaydi?
3. Hujjatda linkni freymga bog'lashni tushuntiring?
4. Ob'ekt nima va qanaqa ob'ektlarni bilasiz?
5. Formaning qaysi nuqtasiga ma'lumot kiritilishini belgilovchi tegni misollar yordamida tushuntiring?
6. Kiritilayotgan matn maydoni parametrlari qanday belgilanadi?
7. Formada harakatlanuvchi ro'yhat qo'llash uchun qanday parametrlar kerak?
8. Freymlar va formalarni qo'llash bilan qanday imkoniyatlarga ega bo'lismiz mumkin?
9. Frame tegi atributlarini m9sollar yordamida tushuntiring?
10. <iframe> elementini qo'llash bilan qanday imkoniyatlarga ega bo'lismiz mumkin?

4- ma’ruza. CSS ga kirish. HTMLga ulanish usullari. Selektorlar turlari

Reja:

1. CSS ga kirish. CSS sintaksisi.
2. CSS bilan HTML hujjatni bog‘lash.
3. Selektorlar turlari (oddiy, kombinasiyalashgan, psevdo-element, psevdo-sinf, atribut selektorlar).
4. O‘lchov birliklari.

Kalit so’zlar: CSS, selector, teg, ID, Klass, hususiyat.

CSS sintaksisi quyidagi ko’rinishga ega:

Selektor

```
{  
hususiyat1: qiymat1;  
hususiyat2: qiymat2;  
hususiyat3: qiymat3 qiymat4;  
}
```

Selektor uch xilda bo’lishi mumkin:

1. Teglar

A {font-size: 12pt; text-decoration: none}

Barcha xavolalar shrift razmeri 12 va tagidagi chiziq olib tashlanadi.

TABLE {border: black solid 1px} Barcha jadvallar chegara chiziq qora uzlucksiz va 1 piksel yo’g’onlikda bo’ladi.

H1, H2, H3, H4, H5, H6 {color: red}

Barcha sarlavhalar qizil rangda belgilanadi.

TABLE A {font-size: 120%}

Jadval ichidagi havola razmeri 120% bo’ladi.

2. Klass

CSS

.class1 {text-decoration: underline; font-size: 80%}

Class1 ga mansub elementlar tagi chiziladi va shrift razmeri 80% belgilanadi.

A.class1 {text-decoration: none;}

Class1 dagi havola tag chiziq olib tashlanadi.

HTML

<h1 class="class1">Men yoqtirgan saytlar</h1>

**

*Kun.uz
*

* Daryo
*

HTML faylda kerakli elementlar class1 ga tegishliligi belgilanadi.

3. ID (identifikator)

Kerakli elementga unikal identifikator beriladi.

CSS

`#supersize {font-size: 200%}`

Supersize identifikatorli elementning razmeri 200% belgilanadi.

HTML

`ADU web-sayti` Identifikator
HTML xujjatda “id=”supersize”” yordamida beriladi.

CSS uch hil uslubda qo'llanishi mumkin:

1. Tashqi faylda:

.css fayl yaratiladi.

HTMLning <head> qismiga qo'shamiz:

`<link rel="stylesheet" type="text/css" href="style.css">`
(css adresi)

Bu yerda CSS adresi internetda joylashgan xam bo'lishi mumkin.

2. HTML faylning <head> qismida:

```
<html>
  <head>
    <style>
    </style>
  </head>
  <body>
    </body>
</html>
```

CSS ko'rsatmalar <style> va </style> teglari orasida HTML faylning <head> qismida yoziladi.

3. Ichki stil:

HTMLda

`ZIYONET`

Stillar qo'llanish tartibi:

1. Ichki stil
2. html fayldagi stil.
3. Tashqi stil.

Selektorlar bo'yicha stillar qo'llanish tartibi:

1. ID
2. Klass
3. Teg

Stillarda vorislik amal qiladi. Yani tashqi elementning stili ichki elementlarga xam tegishli. Misol uchun <body> uchun stil ichidagi elemetlarga xam tasir qiladi.

CSS qo'llash uchun misol.

Index.html fayl matn taxrirlchida (misol uchun Notepad++) yaratamiz:

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8">
<title>Bizning web-sahifamiz</title>
<link href="css/style.css" type="text/css" rel="stylesheet">
</head>
<body>
<h1>Andijon Davlat Universiteti</h1>
<div>
<h2 class="fak">Filologiya fakulteti</h2>
<h2 class="fak">Biologiya fakulteti</h2>
<h2 class="fak">Fizika-matematika fakulteti</h2>
</div>
<p>Kasbiy ta'lim (Informatika va Axborot texnologiyalari) </p>
<div>
<ul>
<li>Birinchi kurs</li>
<li>Ikkinci kurs</li>
<li id="iat3">Uchinchi kurs</li>
<li>To`rtinchi kurs</li>
</ul>
</div>
<h2>2017-2018 o'quv yili</h2>
<a href="http://adu.uz"> ADU websayti</a>

</body>
</html>
```

CSS katalogida style.css faylni matn taxrirlchida yaratamiz:

```
li {color:red}
p {font-size: 150% }
```

```
.fak {color:blue      ;
      font -size:120%;
      }

#s oat {
      width:250px;
}

#iat3 {color: yellow; font      -size: 200%}

div {background-image: url(..../img/bg1.png);}
```

Ushbu misolda **li** va **div** teglar, **fak** nomli klass, **soat** va **iat3** identifikatorlar yordamida elementlarni ekranga chiqarish bezaklari boshqarilmoqda.

Nazorat savollari:

- 1.CSS selektori tarifini bering?
- 2.CSS sintaksisi?
- 3.Selektor turlari?
- 4.CSS joylashish turlari?
- 5.CSS qo'llanish tartibi?

5-ma’ruza. CSS xususiyatlari. CSS sahifalash

Reja:

1. CSS rang xususiyatlari.
2. CSSda chegara xususiyatlari.
3. CSSda orqa fon xususiyatlari.
4. CSS shrift xususiyatlari.
5. Margin xususiyatlari. Padding xususiyatlari. Box model tushunchasi.
6. Box model qoidalari.

CSS dagi asosiy xususiyatlari

- Orqa fon xususiyatlari (background)
- Yozuv xususiyatlari (text)
- Chegara xususiyatlari (border)
- Joylashuv xususiyatlari (box-model)

Orqa fon xususiyatlari (*background properties*)

- background-color – orqa fon rangi
- background-image – orqa fonga tasvir joylashtirish
- background-repeat – orqa fon tasvirining takrorlanish yoki takrorlanmasligi
- background-attachment – orqa fon tasvirining browser mashtabiga muvofiqligi
- background-position – orqa fon rasmining browserdagi joylashuv ko’rinishi

background-color

Istalgan turdag'i elementning orqa fon rangini o'zgartirish uchun ishlataladigan xususiyatdir.

```
h1 {  
    background-color: green;  
}  
  
div {  
    background-color: lightblue;  
}  
body{  
    background-color: yellow;  
}  
.
```

background-image

Sahifa orqa foni uchun rasm yoki biror tasvirni joylashtirish xususiyatidir.

```
body{  
    background-image: url("G:\\myCar.jpg");  
}  
h1{  
    color: lightblue;  
}
```

Yozuv xususiyatlari (text properties)

- color – yozuv rangi
- font-size - yozuv o'lchami
- font-family – yozuv shrift turi
- text-align – yozuv joylashuv o'rni (right – o'ngdan, left-chapdan, center-o'rtadan, justify – ikkala tomondan ham tartiblangan yozuv)
- text-decoration – yozuvga bezak berish (underline – tagchiziqli yozuv, overline-tepa chiziqli yozuv, linethrough – chiziq yozuv ustidan o'tishi)
- text-transform – yozuv harflarining katta-kichikligi (uppercase – katta harfli yozuvar, lowercase – kichik harfli yozuvar, capitalize – faqat bosh harfi katta yozuvar.
- text-indent – yozuv birinchi qatorining ekran yoki boshqa elementdan uzoqlashuvi
- letter-spacing – yozuv harflari orasidagi o'zaro masofani o'zgartirish
- line-height – yozuv va pastki qator yozulari orasidagi o'zaro masofani o'zgartirish
- direction – yozuvning qaysi tomonga yo'nalganligi (rtl – o'ngdan chapga, ltr – chapdan o'ngga)
- word-spacing – yozuvarlar oralig'idagi o'zaro masofani o'zgartirish

```
p{  
    color: blue;  
    font-size: 20px;  
    font-family: Papyrus;  
    text-align: justify;  
    text-decoration: underline;  
    text-indent: 50px;  
}
```

← → ⌂ ⌂ Файл | C:/Users/yadro/Desktop/sahifa.html

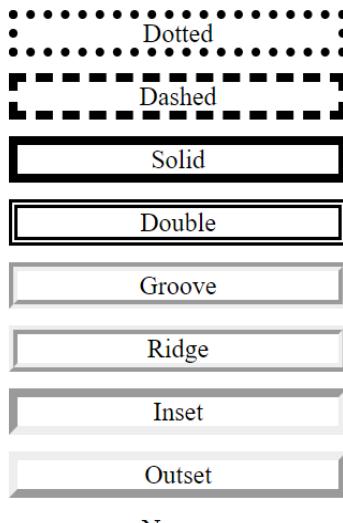
Lorem ipsum dolor sit amet, consectetur adipisicing élit.
Maxime, sapiente tempore rem obcaecati repudiandae nostrum
molestias illo adipisci. Id molestias veniam distinctio inventore
consequuntur impedit aut eos quas. Possimus, provident.

Chegara xususiyatlari (*border properties*)

- border-color – elementning chegara chizig’i rangi
- border-style – element chegara chizig’i turi
- border-width – elementning chegara chiziq qaliligi
- border-radius – element chegarasining radiusi

```
div{  
    border-color: red;  
    border-style: dotted;  
    border-width: 5px;  
}
```

- **border-style: dotted;**
- **border-style: dashed;**
- **border-style: solid;**
- **border-style: double;**
- **border-style: groove;**
- **border-style: ridge;**
- **border-style: inset;**
- **border-style: outset;**
- **border-style: none;**
- **border-style: hidden;**



Elementlarning o’zaro joylashuv xususiyatlari (box-model)

Har bir elementning boshqa elementlari bilan o’zaro joylashuvi, shuningdek aynan shu elementning subelementi ya’ni ichki elementlari bilan o’zaro joylashuvi, ular orasidagi masofa, elementning balandligi va kengligini ifodalovchi xususiyatlarga quyidagilar kiradi:

margin: element va tashqi soha, tashqi maydon, tashqi element orasidagi o’zaro masofa

padding: bitta element ichida joylashgan boshqa elementlar orasidagi o’zaro masofasi

height: element balandligi;

width: elementning kengligi.

border: elementning chegarasi (oldingi slaydlarda chegara xususiyatlari keltirilgan)



6-ma’ruza. CSS3 asoslari. Effektlar, transformasiya, animasiya.

Reja:

1. CSS3 asoslari.
2. CSS matn effektlari.
3. CSS da ikki o‘lchovli transformasiya.
4. CSS da uch o‘lchovli transformasiya.
5. CSS animasiyalar. CSS preprocessorlari (sass, less).

CSS da effektlar

Yozuv effektlari – Text effects(text-overflow, word-wrap, word-break)

Soya effektlari – Shadow effects (text-shadow, box-shadow)

Rang effektlari – Color effects(linear-gradients, radial-gradients)

Yozuv effektlari

text-overflow: overflow xususiyati bilan birlashtiriladi. Mazkur xususiyatlar yozuvning tashqi elementi maydoni o‘z chegarasidan tashqariga chiqib ketish yoki ketmasligini ta’minlovchi xususiyatlardir. (*nowrap, hidden, clip, ellipses*)

text-overflow: inherit va overflow: visible bo’lgan holda yozuv o‘z chegara maydonidan chetga chiqib ketadi va to’liq ko’rinishda ko’rinadi. Quyidagi misolda hover xususiyati ya’ni sichqoncha yozuv ustida harakatlangda yozuvning text-overflow xususiyati inheritga o’zgaradi va yozuv to’liq ko’rinishda ko’rinadi.

word-wrap: element chegarasidan oshib ketadigan uzun so’zlarni keying qatorga bo’g’in ko’cherish yo’li orqali , yoki butunligicha yoki qismga bo’lib ko’chirilishini ta’minlovchi xususiyatdir.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.test1 {
    width: 140px;
    border: 1px solid #000000;
    word-break: keep-all;
}

p.test2 {
    width: 140px;
    border: 1px solid #000000;
    word-break: break-all;
}
</style>
</head>
<body>
<p class="test1">Sig'magan so'zlarni harflarini buzmasdan pastki qatorga ko'chirish</p>
<p class="test2">Sig'magan so'zlarni harflarga ajratib pastki qatorga ko'chirish</p>
</body>
```

SOYA effektlari

text-shadow: yozuvga beriladigan soya effekti xususiyati.

Qiymatlari gorizontal va vertikal soha tushirilishiga qarab beriladi.

Masalan: *text-shadow: 2px, 4px; - 2px gorizontal va 4px vertikal holda soya hosil qiladi.*

text-shadow: 4px 5px red; - 4px gorizontal va 5px vertikal holda qizil rangli soya hosil qiladi.

Agar soyaning gorizontal va vertical siljishi qiymatlaridan keyin yana bitta qiymat kiritilsa soyada blur(bo'yashgan) effect hosil bo'ladi:

text-shadow: 2px 3px 5px red;

2px – gorizontal

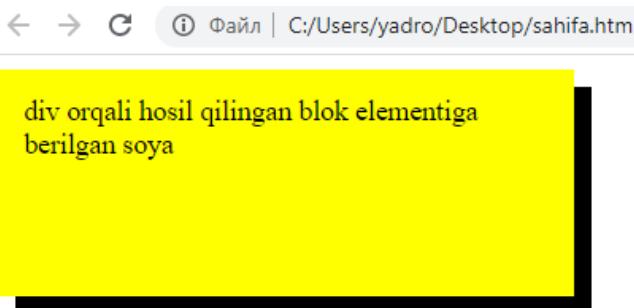
3px – vertical

5px – blur effect

red – qizil rangli soya

box-shadow: elementning tashqi chegarasiga beriladigan soya effektidir.

```
div {  
    width: 300px;  
    height: 100px;  
    padding: 15px;  
    background-color: yellow;  
    box-shadow: 10px 10px;  
}
```



Elementning gorizontal va vertical soya qiymatlarini 0 qilish orqali hamma tomonidan bir xil tushadigan soya hosil qilish mumkin,

```
sahifa.htm|  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div {  
    width: 300px;  
    height: 100px;  
    padding: 15px;  
    background-color: white;  
    box-shadow: 0 0 10px grey;  
}  
</style>  
</head>  
<body>  
<div>Blur effektli, hamma tomonidan bir xil  
tushirilgan soya effekti</div>  
</body>  
</html>
```

Blur effekti, hamma tomondan bir xil tushirilgan soya effekti

Rang effektlari

Gradient bu bir necha xil ranglarni qo'shib yangicha rang effektini hosil qilishdir. CSS3da asosiy ikki xil turdag'i gradient mavjud:

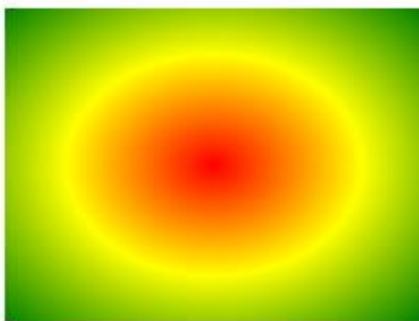
linear-gradient: – bir necha xil ranglarni yonma-yon chiziqli aralashtirish

radial-gradient: – bir necha xil ranglarni aylanasimon aralashtirish

```
background: linear-gradient(red, yellow);
```



```
div {  
    height: 150px;  
    width: 200px;  
    background: radial-gradient(red, yellow, green);  
}
```



Transformatsiya

Transformatsiya – elementning ayni holatdagi shaklini o'zgartirishdir. Tranformatsiya xususiyatida bir nechta asosiy funksiyalar ishlataladi:

transform: scale(50%). Element hajmini o'zgartiruvchi funksiya

transform: scaleX(50%). Element hajmini X o'qi bo'yicha o'zgartirish

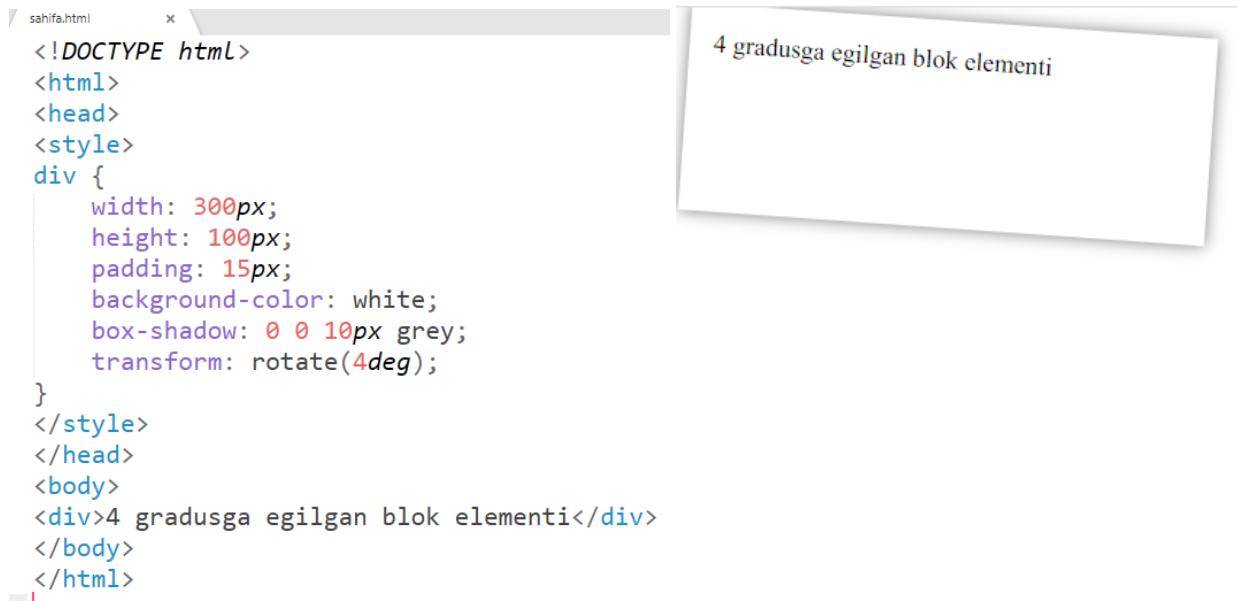
transform: scaleY(50%). Element hajmini Y o'qi bo'yicha o'zgartirish

transform: rotate(5deg). Elementni biror gradusga burish, qiyshiq holga keltirish.

transform: rotateX(50deg). X o'qi bo'yicha biror gradusga burish

transform: rotateY(90deg). Y o'qi bo'yicha biror gradusga burish

transform: rotate(4deg); elementni 4 gradusga burish.



The screenshot shows a browser window with a single red rectangular div element centered on the page. The element has a thin black border and is rotated 4 degrees clockwise. To the left of the browser window, the source code for a file named 'sahifa.html' is displayed in a code editor. The code includes an HTML structure with a head and body section, and a CSS style block that defines a red div with a width of 300px, height of 100px, padding of 15px, white background-color, a 10px grey box-shadow, and a transform: rotate(4deg) rule.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 300px;
    height: 100px;
    padding: 15px;
    background-color: white;
    box-shadow: 0 0 10px grey;
    transform: rotate(4deg);
}
</style>
</head>
<body>
<div>4 gradusga egilgan blok elementi</div>
</body>
</html>
```

4 gradusga egilgan blok elementi

Transition – element o'zgarish jarayonini biror vaqtga kechiktirish, yoki ma'lum vaqt bo'yicha o'zgartirishni amalga oshiradi.

CSS da animatsiya

Animatsiya bu biror elementning harakati, holati, dizayni, stillarining vaqtga mos holda o'zgarish jarayonidir. Ko'p hollarda Web sahifalarda animatsiya asosan JavaScript tili yoki Flash muhitida yordamida amalga oshiriladi. Biroq CSS3 ning o'zida ham murakkab bo'limgan turli xildagi animatsion jarayonlarni hosil qilish mumkin. CSS3 yordamida animatsiya hosil qilishda eng muhim xususiyat va funksiyalar quyidagilardir.

@keyframes – vaqt ichida o'zgarishni ta'minlovchi qo'shimcha element (Standart holatda ishlataladi)

@-webkit-keyframes - vazifasi @keyframes bilan bir xil (Chrome, Opera, Safari brauzerlari qo'llab quvvatlaydi)

@keyframes ga tegishli bo'lgan muhim xususiyatlar:

animation-name: example; - @keyframes elementining nomi

animation-duration: 4s; - animatsion jarayon vaqtini

animation-delay: 2s; animatsiya boshlanishini biror vaqtga kechiktirish

animation-iteration-count: 4; animatsiyaning necha bora takrorlanish soni

animation-direction: reverse; animatsiya yo'naliшини o'zgartirish

from: animatsiya boshlanishidan avvalgi dastlabki holat;

to: animatsiya dastlabki holatidan keying holatga o'tishi;

7-ma'ruza. JavaScript asoslari. JavaScript ssenariyli tilining xususiyatlari

Reja:

1. JavaScript asoslari.
2. JavaScript ssenariylarini HTML-xujjat bilan bog'lash.
3. O'zgaruvchilar. Ma'lumotlar turlari.
4. Massivlar.
5. Operatorlar.

Kalit so'zlar: shartli va boshqaruvi o'tkazuvchi operatorlar, sikl operatorlari, switch operatori.

1. JavaScript tilida shart operatorlari

JavaScript tilida bir necha shart operatorlari bor. Hozir ulardan bazilari bilan tanishib chiqamiz.

JavaScript tilida if operatori

Bu operator JavaScript dasturlash tilidagi muhim operatorlardan biridir. U shartga bog'liq ravishda kod fragmentini bajarishga mo'ljallangan. if operatorining strukturasini quyidagicha ifodalash mumkin:

1- variant. if operatori sintaksisi:

```
if(shart)
{
    Amallar ketma ketligi
}
```

variant. if operatori sintaksisi:

```
if(shart)
{
    Amallar ketma ketligi 1
} else
{
    Amallar ketma ketligi 2
}
```

Bu erda shart JavaScript tilidagi mantiqiy yoki ixtiyoriy shartdir. Agarda shart mantiqiy tipdagi o'zgaruvchi bo'lsa, qiymati rost (True) bo'lsa, u holda 1-amallar ketma ketligi bajariladi. Aks holda 2amallar ketma ketligi bajariladi.

Quyida ***if*** shart operatoriga doir misol ko'rib o'tamiz:

```
function checkData()
{
```

```
if (document.form1.threeChar.value.length==3)
{ return true;
} else
{ alert('3 raqamini kriting'); return false;
}
}
```

JavaScript tilida if operatori obraz kabilardan foydalanadi. If buyrug'i bo'yicha standart operatorga misol keltiramiz:

```
if (a == 1} window.alert("1 topildi!");
```

if operatori bir necha harakat operatorlaridan ham foydalanadi:

```
1: if (a == 1) {
2: window.alert("1 topildi!");
3: a = 0;
4; }
```

Bu operatorlar bloki a o'zgaruvchi ustida tekshiruv o'tkazadi va agar u 1 ga teng bo'lsa, u holda habar aks ettiriladi va a o'zgaruvchi yangi qiymat 0 ga aniqlanadi.

JavaScript tilidagi shart operatorlarida quyidagilardan foydalaniladi:

- == (teng)
- != (teng emas)
- < (kichik)
- > (katta)
- <= (kichik yoki teng)
- => (katta yoki teng)

Java Script tilida else operatori

Biz yuqorida faqat if operatorining asosiy qisminigina ko'rdik. Bu operatorning bir nechta kengaygan shakli mavjud. else operatori if operatorida tekshirilayotgan ifoda noto'g'ri bo'lgan holatdagina kengaytiradi hamda bu holatda yangi shartda biror amal bajaradi.

else operatori yordamida kengaytirilgan if operatorining strukturasini quyidagicha ifodalash mumkin:

```
if (shartli_ifoda)
bajariladigan_blok1 else
bajariladigan_blok2
```

Bu if...else konstruktsiyasi quyidagicha interpretatsiya qilinishi mumkin: agar shart bajarilsa (ya'ni ifoda=true), u holda bajariladigan_blok1 dagi amallar bajariladi, aks holda bajariladigan_blok2dagi amallar bajariladi. else operatoridan foydalanish majburiy emas. Quyida else operatoriga doir misol ko'rib o'tamiz:

```
1: if (a == 1) {  
2: alert("1 topildi");  
3: a = 0;  
4: }  
5: else {  
6: alert("Noto'g'ri belgi: " + a);  
1: }
```

JavaScript tilida elseif operatori

if shart operatorining yana bir kengaygan shakli – bu elseif operatorining qo'llanilishidir. elseif – bu

else hamda if operatorlarining kombinatsiyasidir. U xuddi else operatori kabi if operatorida shart bajarilmagan holda kengaytiradi. Biroq else operatoridan farqi bir-biriga zid amallarni faqat agarda elseif shart rost bo'lgandagina bajaradi. else hamda elseif operatorlari yordamida kengaytirilgan if operatorining strukturasini quyidagicha ifodalash mumkin:

```
if (ifoda1)  
bajariladigan_blok1 elseif  
(ifoda2) bajariladigan_blok2  
else bajariladigan_blokN
```

elseif operatorlari bitta if-blokida bir necha marta uchrashi mumkin. elseif tasdig'i faqat oldinda turgan

if-shartlari hamda elseif-shartlari False qiymatni, berilgan elseif-sharti esa True qiymatni qaytargandagina bajariladi.

Quyida elseif operatoriga doir misol ko'rib o'tamiz:

```
function makeMinutes() { var minstring = ""; var now = new Date(); var min = Date.getMinutes(); if (min < 10) { minstring += ":0" + min; } elseif (min > 10) { minstring += ":" + min; } return minstring; }
```

2. Boshqaruv o'tkazuvchi operatorlar

Ba'zida sikl yoki uning alohida iteratsiya ishini tezda to'xtatishga to'g'ri keladi. Buning uchun **break** hamda **continue** boshqaruv o'tkazuvchi operatorlari kerak bo'ladi.

JavaScript tilida break operatori

Break operatori mavjud siklni amalga oshirishni tugallaydi, for, while, do while yoki switch break strukturani boshqaruvchi, sikl yoki shartni tekshirishni tugallash kerakligini bildiruvchi, uning tarkibiga kiruvchi raqamlı argument bilan qo'llaniladi.

Break operatoriga doir bo'lgan misolni ko'rib chiqamiz:

```
for (i = 0; i <  
a.length; i++) { if
```

```
(a[i] = theValue)
break;
}
```

Ushbu skriptda a massiv elementi theValue o'zgaruvchi qiymatiga teng bo'lganda sikl to'xtatiladi.

JavaScript tilida continue boshqaruv o'tkazuvchi operatori

Ba'zan sikl ishini butunlay to'xtatish lozim bo'lmaydi, faqatgina uning yangi iteratsiyasini boshlash kerak. Continue operatori istalgan siklni amalga oshirish blokidan keyingi instruktsiyalarni o'tkazib yuborish va yangi doira bilan amalga oshirishni davom ettirish imkonini beradi. continue ni uning tarkibida boshqariluvchi konstruktsiyalar ishini yakunlash kerakligini ko'rsatuvchi raqamli argument tarzida ishlatish mumkin.

Oldingi paragrafda berilgan misoldagi break operatorini continue ga almashtiramiz. Bundan tashqari to'rt sikli miqdorini kamaytiramiz.

Misol:

```
i = 0; n = 0; while (i < 5) {      i++;
    if (i ++ 3)      continue;      n ++ i;
}
```

Bu skriptda agarda i uzgaruvchi i=3 qiymatga ega bo'lsa, u xolda n=(1,3,7,12) qiymatlarga ega bo'ladi.

Misol:

```
Check i and j :    while (i<4) {
    document.write(i + "<BR>");           i++1;           checkj :
while (j>4) {
    document.write(j + "<BR>");           j-=1;
if ((j%2)++0)          continue checkj;
    document.write(j + " is odd.<BR>");
}
document.write("i = " + i + "<br>");
document.write("j = " + j + "<br>");
}
```

Ushbu siklda agarda continue operatori topilsa, check j sikli to'xtatiladi va checkj ning keyingi sikliga o'tiladi. Xar safar continue topilganda checkj iteratsiyani boshidan boshlaydi, toki uning sharti false bo'lmagunicha. checkj sharti false bo'lganda, checkiandj ning qolgan operatorlari bajariladi va bu sikl xam checkiandj sharti false bo'lguniga qadar davom etadi.

3. Sikllar

JavaScript tilida shartga bog'liq ravishda qaytariladigan amallardan iborat bir nechta konstruktsiyalar mavjud. Bu while, do..while, foreach hamda for sikllardir. Ularni batafsil ko'rib chiqamiz.

JavaScript tilida While sikli

While siklining strukturasi quyidagicha bo'ladi:

```
while (ifoda) { bajariladigan_blok }
yoki while (ifoda):
bajariladigan_blok endwhile;
```

while – bu oddiy sikl. U ifoda qiymati True (bu erda xuddi if operatori kabi ifoda mantiqiy tipga o'zlashtiriladi) bo'lgunicha bajariladigan_blokdagi buyruqlarni bajarishga buyuradi. Ifoda qiymati har sikl boshlanganda tekshirib boriladi, agarda uning qiymati bajariladigan_blok bajarilish jarayonida o'zgargan taqdirda ham iteratsiya tugamaguncha (ya'ni bajariladigan_blokdagi barcha buyruqlar bajarilmaguncha) sikl to'htatilmaydi.

Misol:

```
n = 0; x = 0; while( n < 3 ) {     n++;      x += n;
}
```

JavaScript tilida do... while sikli

do..while sikli while siklga o'xshaydi, ammo farqli tomoni shundaki, ifodaning rostligiga sikl boshida emas, balki oxirida tekshiriladi. Qulay tomoni shundaki, bajariladigan_blok do..while sikli ichida hech bo'lmasganda bir marta bajariladi. Strukturasi quyidagicha:

```
do { do..while sikli } while (ifoda);
do..while operatori asosidagi misolni ko'rib
o'tamiz: do {    i+=1;
    document.write(i);
} while (i<5);
```

JavaScript tilida for sikli

Bu JavaScript tili sikl operatorlaridan biri hisoblanadi. Ular C dasturlash tilidagi sikllar kabitidir. Strukturasi quyidagicha:

```
for (ifoda1; ifoda2; ifoda3) {
bajariladigan_blok } yoki
for (ifoda1; ifoda2; ifoda3): bajariladigan_blok endfor;
```

Bu erda ko'rinish turibdiki shart uchta ifodadan tashkil topadi. Birinchi ifoda1 ifoda sikl boshida shartsiz bajariladi. Har bir iteratsiyaning boshlanishida ifoda2 bajariladi. Agar u True

qiymatni qabul qilsa, u holda sikl o'z ishini davom ettiradi va bajariladigan_blokdag'i barcha buyruqlarni bajaradi. Agar ifoda2 False qiymatni qabul hilsa, u holda sikl to'xtatiladi. har bir iteratsiya (ya'ni bajariladigan_blokdag'i barcha buyruqlarni bajarilishidan keyin) oxirida ifoda3 bajariladi.

Har bir 1-,2- va 3-ifodalar bo'sh bo'lishi mumkin. Agar ifoda2 bo'sh bo'lsa, bu siklni cheksiz bajarilishini bildiradi. Bu unchalik befoyda emas, chunki siklni break operatori yordamida to'xtatsa bo'ladi.

Misol: Formada tanlangan elementlar sonini ekranga chiqarish;

```
<SCRIPT>
function
howMany(selectObject) {
var numberSelected=0;
    for (var i=0; i < selectObject.options.length;
i++) {        if
(selectObject.options[i].selected==true)
numberSelected++;
}
    return numberSelected;
}

</SCRIPT>
<FORM NAME="selectForm">
<P><B>Choose some music types, then click the button below:</B>
<BR><SELECT NAME="musicTypes" MULTIPLE>
<OPTION SELECTED> R&B
<OPTION> Jazz
<OPTION> Blues
<OPTION> New Age
<OPTION> Classical
<OPTION> Opera
</SELECT>
<P><INPUT TYPE="button" VALUE="How many are selected?">
onClick="alert('Number of options
selected:' +howMany(document.selectForm. musicTypes))">
</FORM>
```

Agarda ushbu operator ichidagi barcha uchala ifoda ham tushirib qoldirilsa, u holda schyotchik var i o'zgaruvchini boshlang'ich qiymati berilmaydi va har bir sikl oxirida u o'zgarmaydi. Bu barcha buyruqlarni alohida buyruqlar ko'rinishida yoki sikldan avval bajariladigan_blok ichida yozsa ham bo'ladi:

```
var i=0; // hisoblagichni boshlang'ich
qiymatini beramiz for ( ; ; ){      if (i>=10)
break;
// agar i katta yoki teng 10 bo'lsa, u holda sikl ishini
to'xtatamiz.      if (i % 2 == 0) print $i;
```

```
// agar son juft bo'lsa, uni ekranga  
chiqaramiz. i++; // schyotchik qiymatini bittaga  
oshiramiz.  
}
```

for sikli konstruktsiyasidagi uchinchi ifodada verguldan keyin yana bir nechta oddiy buyruqlarni ham yozsa bo'ladi. Masalan, agar biz oddiygina barcha sonlarni ekranga chiqarmoqchi bo'lsak, dasturni quyidagicha yozsa bo'ladi:

```
for (i=0; i<10; document.write(i), i++)  
/* Agarda bajariladigan_blok buyruqlardan  
tashkil topmagan yoki bitta buyruqdan tashkil  
topgan bo'lsa, figurali qavsga olingan qismni  
tashlab ketsa bo'ladi.*/
```

4. JavaScript tilida Switch operatori

Yana bir shartni tekshirib turli amallarga bog'liq ravishda ish ko'rsatadigan konstruktsiya bu – switch operatoridir. Bu operatorni o'zbek tiliga tarjima qilinganda “yo'nalishni o'zgartirgich” ma'nosini beradi hamda bu operatorning vazifasi ham shunga o'xshashdir.

Switch operatori bir nechta asosiy elementlardan tashkil topgan:

- Switch boshlanish operatori. Bu operator qavs ichida kiritilgan belgilarni taqqoslaydi.
- Figurali qavslar {}. If operatoridagi qavslar kabi vazifani bajaradi. □ Bir yoki bir necha case operatori.
- Break kalit so'zi. Case operatoridan chiqish harakatini aniqlash uchun ishlataladi.

O'zgaruvchini qanday qiymatni qabul qilishiga bog'liq ravishda u yo'nalishni o'zgartirib turli bloklardagi amallarni bajaradi. switch operatori if...elseif...else yoki if operatori majmuiga juda o'xshash bo'ladi. switch operatorining strukturasini quyidagicha ifodalash mumkin:

```
switch (ifoda yoki  
o'zgaruvchi){ case qiymat1:  
  
amallar_b  
loki1  
break;  
case  
qiymat2:  
    amallar_bloki2 break;  
... default:  
amallar_bloki_avtomatik_t  
arzda  
}
```

if operatoridan farqli tomoni bu erda ifodalar mantiqiy tip qabul qilmay, balki faqat case kalit so'zidan

keyingi qiymatlarni (qiymat1, qiymat2 va h.k.) taqqoslaydi xolos. Agar ifoda qiymati qandaydir variant bilan ustma-ust tushsa, ikki nuqtadan keyingi break operatoriga bo'lgan amallar_blokidagi amallarni bajaradi. Agarda ifoda qiymati berilgan variantlardan hech biriga ustma-ust tushmasa, default kalit so'zidan keyingi avtomatik tarzda bajariladigan blok (amallar_bloki_avtomatik_tarzda) bajariladi. switch operatoridagi ifoda faqat bir marta hisoblanadi, elseif operatorida esa har bir tekshirishda hisoblanadi, shuning uchun agarda ifoda etarli darajada murakkab bo'lsa, u holda switch operatori tezroq ishlaydi.

Quyida switch operatoriga doir misollar ko'rib o'tamiz:

```
function getName(){
var names = array("Name1", " Name2",
"Name3"); var selected=""; switch
(names[0]){
case "Name1": selected="Name1 is
selected"; break;
case "Name2": selected="Name2 is
selected"; break;
case "Name3": selected="Name3 is
selected"; break;
default: selected="Default select"=$names[0];
}
return selected;
}
```

Yuqoridagi misoldan ko'rinish turibdiki, getName() funksiyasi ishga tushirilganda, names massividagi birinchi element tekshiriladi. Bu xolda massiv 1-elementi Name1 bo'lgani uchun bizga selected="Name1 is selected"; qiymatni qaytaradi.

2-misol.

```
var change =prompt("Xarakatni tanlang:/n1 - Mashina sotib olish/n2
-
Mashina sotish /n3 - Mashina
almashtirish "); switch (change)
{ case "1": {
    document.write("Siz mashina sotib olishni xoxlaysiz
"); break;
}
case "2": {
    document.write("Siz mashina sotishni
xoxlaysiz "); break; } case "3": {
    document.write("Siz mashina almashtirishni
xoxlaysiz "); break; } default: {
    document.write("Siz noto'g'ri buyruq kiritdingiz
"); break;
}
}
```

switch operatorining konstruktsiyasi uchun xuddi if operatori kabi alternativ sintaksisi mavjud. Bu erda switch operatoridagi ochiladigan figurali qavs ikki nuqtaga o'zgartiriladi, yopiladigani esa mos ravishda **endswitch** kalit so'ziga o'zgartiriladi.

Nazorat savollari:

1. JavaScript tilida shart operatorlarining ishlashi va vazifasini tushuntirib bering.
2. Boshqaruv o'tkazuvchi operatorlar nima va ularning vazifasi nimalarda namoyon bo'ladi?
3. Qanaqa sikl operatorlarini bilasiz va ular qanday ishlaydi? 4. Switch operatori ishlash prinsipini tushuntirib bering?
5. Turli shart operatorlarining bir-biridan farqli tomonlarini keltiring? 6. Qay holatda qaysi shart operatoridan foydalanish qulay deb o'ylaysiz?
7. Har bir operatorni tegishli misollar yordamida tushuntiring?
8. Switch operatoridan foydalanish qanday afzalliklarga ega?
9. JavaScript tilidagi operatorlar va sikllarning boshqa dasturlash tillaridagi operatorlardan farqli jihatlari bormi? 10. Boshqaruv o'tkazuvchi operatorlarning ishlash prinsipini ko'rsating?

8-ma'ruza. JavaScript funksiyalari va ob'yektlari

Reja:

1. JavaScriptda funksiyalar.
2. JavaScriptda ob'yektlar.
3. JavaScript funksiyalari yordamida hodisalarni qayta ishslash.
4. JavaScriptda doimiy ifodalar.
- 5.

Kalit so'zlar: JavaScript tilidagi funksiyalar, Global, Math, Date sinflari, ob'ektlar,kattaliklar, sinflar, usullar, ob'yektni model.

1. JavaScript tilida funksiyalar

JavaScript ssenariyli tili ob'ektga-mo'ljallangan tildir. JavaScript ob'ektlari xususiyatlar va usullar to'plamini ifodalaydi. Ob'ekt xususiyati – bu, ob'ektga bog'liq bo'lgan ma'lumotlardir, usullar esa - ob'ekt ma'lumotlarini qayta ishlovchi funksiyalardir. JavaScript ssenariyda xususiyatlarni adreslash ularning nomlari bilan yoki ularning nomerlari bilan amalga oshishi mumkin. Keyingi variant bo'yicha, har bir xususiyat massivning bir elementi sifatida olinadi va ular o'zlarining unikal nomerlariga ega bo'ladilar.

JavaScript tilida C va Java dasturlash tillaridagi kabi protsedura va funksiyalar mavjud bo'lib, ular quyidagicha e'lon qilinadi:

```
function kalit so'zi; funksiya nomi; vergul
va qavs bilan ajratilgan argumentlar
ro'yxati;
figurali qavs ichiga olingan
funksiya tanasi. function
myFunction(arg1, arg2, ...) { ...
Operatorlar ketma-
ketligi ...
}
```

Bu erda:

myFunction – funksiya nomi, arg1, arg2 – parametrlar.

Misol:

```
function Factorial(n) {
if((n<0)|| (round(n)!=n)) {
alert("Factorial funksiyasi ushbu argumentda
aniqlanmadi "+n); return NaN; } else {
Result=(n*Factorial(n-
1)); return result;
} }
```

Funksiyada return kalit so'zi orqali qiymat qaytarilmasligi ham mumkin.

Misol:

```
function Greeting(s)
{
document.write("Hello,
"+s+"!"); return ;
}
```

Funksiyani chaqirish aniq parametrlar bilan chaqiriladi:

Misol:

Factorial(3); - bu funksiya natijasi 6 ga teng,
Greeting("world"); - bu funksiya ekranga "Hello, world!" satrini chiqaradi.

Har bir funksiya, masalan, myFunction funksiyasi myFunction nomli ob'ekt xisoblanadi, agarda argumentlar arguments nomi bilan berilsa, unga murojaat quyidagicha:

`myFunction.arguments[i]`, bu erda `i` — argumenta nomeri (rahamlash 0 dan boshlanadi).

Funksiya e'lonida aniq parametrlar formal parametrarga teng eki ko'p sonda bo'lishi lozim. Bunda funksiya ishga tushirilganda jo'natilayotgan argumentlar miqdori `myFunction.arguments.length` maydoni yordamida aniqlanadi va ushbu maydondagi qiymatni qayta o'zlashtirishni dinamik o'zgartirish mumkin.

Misol:

Ekranga HTML formatidagi ro'yoxatni chiqarish.

Bu erda (`ListType`) ning birinchi argumenti tartiblanmagan ro'yxat uchun "o" eki "O", tartiblanmagan ro'yxat uchun "u" eki "U" bo'lishi mumkin.

```
function myList(ListType)
{
document.write("<"+ListType+"L");
for(var i=1; i < myList.arguments.length; i=i+1) {
document.write("<LI>"+myList.arguments[i]);
}
document.write("</"+ListType+"L>");
}
```

HTML hujjatida funksiyaga murojaat quyidagicha amalgam oshiriladi:

```
<script>
myList("o", "matn", 2, "3")
</script>
```

Natija: matn 2

2. Global klassi

Ushbu klass JavaScript ning funksional qismi bo'lib, bu klass bir ob'ektda bir nechta usul va xossalarni birlashtirish vazifasini bajaradi. Usulga murojaat qilinganda ob'ekt ko'rsatilmaydi, aniqrog'i bu usul konstruktorga ega bo'lmaydi. Bunday xossa va usullarga quyidagilarni keltirish mumkin:

Hossa	Mazmuni
Nan	NaN (Not A Number)
Infinity	Number.POSITIVE_INFINITY qiymatni o'z ichiga oladi.

Usul	Mazmuni
escape	Qatorni barcha platformalarga mos holda tasvirlash
eval	JavaScript tili funksiyasi yoki usullarini uzatish
isFinite	Argumentning oxirgi raqamliligini aniqlash
isNaN	Argumentning raqam yoki raqam emasligini aniqlash
parseFloat	Qatorni ko'chib yuruvchi nuqtali son ko'rinishida tasvirlash
parseInt	Qatorni butun songa aylantirish
unescape	Escape funksiyasi natijasini qaytarish

E va l(s) funksiyasi - s qatorni JavaScript operatorlari ketma-ketligi ko'rinishida tasvirlash.
getClass(Jobj) funksiyasi – JavaObject tipidagi argument uchun JavaClass ob'ektini qaytaradi.

Misol:

```

var myJavaRClass=new
java.awt.Rectangle() var
myJavaRClass=getClass(myJavaRect)
getClass() Java-metodi bilan
adashtirmang:
var myJavaRObject=myJavaRect.getClass()      - bu
java.awt.Rectangle
klasining Java tilidagi realizatsiya holati.
isNaN(x) funksiyasi - x "Not a Number", ya'ni son emasligini
tekshirish.
parseFloat(s) funksiyasi - Float tipidagi s rahamni aniqlash. Agar
son
topilmasa u holda NaN ("Not a Number") qiymati qaytariladi.
parseInt(s) – Integer tipi uchun yuqoridagi holat.

```

eval(s) funksiyasi

`eval(s)` funksiyasi – JavaScript ning ichki funksiyasi hisoblanadi. Ushbu funksiya bir yoki bir nechta JavaScript operatorlaridan iborat bo’lgan s satrni argument tomonidan uzatilgan kodni bajaradi. Bunda s satridagi operatorlar nuqtali vergul yordamida ajratiladi. Bu funksiya nafaqat operatorni bajarish, balki biror amallarni hisoblash imkonini ham beradi. Bunda u kodda keltirilgan amal xisobining oxirgi qiymatini qaytaradi.

isNaN(x) funksiyasi

Bu funksiya `x` argumentning “son emas” ligini tekshiradi. Natija `NaN` qiymatga ega emasligini, ya’ni mumkin bo’lmagan son (masalan, nolni nolga bo’lish natijasi) ni tekshiradi. Ushbu funksiya JavaScript da literal ko’rinishda `NaN` qiymatni berish mumkin emasligi uchun muhimdir. Bundan tashqari `parseFloat(s)` va `parseInt(s)` funksiyalar natijalarini tekshirish (mumkin bo’lgan son ekanligini) va arifmetik xatolar mavjudligi, masalan, nol soniga bo’lish mavjudligini tekshiradi.

parseFloat(s) funksiyasi

`s` satrini sintaktik analiz qilish va dastlab raqamni qaytarish (satrni raqamgga aylantiradi). `parseFloat(s)`

da `s` satrida ruxsat etilmagan raham elementlari (masalan, belgilar, raham, o’nli vergullar, daraja ko’rsatkichi va hokazo) mavjud bo’lsa analiz to’xtatiladi va qiymat qaytariladi. Agarda `s` satrda son bilan boshlanmasa, u holda `parseFloat(s)` funksiya `NaN` qiymatni qaytaradi.

parseInt(s) funksiyasi

Bu funksiya satrni butun songa aylantiradi. `parseInt(s)` funksiyadagi `s` satrda hisoblash tizimida

ko’rsatilmagan qiymatlarga ega bo’lganda sintaktik analiz to’xtatiladi va qiymat qaytariladi. Odatda, `parseFloat` va `parseInt` funksiyalar `s` satr son bilan boshlanmaganda `NaN` qiymat qaytaradi.

`parseInt(s,n)` holatida `n` asos hisoblanib, agarda `n=10` bo’lsa, `parseInt(s)` funksiya satrdagi 10 lik sanoq sistemasidagi sonlarni tekshiradi. `N=8` bo’lsa, 8 lik sanoq tizimidagi sonlar mavjudligini (bunda `n` 0 dan 7 gacha bo’lgan sonlar qiymatiga ega bo’lishi mumkin). `N=16` bo’lsa, 16 lik sanoq tizimidagi sonlar mavjudligini (bunda 0 dan 9 gacha bo’lgan sonlar va A dan F gacha bo’lgan harflar qiymatiga ega bo’linadi). Agarda `n=0` bo’lsa yoki qiymat berilmasa, u holda `parseInt(s)` funksiya satrning o’zidan asosni aniqlaydi. Bu holatda agarda satr `0x` bilan boshlansa, unda funksiya satrning qolgan qismini 16 lik sanoq tizimidagi son sifatida analiz qiladi, agarda satr `0` dan boshlansa, satr 8 lik sanoq tizimidagi qiymat sifatida analiz qilinadi.

3. Math klassi

Math – konstantalar va metodlardan iborat klassdir. Ular ob’ekt uchun odatdagidek murojaat qilinadi: `Math.konstanta`

`Math.funksiya(i..)`

Math klassi konstantalari

- E – e soni (natural logorifm asosli)

- LN10 — 10 li natural logorifm ($\ln 10$ soni) □ LN2 — 2 li natural logorifm ($\ln 2$ soni)
- LOG10E — 10 asosli e lagorifm ($\log_{10} e$ soni) □ LOG2E — 2 asosli e lagorifm ($\log_2 e$ soni)
- PI — pi konstantasi ("pi" soni)
- SQRT1_2 — 2 ning teskari kvadrat ildizi ($1/\sqrt{2}$)
- SQRT2 — 2 ning kvadrat ildizi ($\sqrt{2}$)

Math klassi metodlari

- abs(x) (x-son yoki ifoda) – absolyut qiymatni hisoblash;
- acos(x) (x bu erda [-1.0;1.0] radian intervaldagi son yoki ifoda) – arkkosinusni hisoblash qaytariladigan qiymat 0 dan pi radian oralig'ida bo'ladi.
- asin(x) (x bu erda [-1.0;1.0] radian intervaldagi son yoki ifoda) – arksinusni hisoblash qaytariladigan qiymat -pi/2 dan pi/2 radian oralig'ida bo'ladi.
- atan (x) (x – son yoki ifoda) – arktangensni radianlarda hisoblash. qaytariladigan qiymat -pi/2 dan pi/2 radian oralig'ida bo'ladi.
- atan2(x,y)(x,y — to'g'ri burchakli kordinata sistemasi kordinata nuqtalari) – Qutb kordinatasida (x,y) nuqtalar burchagini hisoblaydi. Qiymati 0 dan 2pi radian oralig'ida bo'ladi.
- ceil(x) (x — son yoki sonli ifoda) – sonni butun songa yo'naltirilgan holda yaxlitlash. Manfiy sonlar 0 soni yo'nalishiga qarab yaxlitlanadi.
- cos(x) (x – radiandagi burchak) – kosinusni hisoblash, qaytariladigan qiymat -1.0 dan 1.0 radian oralig'ida bo'ladi.
- sin(x) (x – radiandagi burchak) – kosinusni hisoblash, qaytariladigan qiymat -1.0 dan 1.0 radian oralig'ida bo'ladi.
- Exp(x) (x — son yoki sonli ifoda) — e eksponentsini hisoblash.
- Floor(x) (x — son yoki sonli ifoda) – sonni butun qismiga yo'naltirib yaxlitlash, masalan, floor(-1,1) teng (-2); floor(1,1) teng 1.
- Log(x) (x — musbat son yoki ifoda) – natural lagorifmni hisoblash.
- max(a,b) (a,b — son yoki ifoda) – ikki qiymatdan kattasini qaytaradi.
- min(a,b) (a,b — son yoki ifoda) – ikki qiymatdan kichigini qaytaradi.
- pow(x,y) — x ni hisoblash (birinchi argumentni darajaga ko'tarish). □ random — 0 dan 1 gacha intervaldagi tasodifiy sonlarni hisoblash.
- round — sonni butun qismiga qarab yaxlitlash (masalan, round(15.5) natijasi 16 ni beradi, round(-15.5) daet -15).
- Math.round(x) (x — son yoki ifoda)
- Math.sin(x) (x — radianda berilgan burchak)
- Math.sqrt(x) (x — 0 ga teng yoki katta bo'lgan son yoki ifoda) □ tan —tangensni hisoblash.
- Math.tan(x) (x — radianda berilgan burchak)

9-10-ma’ruza. JavaScript brauzer va veb-hujjat ob’yektlari modeli bilan ishlash Reja:

1. Hujjatning ob’yektlari modeli haqida tushuncha (DOM).
2. Hujjatning ob’yektlari modelida HTML ob’yektlari bilan ishlash.
3. Hujjatning ob’yektlari modelida CSS xususiyatlari bilan ishlash.
4. JavaScript brauzer ob’yekti modeli (BOM).
5. Date ob’yekti usullari va xususiyalari

JavaScript tilidagi ob'ektlar

JavaScript tilida uch turdagи ob'ektlar mavjud:

1. standart ob'ektlar;
2. brauzer ob'ektlari;
3. dasturchi tomonidan yaratiluvchi ob'ektlar.

Ularning har biri o’zlarining tasnifi va xususiyatlariga ega.

Standart ob'ektlar

Quyida JavaScript da qo’llaniluvchi standart ob'ektlar, xususiyatlar va usullar keltirilgan.

Ularni ishlatishda oldindan e’lon qilish talab etilmaydi.

Ob'yekt	Tasnifi
Array	Massiv
Boolean	Mantiqiy ma'lumotlar
Date	Kalendarli vaqt
Function	Funksiya
Global	Global usullar
Math	Matematik konstanta va funksiyalar
Number	Son
Object	Ob'yekt
String	Satr

Standart ob'ektlar bilan qanday ishlash mumkin? Ancha oddiy. Ob'ektni realizatsiya hiluvchi dastur yoziladi va uning xususiyat va usullariga murojaat qilinadi. Misol sifatida joriy vaqtini ko’rsatuvchi HTML xujjatni ko’ramiz.

```
<HTML> <HEAD> <TITLE>Joriy kun va vaqt </TITLE> </HEAD>
<BODY BGCOLOR=WHITE>
<H1> Joriy kun va
vaqt </H1> <SCRIPT
LANGUAGE="JavaScript">
var dt; var
```

```

MyDate="";
dt = new
Date();
MyDate = "Date: " + dt.getDate() + "." + dt.getMonth() + "."
+ dt.getYear(); document.write(MyDate);
document.write("<BR>");
document.write("Time: " + dt.getHours()
+ ":" + dt.getMinutes() + ":" + dt.getSeconds());
</SCRIPT> </BODY></HTML>

```

Bu erda JavaScript ssenariy new kalit so'zi yordamida Data ob'ektini yaratadi. Bunda Date konstruktori parametrlarsiz keltiriladi:

```

var dt;
dt = new Date();
MyDate = "Date: " + dt.getDate() + "."
+ dt.getMonth() + "." + dt.getYear();

```

getDate, getMonth va getYear usullar yordamida joriy sana olinadi. Ushbu usullar dt ob'ekti uchun chaqiriladi.

Matn satri esa HTML xujjatga write usuli yordamida bosmaga chiqariladi. Bu usul document ob'ektining usuli hisoblanadi:

```
document.write(MyDate);
```

Date ob'ekti joriy vaqtini ham o'z ichiga oladi. Bu ma'lumotlar getHours, getMinutes va getSeconds (soat, minut va sekund) usullari yordamida ko'rildi:

```

document.write("Time: " + dt.getHours()
+ ":" + dt.getMinutes() + ":" +
dt.getSeconds());

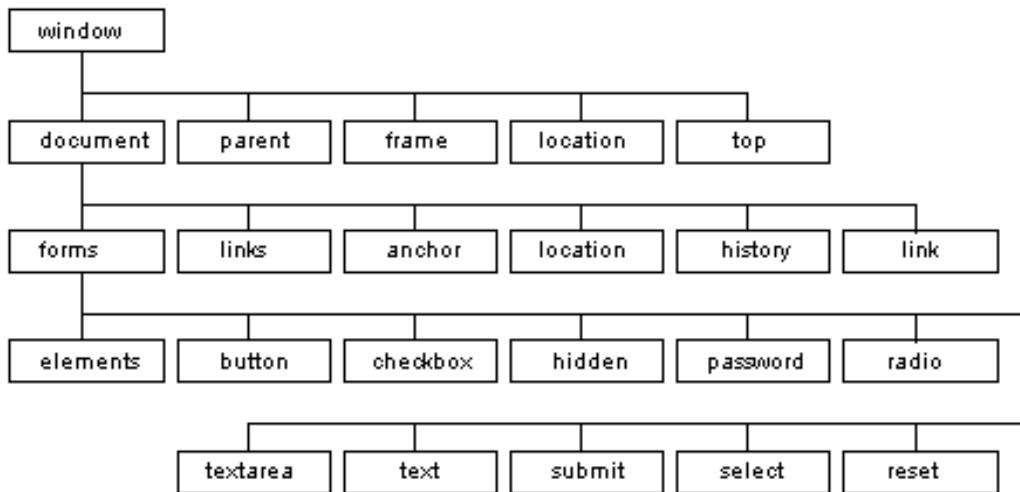
```

Brauzer ob'ektlari

JavaScript ssenariy nuqtai nazari bo'yicha ob'ektlar ierarxik daraxt ko'rinishda tashkil etiladi. Brauzer ob'ektlari foydalanuvchi uchun yaratilgan, brauzer oynasida joylashgan ob'ektlar hisoblanadi. JavaScript stsenarida brauzer ob'ektlari, xususiyat va usullaridan foydalanib bir klass asosida boshqa klass yaratib bo'lmaydi.

Brauzer ob'ektlari ierarxiyasi

Quyidagi rasmda ob'ektlar daraxti ierarxiyasi keltirilgan.



Window ob'ekti bu ierarxiyaning ildizi hisoblanadi. Qachonki HTML xujjat yuklansa unmng ichida
document, parent, frame, location va top boshqa ob'ektlar hosil bo'ladi.

Ob'ektlar bilan bog'liq holatlar

Brauzerning har bir ob'ekti bilan aniq bir holatlar to'plamidan tashkil topadi.

Masalan, window ob'ekti onLoad va onUnload holatlari bilan bog'liq holda ishlaydi.
Birinchi holat

brauzer oynani yuklab bo'lgach ishga tushadi. Ikkinchisi esa brauzer oynani yopish vaqtida ishga tushadi.

Dasturchi tomonidan yaratiluvchi ob'ektlar

Avvalo myRecord nomli klass yaratamiz. Hozircha unda usullar mavjud emas, ularni keyinchalik

qo'shamiz. Bu klass quyidagicha yaratiladi:

```

function myRecord(name, family, phone,
address) { this.name = name;
this.family = family; this.phone =
phone; this.address = address;
this.secure q=false;
}
  
```

Yaratilayotgan ob'ektni xususiyatlarini ko'rsatish uchun maxsus this kalit so'zidan foydalaniladi. Bu kalit so'z ob'ektning xususiyatlariga bo'lgan murojaatini ko'rsatadi.

Keltirilgan klassdan qanday foydalanish mumkin? Yaratilgan klass asosida istalgan sondagi ob'ektlar yaratish mumkin. Quyida berilgan myRecord klassida ikkita rec1 va rec2 ob'ektlari yaratilgan:

```

var rec1; var rec2;
rec1 = new myRecord("Ivan", "Ivanov",
"000-322-223", "A. Temur ko'cha, d. 225, kv.
226"); rec2 = new myRecord("Petr", "Petrov",
  
```

```
"001-223-3334", "Bobur ko'cha, d. 552, kv. 662");  
rec2.secure = true;  
Ob'ektlar new operatori yordamida yaratiladi.
```

Date klassi

Date() metodi argumentsiz berilganda qiymati joriy sana va vaqtga ega Date ob'ekti yaratiladi. Date() metodida yangi ob'ekt uchun argumenti sifatida sana va zarur hollarda vaqt ko'rsatiladi. Date metodi JavaScript tili ob'eksti hisoblanib, HTML tilida hech qanday analogga ega emas. Ko'p hollarda Date ob'ekti metodlari uning ekzempliyati yordamida chaqiriladi, masalan:

```
D=new Date(); // bugungi sana va vaqtini olish  
system.write("Today is: "+d.toLocalString()); // va uni tasvirlash
```

Date ob'ektini yaratishning yuqoridagi sintaktikasida ko'rsatilgani bo'yicha, sana va vaqt hududiy vakt bo'yicha beriladi. Agarda tuzilayotgan dastur foydalanuvchi joylashgan chasovoy poyasga bog'liq bo'lмаган holda ishlashi zarur bo'lsa, u holda Grinvich (GMT) eki universal koordinatsiya vaqt (UTC) bo'yicha sanani ko'rsatish kerak bo'ladi.

Date ob'ektini yaratishda quyidagi 5 ta sintaktik variantdan foydalanish mumkin. 3-5 variantlarda vaqt
hududiy tarzda interpretatsiya qilinadi (Grinvich da emas):

```
new Date();  
new Date(millisecond) – bu erda millisekund joriy sana bilan 01.01.1970 sana yarimkuni orasidagi  
son; new Date(sana satri) – bunda sana satri q oy nomi, dd, gg  
[chch:mm[:ss]] new Date(yil, oy, kun) – bunda, yil 2011; oy  
0-11; kun 1-31; new Date(yil, oy, kun , soat, minut, sekund)  
– 24 soatliliktizimda.
```

Date klassi metodlari

getDate() - Date ob'ektining 1 dan 31 gacha oraliqdagi qiymatini beradi;
getDay() - Date ob'ektining 0 [yakshanba] dan 6 [shanba] gacha oraliqdagi xaftha kunlari beradi; getHours() - Date ob'ektining 0 [yarim tun] dan 23 gacha oraliqdagi soat qiymatini beradi; getMinutes() - Date ob'ektining 0 dan 59 gacha oraliqdagi minut qiymatini beradi; getSeconds() - Date ob'ektining 0 dan 59 gacha oraliqdagi sekund qiymatini beradi; getMonth() - Date ob'ektining 0 [yanvar] dan 11 [dekabr] gacha oraliqdagi oylarni beradi; getTime() - Date ob'ekti vaqt ko'rsatkichining millisekndlardagi qiymatini beradi;
getYear() - Date ob'ekti vaqt ko'rsatkichining yillar maydoni qiymatini beradi; bunda 2011 yil 11
ko'rinishida beriladi; parse() – sananing satr ko'rinishidagi holatini sintaktik analiz qiladi va natijani millisekund formatida
beradi; setDate() - Date ob'ekti vaqt
ko'rsatkichini o'rnatadi; data.setDate(oy soni) //oy soni 1-31 oraliqda.

toLocalString() – joriy hududiy vaqt maydonini asosida Date formatini matnli (String) ko’rinishga keltiradi;
UTC() – sana va vaqtning rahamli ko’rinishini millisekundli formatga aylantiradi.

Nazorat savollari:

1. JavaScript tilining qanaqa funksiyalarini bilasiz? 2. Global klassi va uning operatorlarini sanab o’ting?
3. Math klassi va uning operatorlarini ayting?
4. Date klassi va uning operatorlarini ayting?
5. Qanday turdagи ob'ektlar mavjud?
6. Brauzer ob'yektlari ierarhiyasini chizma yordamida tushuntirib bering?
7. Turli klasslardan foydalanishning ahamiyatli tomonlarini ayтиб o’ting? 8. Turli klasslar orasidagi farqli jihatlarni ko’rsating?

11-ma’ruza: JQUERY ga kirish. Asosiy tushunchalar

Reja:

1. JQuery ga kirish. JQuery sintaksisi.
2. JQuery dan foydalanish usullari.
3. JQuery selektorlari.
4. JQuery hodisalari.
5. JQuery UI.

Birinchidan, jQuery nima ekanligini bilib olaylik?

Shunday qilib, jQuery - bu brauzer orqali kross-brauzer orqali qo'llab-quvvatlanadigan JavaScript kutubxonasi (Internet Explorer 6.0+, Mozilla Firefox 2+, Safari 3.0+, Opera 9.0+ va Chrome-da ishlaydi).

Kutubxonaning muallifi Jon Resig birinchi marta 2006 yil yanvar oyida Nyu-Yorkdagи kompyuter konferentsiyasida o'z ijodini taqdim etdi va o'sha yilning avgust oyida kutubxonaning birinchi barqaror versiyasi chiqdi.

O'tgan yillar davomida kutubxona ko'plab o'zgarishlarni boshidan kechirdi va bugungi kunda vazifalarning eng keng doirasi uchun foydali bo'lgan funktsiyalarni o'z ichiga oladi. U kichik o'lchamga ega (masalan, minimallashtirilgan versiyaning hajmi 55.9 Kbait, siqilgan versiyasi esa 19 Kbait) va unchalik katta bo'limgan identifikatorlar bilan global nomlar maydoniga to'sqinlik qilmaydi.

Selektor mexanizmining ajoyib xususiyatlari, bu hujjat ob'ektlari modelining har qanday elementiga kirishni osonlashtiradi, jQuery-ni juda mashhur qildi.

O'zingiz uchun hukm qiling. Javascript-dan foydalanib DOM elementiga ma'lumot olish uchun getElementByld () usuli ishlatalidi. Masalan, div'l identifikatori bilan elementning HTML kodini o'zgartiring.

```
Document.getElementByld ("divl"). InnerHTML \u003d "Yangi matn";
```

```
Xuddi shu narsani bajaradigan JQuery kodi yarimga teng bo'ladi.
```

```
$("# divl"). html ("Yangi matn");
```

Albatta, yolg'iz ushbu yo'naliш uchun butun kutubxonani ulashning ma'nosi yo'q. Ammo hamma narsa shundan iboratki, selektsionerlarning funktsiyalari bitta identifikator bilan chegaralanib qolmaydi. Selektorlarning kuchini faqat Perldagi oddiy iboralar bilan taqqoslash mumkin. Misol tariqasida, A elementlarining barchasida matn rangini o'zgartiramiz, href parametrda HTML hujjat bilan bog'lanish mavjud va A elementi clsl uslubi klassiga ega bo'lgan DIV element ichida joylashgan bo'lishi kerak.

```
$( "div.clsl a" ). css ("rang", "qizil");
```

JavaScript-dan foydalanib, shunga o'xshash operatsiyani bajaring. Buning uchun bir nechta kod kodlari kerak bo'ladi.

JQuery kutubxonasingning yana bir o'ziga xos xususiyati bu usul qo'ng'iroqlari zanjirlarini amalga oshirish qobiliyatidir, chunki ko'pchilik jQuery usullari keyinchalik boshqarilishi mumkin bo'lgan ob'ektini qaytaradi.

```
$( "# message" ) // id \u003d xabarli elementga havola mavjud  
.html ("Xabar") // Element ichidagi matnni o'zgartirdi  
.parent () // Asosiy elementga havola mavjud  
.css ("background-color", "# fff4dd") // Set rangi fon  
kengligi (300) // Kenglik  
.height (200) // Balandlik
```

// Shaffoflikni o'zgartirib, elementni silliq namoyish eting
.fadeIn(3000);

JQuery kutubxonasi veb-sahifani qayta yuklamasdan server bilan ma'lumot almashish imkonini beruvchi AJAX texnologiyasini e'tiborsiz qoldirmadi. 10-bo'limda biz XMLHttpRequest ob'ektining asosiy xususiyatlari va usullarini ko'rib chiqamiz va 11-bobda jQuery kutubxonasi tomonidan taqdim etilgan AJAX kirish interfeysini ko'rib chiqamiz.

Ko'p qirraliligi sababli, jQuery kutubxonasi deyarli har qanday ishlab chiqaruvchiga foydali bo'ladi. Bu yangi boshlanuvchiga mos keladi, chunki u brauzerda kross-brauzer bilan bog'liq muammoni unutishga imkon beradi (yangi veb-brauzer chiqdi, kutubxona versiyasini o'zgartirdi va hamma yana ishlaydi). Kutubxona professionallar uchun juda mos keladi, chunki bu kodni kamida uch marta qisqartirishga imkon beradi. Va bu, o'z navbatida, sizga minimal kuch va vaqtini yo'qotish bilan juda murakkab kodni yozishga imkon beradi.

Sizga maroqli o'qishni tilayman va ushbu kitob sizning kundalik harakatlariningizda ishonchli sherik bo'lib qoladi va JavaScript dasturlash uslubingizni o'zgartiradi degan umiddaman.

JQuery asoslari

Ushbu kitobda muhokama qilingan barcha fayllar UTF-8 kodlashda saqlanadi. Shuning uchun, jQuery kutubxonasining imkoniyatlarini o'rganishdan oldin, biz UTF-8 boshqa kodlashlardan qanday farq qilishini va ushbu kodlashda faylni qanday qilib to'g'ri saqlashni ko'rib chiqamiz.

Kompyuter ichidagi monitor ekranida ko'rgan barcha belgilar raqam sifatida saqlanadi. Har bir belgi ma'lum bir raqamga mos keladi (belgilarni kodi). Muayyan kod bilan ifodalangan belgi qanday ko'rinishga ega bo'lishi kerakligi haqidagi savolga javob berish uchun kodlash deb nomlangan yozishmalar jadvallari mo'ljallangan. Kodlash bitta va ko'p baytli bo'lishi mumkin.

Bit baytli kodlashda belgi bir bayt bilan kodlangan. Dastlabki 7 bit ASCII kodlashga mos keladigan 128 belgini kodlash imkonini beradi. Ushbu belgilar raqamlar, lotin alifbosidagi harflar, tinish belgilari va ba'zi yordamchi belgililar (masalan, chiziq uzilishlari, yorliqlar

va boshqalar). Ushbu belgilarning kodlari deyarli bitta baytli kodlashlarda bir xil. Sakkizinch bit milliy alifbolarning harflarini kodlash uchun mo'ljallangan. Shunday qilib, bitta baytli kodlash sizga jami 256 belgini kodlash imkonini beradi. Rus tilidagi harflarni kodlash uchun beshta kodlash ishlab chiqildi - windows-1251 (cp1251), cp866, iso8859-5, koi8-r va mac-kirill. Qiyinchilik bu kodlashlarda bir xil rus harfini kodi boshqacha bo'lishi mumkinligidadir. Bu juda ko'p muammolarni keltirib chiqaradi.

UTF-8 kodlashda bitta belgi bir necha baytda kodlanishi mumkin. Dastlabki 128 belgi ASCII kodlangan va faqat bitta baytda kodlangan. Qolgan belgilar o'zgaruvchan miqdordagi baytlar bilan kodlangan - ikkitadan oltitagacha (amalda to'rttagacha).

Rus alifbosi va boshqa ba'zi Evropa tillarining harflari ikki baytda kodlangan. Boshqacha qilib aytganda, UTF-8 kodlashi barcha mavjud alifbolarning belgilarini kodlash imkonini beradi va bir vaqtning o'zida barcha kodirovkalarini almashtirishga qodir. Sayt rus tilida yoki boshqa tilda bo'lishi mumkin va kodlash bir xil bo'ladi. Biz ushbu kodlashni ishlatamiz.

UTF-8 kodlashda fayllarni saqlashda, siz buning uchun Notepad dasturidan foydalana olmasligingizni yodda tutishingiz kerak, chunki saqlash paytida faylning boshida qisqartirilgan BOM (Byte Order Mark, bayt buyurtma belgisi) deb nomlangan xizmat ko'rsatish belgilari qo'yildi. UTF-8 kodlash uchun ushbu belgilar majburiy emas va kelajakda bizga, masalan, server javob sarlavhalarini o'rnatishga imkon bermaydi. UTF-8 kodlash bilan ishlash uchun kompyuteringizga Notepad ++ ni o'rnatish kerak.

Siz dasturni <http://notepad-plus.veb-sahifasidan.mutlaqo.bepul.yuklab.olishingiz.mumkin.sourceforge.net/ru/site.htm>. Ikkala variantdan (ZIP arxivi va o'rnatuvchisi) men sizga

o'rnatuvchini tanlashingizni maslahat beraman, chunki bu holda o'rnatish paytida siz dastur interfeysi tilini tanlashingiz mumkin. Notepad ++ ni o'rnatish tartibi juda sodda va izohlarga muhtoj emas. Yangi hujjat yaratishda Encodings menyusida "UTF-8 kodi (PTO holda)" katagiga belgi qo'ying.

UTF-8 kodlash usulidan foydalanishimiz jQuery kutubxonasi faqat shu kodlash bilan ishlashini anglatmaydi. Loyihalaringizda har qanday boshqa kodlashni ishlatishingiz mumkin. Ammo yodda tutingki, AJAX so'rovlari UTF-8 kodlashda sukut bo'yicha bajariladi. Agar boshqa kodlash usullaridan foydalansangiz, qayta kodlashingiz kerak bo'ladi.

JQuery kutubxonasiga ulanish

JQuery kutubxonasidan foydalanishdan oldin, siz uni <http://jquery.com/> saytidan yuklab olishingiz, uni serveringizga joylashtirishingiz va keyin uni HTML-hujjatga ulashingiz kerak. Ulanish src parametrida kutubxonaga mutlaq yoki nisbiy yo'l ko'rsatilgan skript yorlig'i yordamida amalga oshiriladi.

Skript yorlig'ining o'zi HTML hujjatining HEAD qismida joylashtirilishi kerak.

JQuery kutubxonasini <http://ajax.googleapis.com/> da yuklab olish mumkin. Bunday holda, aloqa quyidagicha bo'ladi.

Agar ilgari jQuery kutubxonasi <http://ajax.googleapis.com/> saytidan yuklangan boshqa saytga tashrif buyurgan bo'lsangiz, veb-brauzer kutubxonani qayta yuklamaydi va keshda saqlangan ma'lumotlardan foydalanadi. Shunday qilib, saytingiz tezligi oshishi mumkin. Bu usulning afzalligi. Ammo, agar <http://ajax.googleapis.com/> sayt mavjud bo'lmasa, unda muammolar paydo bo'lishi mumkin.

Hujjatlar tayyorligini aniqlash

Odatda, aksariyat harakatlar HTML hujjat elementlarida bajariladi. Shuning uchun, skriptlarni faqat hujjat to'liq yuklanganidan keyin ishga tushirishingiz kerak. JQuery kutubxonasi bo'lmasan holda, deraza ob'ektining yuklanish hodisasi odatda shu maqsadda ishlatiladi.

```
windows.onload = function() {
    ogohlantirish();
}
```

Biroq, bu voqeа faqat HTML-hujjatning o'zi va rasmlar kabi barcha boshqa elementlar to'liq yuklanganidan keyin ro'y beradi. Rasmlarni yuklab olish (yoki boshqa domendagi bannerlar) biroz vaqt talab qilishi mumkin, bu vaqt davomida skriptlarning ishlashi to'xtatiladi.

JQuery kutubxonasi bizni vaqtini behuda sarflashdan saqlaydi va hujjat tuzilishi shakllangandan so'ng darhol boshqa elementlarning yuklanishini kutmasdan skriptlarni bajarish imkoniyatini beradi. Siz ushbu hodisani tayyor () usul yordamida boshqarishingiz mumkin.

```
jQuery(hujjat).allaqachon(function() {
});
```

JQuery () funktsiyasi \$ () taxallusiga ega. Ushbu taxallusdan foydalanib, siz hodisani quyidagicha boshqarishingiz mumkin.

```
ogohlantirish("Hujjat skriptlarni bajarish uchun mavjud");
});
```

Agar siz j Query () funktsiyasining parametrlari sifatida funktsiyaga havolani ko'rsatsangiz, u hujjat tuzilmasi shakllantirilgandan so'ng darhol bajariladi.

```
jQuery(function() {
    ogohlantirish("Hujjat skriptlarni bajarish uchun mavjud");
});
```

Ushbu kodni hali ham qisqartirish mumkin.

```
$ (funktsiya () (
ogohlantirish ("Hujjat skriptlarni bajarish uchun mavjud");
});
```

Shuni ham ta'kidlash kerakki, dastur tayyor () usullarga bir nechta qo'ng'iroqlarga ega bo'lishi mumkin. Bunday holda, usullar dastur ichida e'lon qilingan tartibda bajariladi.

jQuery da html bilan ishslash usullari

HTML bilan ishslashning asosiy usullari va bu elementlar uchun usullarini ko'rib chiqamiz. Shunday qilib asosiy usullari:

html(val)-html-kod elementini qo'shadi.

Misol:

```
$("div.sp").html("<span>Salom</span>");
```

Matn "Salom" so`zni span bilan div-sp hamma bo`limiga qo'shadi.

text() / val() -element / qiymati element matn mazmunini qo`shadi.

Misol:

```
$("p").text();
```

Ushbu bandning abzas matnni elementini qo`shadi.

text(val) / val(val) -tovar / qiymati element uchun matnni to`g`irlashni belgilash.

Misol:

```
$ ("P") Matn ("Salom!");
```

Ushbu ko'rsatmalar "Salom" so'zini abzasdan yozish uchun qo`shadi.

HTML-kod:

```
<html>
```

```
<head>
<title>jQuery da html</title>
<link rel="stylesheet" type="text/css" href="style.css">
<script type="text/javascript" src="jquery-1.2.6.js"></script>
<script type="text/javascript" src="script.js"></script>
</head>
<body>
<p id="p1">Matn birinchi abzas!</p>
<p id="p2"></p>
<input type="button" value="Qaytarish!" onclick="addHtml();">
</body>
</html>
```

Funksiya quyidagi ko`rinishda

```
function addHtml(){
var tp1=$("#p1").text();
$("#p2").text(tp1);
}
```

Birinchi o'zgarmaydigan tP1 birinchi xatboshida (birinchi satr) matni mazmunini yozish va keyin ikkinchi xatboshida (ikkinchi chiziq) beringlar chiqarish.

Endi element joylashtirilgan manipulyatsiya usullari bilan tanishib chiqaylik.

append(content) -mavjud mazmuni so'ng barcha tanlangan ob'ektlar ichida so`zni qo'shadi.

Misol:

```
$ ("P") append ("<b> Salom </ b>");
```

Ushbu ko'rsatmalar qalin turi matni "Salom" so'zini oxirida ilovaga qo'shadi.

appendTo(content)-ishni barcha content tanlangan ob'ektlar qo'shiladi.

Misol:

```
$( "p" ).appendTo( "#main" );
```

Bu qo'llanma id "main" elementi oxirigacha qo'shadi.

Boshqa so'zlar bilan aytganda:

```
$ (A) .append (B) - B bilan boshlanib A gacha qo'shish
```

```
$(A).appendTo(B) -A bilan boshlanib B gacha qo'shish
```

prepend(content)-mavjud tarkibga barcha tanlangan ob'ektlar ichida so`zni qo'shadi.

Misol:

```
$( "p" ).prepend( "<b>Salom</b>" );
```

Ushbu ko'rsatmalar qalin turi abzas boshiga matn boshiga "Salom" kiritiladi.

```
prependTo(content)-teskari ishni content barcha tanlangan ob'ektlariga qo'shiladi.
```

Misol:

```
$( "p" ).prependTo( "#main" );
```

Bu qo'llanma id "main" element bilan yuqori bir xat boshiga chiqadi.

```
$(A).prepend(B) - B dan boshlanib A gacha qo'shadi.
```

```
$(A).prependTo(B) -A dan boshlanib B gacha qo'shadi.
```

after(content)-Append bilan bo'lgani kabi element oxiri so'ng tanlangan elementlarni (Eslatma barcha content bo'limiga ketma-ket keyin) qo'shiladi.

Misol:

```
$( "p" ).after( "<b>Salom</b>" );
```

Ushbu ko'rsatma "Salom" qalin turini abzas matni so'ngigacha kiritiladi(salom so`zini takrorlab abzas boshidan chiqaradi).

befor(content)-barcha tanlangan elementlar content bolimiga qo'shadi.

Misol:

```
$( "p" ).befor( "<b>Salom</b>" );
```

Ushbu ko'rsatmalar qalin turi abzastdan "Salom" matni oldin kiritiladi.

```
insertAfter(content) -content-a so'ng barcha tanlangan ob'ektlarni qo'shadi.
```

Misol:

```
$( "p" ).insertAfter( "#main" );
```

Bu qo'llanma id "main" bilan element so'ngigacha xat boshidan chiqadi.

Boshqa so'zlar bilan aytganda:

```
$(A).after(B) -B dan qo'shib A da tugaydi.
```

```
$(A).insertAfter(B) -A dan qo'shib B da tugaydi.
```

insertBefore(content) -content-uchun oldin barcha tanlangan ob'ektlar ketma-ket qo'shadi.

Misol:

```
$( "p" ).insertBefore( "#main" )
```

Ushbu qo'llanma id "main" element bilan oldin birinchi xat boshini chiqaradi.

Boshqa so'zlar bilan aytganda:

```
$(A).before(B) -B dan qo'shib A da tugaydi.
```

```
$(A).insertBefore(B) -A dan qo'shib B da tugaydi.
```

Bir misolni ko'rib chiqaylik. Agar qizil to'rtburchak ichida "Qo'shish" tugmasini bosing sariq kvadratchalar paydo bo'lsa, (- bir quti bir marta bosishda takror) uni yarataylik.

Html kod:

```
<html>
```

```
<head>
```

```
<title>jQuery html</title>
```

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```

<script type="text/javascript" src="jquery-1.2.6.js"></script>
<script type="text/javascript" src="script.js"></script>
</head>
<body>
<div id="red_sk"></div>
<input type="button" value="Qo`shish!" onclick="addYellowSquare();">
</body>
</html>

```

style.css quyidagicha berilgan

```

#red_sk{
width:700px;
height:100px;
background:red;
}
.yellow_square{
float:left;
width:80px;
height:80px;
background:yellow;
margin:5px;
}

```

script.js funktsiya o'zi juda qisqa kod va u quyidagicha berilgan

```

function addYellowSquare(){
$("#red_sk").prepend("<div class='yellow_square'></div>");
}

```

Bu hamma shunday oddiy. Lekin uslublarni qo`shish ko'rib chiqish davom etadi:
wrap(html) -har bir html element uchun element qo`shadi.

Nazorat savollari

1. JQUERY qanday kutubxona hisoblanadi?
2. JQUERYni sahifaga qanday bog'lanadi?
3. JQUERY ning sintaksi qanday ko'rinishda?

12-13-ma’ruza: Angular JS ga kirish. Animatsiya va formalar bilan ishlash

Reja:

1. AngularJS ga kirish.
2. AngularJS da direktivalar.
3. AngularJS da ifodalar.
4. AngularJS da modullar.
5. AngularJS da animasiyalar.
6. Forma elementlari bilan ishlash.

AngularJS o'zi nima? AngularJS Google kompaniyasi tomonidan ishlab chiqilgan JavaScript framework! U bizga yani dasturchilarga sahifani yanada dinamik qilishda yordam beradi, AngularJS eski framework emas u 2009 ishlab chiqilgan lekin 1-versiyasi esa 2012 yil ommaga taqdim etilgan. Shu kunda beri web dasturchilar tomonidan judda keng qo'llanib kelinmoqda va dasturlashdagi eng so'nggi texnologiyalar sirasiga kiradi. Siz oldin jQuery va shu kabi boshqalar bilan ishlab ko'rgan bo'lisingiz mumkin. Ana endi jQueryni vaqtincha unutib AngularJS boshlashningiz kerak.

Angular - bu sifatli JavaScript-klient dasturlarini ishlab chiqish uchun MVW-ramka. U Google tomonidan yaratilgan va qo'llab-quvvatlanadi va kelajakda biz uchun qanday yangi xususiyatlar va standartlar tayyorlanishini taklif qiladi.

MVW Model-View-View (model - ko'rinish - har qanday) degan ma'noni anglatadi, ya'ni dasturlarni ishlab chiqishda dizayn naqshlarini tanlashda moslashuvchanlikni anglatadi. MVC (Model-View-Controller) yoki MVVM (Model-View-ViewModel) modellarini tanlashimiz mumkin.

Ushbu o'quv qo'llanma AngularJS, uning tushunchalari va API-larini o'rganishda boshlang'ich nuqtasi bo'lib, sizga zamonaviy veb-ilovalarni yaratishda yordam beradi.

AngularJS HTMLni takomillashtirish uchun asos sifatida o'zini joylashtirmoqda. U turli xil dasturlash tillarida, ham JavaScript, ham server tomonidan tushunchalarni to'plagan va HTML-dan tashqari dinamik narsalarni yaratmoqda. Ilovalarni ishlab chiqishda ma'lumotlarga asoslangan yondashuvni olamiz. Model, DOM-ni yangilash yoki boshqa vaqt talab qiladigan operatsiyalarni amalga oshirishning hojati yo'q, masalan, brauzer xatolarini tuzatish uchun. Biz ma'lumotlarga e'tibor qaratamiz, ma'lumotlar HTML-ga g'amxo'rlik qiladi va biz faqat dasturni dasturlash bilan shug'ullanamiz.

AngularJS-ning ma'lumotlar va boshqa muhandislik kontseptsiyalari bilan ishlash bo'yicha pozitsiyasi Backbone.js va Ember.js kabi tizimlardan farq qiladi. Biz allaqachon bilgan HTML-dan mammunmiz va Angular uni yaxshilaydi. Angular DOM-ni ma'lumotlar bilan aloqa qilish uchun toza JavaScript ob'ektlarida yashaydigan Modelga har qanday o'zgartirishlar bilan yangilaydi. Model yangilanganida, Angular dasturning joriy holatini o'z ichiga olgan Ob'ektlarni yangilaydi.

Agar siz statik saytlar yaratishga odatlangan bo'lsangiz, sahifaga kerakli ma'lumotlarni kiritganingizda va HTML-ning o'xshash qismlarini qayta-qayta takrorlaganingizda, HTML-ni qo'lida, bo'lak-bo'lak yaratish jarayoni bilan tanishasiz. Bular panjarali ustunlar, navigatsiya uchun tuzilma, havolalar yoki rasmlar ro'yxati va boshqalar bo'lishi mumkin. Kichik bir tafsilot o'zgarganda, siz shablonni va undan keyingi foydalanishni yangilashingiz kerak. Har bir navigatsiya elementi uchun bir xil kod qismlarini nusxalashingiz kerak. Kafedrada turing - Burular vazifalarni ajratish va dinamik HTMLga ega. Bu shuni anglatadiki, bizning ma'lumotlarimiz Modelda, HTML esa Viewga aylantiriladigan kichik shablon ko'rinishida yashaydi va biz ushbu ikkita tushunchani ulash uchun Controller-dan foydalanib, Model va Viewga o'zgartirishlarni qo'llab-quvvatlaymiz. Ya'ni, navigatsiya dinamik ravishda ko'rsatilishi mumkin, bitta ro'yxat elementidan yaratiladi va Modeldan har bir element uchun avtomatik ravishda takrorlanadi. Bu soddalashtirilgan tushunchadir, keyinchalik biz shablon haqida ko'proq gaplashamiz.

MVC va MVVM o'rtasidagi farq shundaki, MVVM interfeyslarni ishlab chiqish uchun maxsus moljallangan. Ko'rinish taqdimot qatlidan iborat, ViewModelda taqdimot mantig'i va Model o'z ichiga biznes mantig'i va ma'lumotlarini oladi. MVVM AngularJS kabi ramkalar rivojlanib boradigan ma'lumotlarning ikki tomonlama aloqasini osonlashtirish uchun yaratilgan. Biz MVVM yo'liga e'tibor qaratamiz, chunki so'nggi yillarda Angular bu erda suyanmoqda.

Avval [AngularJS](http://angularjs.org)ning rasmiy saytiga kiramiz: <http://angularjs.org> kiramiz. Downloadga kirib yuklab olamiz. Atiga bitta angular.js bitta fayli mavjud bo'ladi. Uni HTML sahifamizga ulashimiz kerak bo'ladi.

```
<!DOCTYPE html>
<html ng-app>
<head>
<title>Mening Sahifam</title>
<script type="text/javascript" src="angular.js"></script>
</head>
<body>

</body>
</html>
```

mana bunday AngularJS ulanadi lekin e'tibor bering <html ng-app> yozilib AngularJS ishga tushurilmoqda ya'ni angular.js o'zini ulab qo'yish yetmaydi uni ishga tushurish uchun html tegining ichida ng-app deb yozish kerak bo'ladi.

Demak AngularJS ish uchun tayyor bo'ldi deb o'layman. AngularJSda ishlataladigan barcha maxsus html teglar va attributlar direktivlar deyiladi. Direktivlar ikkita figurali qavslar orasiga yoziladi {{bu yerda direktiv yoziladi}}

Direktivlar ishlashini ko'rish uchun

```
<!DOCTYPE html>
<html ng-app>
<head>
<title>Mening Sahifam</title>
```

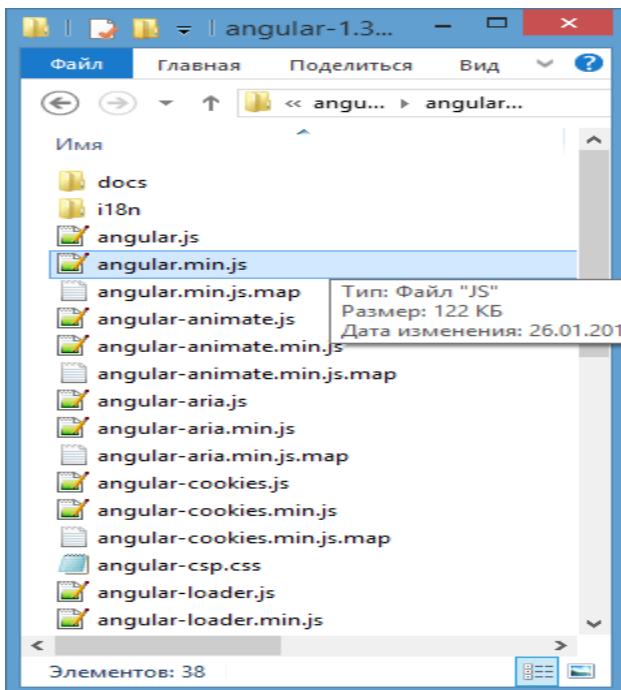
```
<script type="text/javascript" src="angular.js"></script>
</head>
<body>
{{5+2}}
</body>
</html>
```

saqlab natijani ko'rsak sahifada 7 paydo boladi. Ya'ni direktivlarni ichiga be'malol kod yozsak bo'ladi hatto direktivkarni ichida funksiya yo'zsak bo'ladi lekin bu sal xato yo'l bo'ladi.

Yana bir misolni ko'ramiz:

```
<!DOCTYPE html>
<html>
<head>
<title>Mening Sahifam</title>
<script type="text/javascript" src="angular.js"></script>
</head>
<body>
<input type="text" ng-model="ism" />
{{ism}}
</body>
</html>
```

ko'rib turganingiz dek oddiy html forma mavjud lekin unga ma'lumot yozganingizda haqiqiy mo'jiza yuz beradi :) Yozgan ma'lumotlarimiz sahifada ko'rinishni boshlaydi. Bu yerda formaga ng-model attributi orqali ism modeli ko'rsatdik keyin esa pastroqda direktiv yordamida ekranga chiqardik. Agar shuni toza JavaScript o'ziningiz yozmoqchi bo'lganingizda bir necha qator yoki o'nlab qator kod yozishingizga to'g'ri kelar edi. AngularJS esa bizga bu borada yordam borib yuqori darajadagi sahifa yaratishda yordam beradi.



- **angular-touch.js** : sensorli ekran hodisalarini qo'llab-quvvatlaydi
- **burchak-animate.js** : animatsiya funktsiyasini ta'minlaydi
- **angular-aria.js** : aria-atributlarni qo'llab-quvvatlaydi (boy Internet-ilova)
- **angular-mocks.js** : birlklarni sinash uchun masxaralash ob'ektlarini taqdim etadi
- **angular-train.js** : marshrutlash mexanizmi mavjud
- **angular-sanitize.js** : potentsial xavfli tarkibni boshqarish uchun funktsiyani ta'minlaydi (javascript, html)
- **angular-cookies.js** : cookie fayllarini boshqarish uchun funktsional imkoniyatlarni ta'minlaydi
- **angular-loader.js** : angularjs skriptlarini yuklash uchun ishlatiladi
- **angular-messages.js** : xabarlarni namoyish qilish uchun funktsional imkoniyatlarni ta'minlaydi
- **angular-resource.js** : manbalar bilan ishlash uchun funktsional imkoniyatlarni ta'minlaydi
- Papka **I18N** : turli joylarda uchun js-fayllarni o'z ichiga olgan
- Yagona sahifali dastur (SPA), davlat boshqaruvi va Ajax (HTTP)

Yagona sahifali dasturda (SPA) yoki barcha kerakli kodlar (HTML, CSS va JavaScript) bitta sahifa yuklanishida chaqiriladi yoki kerakli resurslar dinamik ravishda ulanadi va kerak bo'lganda sahifaga qo'shiladi, odatda foydalanuvchi harakatlariga javoban. Sahifa ish paytida qayta yuklanmaydi, boshqaruvni boshqa sahifaga o'tkazmaydi, garchi HTML5 dan zamonaviy texnologiyalar bitta dasturga bir nechta mantiqiy sahifalarda ishlashga imkon beradi. SPA bilan o'zaro aloqa ko'pincha server bilan fon aloqasi orqali ro'y beradi.
- Eski dasturlarda, dasturning holati serverda saqlanganida, foydalanuvchi ko'rgan narsalar va serverda saqlangan narsalar o'rtasida farqlar mavjud edi. Modeldag'i dasturlar holati etishmasligi ham mavjud edi, chunki barcha ma'lumotlar HTML shablonlarida saqlangan va dinamik emas edi. Server statik shablonni tayyorladi, foydalanuvchi u erda ma'lumotni kiritdi va brauzer uni qaytarib yubordi, shundan so'ng sahifa qayta yuklandi va serverning holati yangilandi. Saqlanmagan har qanday holat yo'qoldi va brauzer sahifalarni yangilaganidan keyin barcha ma'lumotlarni yuklab olishlari kerak edi.
- Vaqtlar o'zgardi, brauzer dastur holatini saqlaydi, murakkab mantiq va ramkalar mashhurlikka erishdi. AngularJS holatni brauzerde saqlaydi va agar kerak bo'lsa, GET, POST, PUT va DELETE usullaridan foydalanib, o'zgarishlarni Ajax (HTTP) orqali uzatadi. Go'zallik shundaki, server old qismdan mustaqil bo'lishi mumkin, va old qismi serverdan mustaqil bo'lishi mumkin. Xuddi shu serverlar butunlay boshqa old tomoniga ega mobil ilovalar bilan ishlashlari mumkin. Bu bizga moslashuvchanlikni beradi, chunki biz JSON ma'lumotlari bilan har qanday server tilida biz uchun qulay bo'lgan har qanday usulda ishlaymiz.

- Burchaklar turli xil API-larga ega, ammo dasturning tuzilishi odatda bir xil, shuning uchun deyarli barcha dasturlar shunga o'xshash tarzda qurilgan va ishlab chiquvchilar loyihaga bemalol qo'shilishlari mumkin. Shuningdek, u oldindan taxmin qilinadigan API va disk raskadrovska jarayonlarini ta'minlaydi.

1.

AngularJSda direktivlar.

Salom. AngularJSda direktivalar bilan tanishishni davom etamiz.

ng-change - bu direktiv orqali modelning qiymati o'zgartirilgandan ng-change ishga tushadi. ng-change ichida funksiya yoki biron-bir ifodani ko'rsatish mumkin. Masalan:

```
<input type="text" ng-model="modelName" ng-change="ifoda()" />
```

Agar formaga ma'lumot kiritilsa yani modelName yangi qiymat olsa, ifoda() funksiyasi ishga tushadi.

ng-checked - checkboxni modeliga qarab bosilgan yoki bosilmagani tekshiradi.

```
<label>Model orqali checkbox <input type="checkbox" ng-model="master"></label> <input id="checkSlave" type="checkbox" ng-checked="master" aria-label="Slave input">
```

ng-class - html tegning css klassini o'zgartirish uchun model orqali qo'llaniladi. Masalan:

```
<input id="setbtn" type="button" value="set" ng-click="myVar='my-class'"> <input id="clearbtn" type="button" value="clear" ng-click="myVar=''"> <span class="base-class" ng-class="myVar">M atn</span>
```

Bu yerda biz ikkita tugma yasadik. Birinchi tugma matnimizga "my-class" css klassini qo'shish uchun, ikkinchi tugma orqali esa matndagi csss klassni olib tashlash uchun.

ng-copy - forma ichidagi ma'lumot buffer hotiraga ko'chirilgan yoki ko'chirilmaganini aniqlaydi.

```
<input ng-copy="copied=true" ng-init="copied=false; value='copy me'" ng-model="value"> copied : {{copied}}
```

ng-cut - forma ichidagi ma'lumot kesilb olinga(cut)ni aniqlaydi

```
<input ng-cut="cut=true" ng-init="cut=false; value='cut me'" ng-model="value"> cut: {{cut}}
```

ng-dblclick - sichqoncha orqali ikki marta bosilganda bajariladi.

```
<button ng-dblclick="count = count + 1" ng-init="count=0"> Increment (on double click) </button>
```

14-15-ma’ruza. Bootstrap freymvork texnologiyasi. Bazaviy shablonlarni sozlash. Komponentlar.

Reja:

1. Bootstrap freymvork texnologiyasi.
2. Freymvorkni ularash, bazaviy shablonlarni sozlash.
3. Bootstrap freymvorkda tugmalar, ro‘yxatlar, navigasiya menyulari, formalar bilan ishslash.
4. Bootstrap komponentlari

Bootstrap – web sahifalarning (UI) interfeysi yaratish uchun mo’ljallangan ochiq kodli FrontEnd Freymvork.

Asoschilari: *Mark Otto va Yakob Tornton*.

Ishga tushirilgan sana: 2011-yil, Avgust.

Yaratilgan joy: Twitter



Bootstrap

- * Bootstrap qulay va qisqa fursat ichida web sahifa va web ilovaning frontend qismini yaratishga yo’naltirilgan freymvork
- * Bootstrap HTML va CSS yordamida yaratiladigan bazaviy shablonlarni o’z ichiha oladi.
- * Bootstrap responsive web dizayn yaratishga imkon beruvchi texnologiyadir.

Responsiv web dizayn – yaratilgan web ilova, web sahifalarni ekrani turli xil o’lchamli qurilmalarda avtomatik tarzda moslashishini ta’minlaydi.

Freymvorkni sahifada qo’llashni 2 xil usulda amalga oshirish mumkin:

1. Freymvorkni getbootstrap.com saytidan yuklab olish
 2. Bootstrap CDN (Content Delivery Network) orqali. (MaxCDN)
 3. <!-- CSS-->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
- ```
<!-- jQuery kutubxonasi -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```

<!-- JavaScript-->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<!DOCTYPE html>
<html lang="en">
<head>
 <title>Bootstrap Example</title>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
 <h1>My First Bootstrap Page</h1>
 <p>This is some text.</p>
</div>

</body>
</html>

```

- \* Bootstrapning Grid tizimi responsive dizayn yordamida sahifani jami 12 ta ustundan iborat qismlarga bo'lish uchun tayanch vazifasini o'taydi.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1			
span 4				span 4				span 4							
span 4				span 8											
span 6						span 6									
span 12															

- \* xs – ekran kengligi o'lchami 768px dan kichik mobil telefonlar uchun
- \* sm - ekran kengligi o'lchami 768px ga teng yoki kattaroq tabletlar uchun
- \* md - ekran kengligi o'lchami 992px ga teng yoki kattaroq laptoplar uchun
- \* lg - ekran kengligi o'lchami 1200px ga teng yoki katta kompyuterlar uchun
- \* <div class="row">
 <div class="col-sm-4">.col-sm-4</div>
 <div class="col-sm-4">.col-sm-4</div>
 <div class="col-sm-4">.col-sm-4</div>
 </div>

## Jadvallar/Tables

.table – oddiy dizaynga ega jadval

.table-striped – qatorlari zebrasimon dizaynga ega jadval

.table-bordered – chetki chiziqlar mavjud jadval

.table-hover – qatorlariga hover effekti berilgan jadval

.table-condensed – qator chiziqlari yarmi mavjud shakldagi jadval

.table-responsive – turli xil o'lchamdagisi ekranga ega qurilmalar uchun avtomatik tarzda moslashadigan jadval

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

## Rasmlar/Images

.img-rounded – cheti ovalsimon shakldagi rasm

.img-circle – aylana shakldagi rasm

.img-thumbnails – cheti ramka bilan o'raglan rasm

.img-responsive – Responsiv dizayndagi rasm

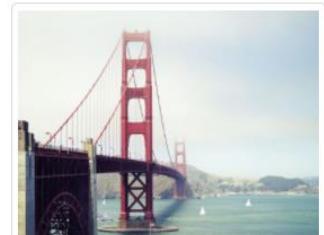
Rounded Corners:



Circle:



Thumbnail:



- Maxsus tarkibli e'tiborga molik ma'lumotlarni chiqazishga mo'ljalangan katta quti.
- Jumbotron kulrang, chetki burchaklari yumaloqlashgan shaklda bo'ladi.
- Shuningdek jumbotron ichidagi matn o'lchamlari maxsus kattalashgan ko'rinishda bo'ladi.
- Jumbotron ichida har qanday HTML hamda Bootstrap elementlarini, joylashtirish mumkin.
- Jumbotron hosil qilish uchun <div> elementidan foydalanish tavsiya etiladi.

.well – chetki burchaklari biroz aylanasiomon, orqa fon rangi kulrang rangdagi, ichki oraligi (padding) biroz kengaytirilgan element.

.well o'lchamiga ko'ra:

Kichik, O'rtacha va Katta (Small, Normal and Large) bo'lishi mumkin.

<div class="well well-sm">Small Well</div>

```
<div class="well">Normal Well</div>
<div class="well well-lg">Large Well</div>
```

## Alert

Bootstrap da oldindan belgilangan ogohlantirish xabarlarini hosil qilishning oson usuli mavjud. Bunday xabarlar alert sinfi orqali hosil qilinadi. Xabarlarning quyidagi turlari mavjud:

alert-success: muvaffaqiyatli pozitiv xabar

alert-info: neytral malumotli xabar

alert-warning: nimadandir ogohlik beruvchi xabar

alert-danger: muvaffaqiyatsiz, negative xabar

## Button Styles

.btn

.btn-default

.btn-primary

.btn-success

.btn-info

.btn-warning

.btn-danger

.btn-link



## Button Groups

btn-group: Bir nechta buttonlarni birgalikda bitta guruhga birlashtirib gorizontal yoki vertikal liniya bo'yicha joylashtirish uchun ishlataladi.

```
<div class="btn-group">
 <button type="button" class="btn btn-primary">Apple</button>
 <button type="button" class="btn btn-primary">Samsung</button>
 <button type="button" class="btn btn-primary">Sony</button>
</div>
```

Glypaticon lar glyphicon klassi orqali hosil qilinadi

Har bir ikonkaning o'z nomi mavjud.

```

 Chop etish 
 Izlash 
 Konvert 
 Yuklash 
 Jo'natish 

```

**.pagination** joriy sahifadan boshqa sahifalarga o'tishni amalga oshiruvchi komponentlarni o'z ichiga oladi.

```

<ul class="pagination">
 1
 <li class="active">2
 3
 4
 5


```

1	2	3	4	5
---	---	---	---	---

Navigatsion menuy yaratish uchun .navbar klassidan foydalilanadi.

```

<nav class="navbar navbar-default">
 <div class="container-fluid">
 <div class="navbar-header">
 WebSiteName
 </div>
 <ul class="nav navbar-nav">
 <li class="active">Home
 Page 1
 Page 2
 Page 3

 </div>
</nav>

```



## **16-ma’ruza. PHP asoslari. Ma’lumotlar turlari**

**Reja:**

1. PHP ga kirish. PHP asoslari, sintaksisi.
2. PHP versiyalari.
3. O‘zgaruvchilar, konstantalar, ma’lumotlar turlari.
4. Operatorlar.

**Kalit so’zlar:** PHP dasturlash tili, izohlar, o’zgaruvchilar, konstantalar, tashqi o’zgaruvchilar, server tomonda dasturlash, denver paketi, Denver Distributivi.

### **1. PHP dasturlash tiliga kirish**

Hozirgi kunda internet keng ommalashgani sababli, zamon taraqqiyotini web-texnologiyasiz tassavvur etish mumkin emas. Web texnologiyalariga talab oshgan sari Web-dasturlash tillarini bilish har bir dasturchi uchun muhim vazifa sanalmoqda. Shularni inobatga olgan holda zamonaviy web-dasturlash tillaridan biri hisoblangan, sodda, o’rganishga qulay, barcha ma’lumotlar bazasi bilan ishlay oladigan PHP haqida batavsilroq to’xtalishga niyat qildik. Kelgusida bu til o’zbek tilida yoritilib boriladi hamda mutaxassis va o’rganuvchilar uchun forum tashkil qilinadi.

**PHP tarixi.** Ko’pgina boshqa dasturlash tillaridan farqli ravishda, PHP qandaydir tashkilot yoki kuchli dasturchi tomonidan yaratilgan emas. Uni oddiy foydalanuvchi Rasmus Lerdorf 1994 yili o’zining bosh sahifasini interaktiv uslubda ko’rsatish uchun yaratgan. Unga Personal Home Page (PHP – shaxsiy bosh sahifa) deb nom bergan.

1995 yili Rasmus PHPni o’zining HTML formalari bilan ishlaydigan boshqa dastur bilan umumlashtirib PHP/FI Version 2 (“Form Interpretator”) hosil qildi. 1997 yilga borib PHP dan foydalanuvchi saytlar 50 mingdan oshdi. Shundan so’ng web texnologiya ustalari PHP g’oyasi asosida mukammal til yaratishga Ziva Suraski va Endi Gutmans asoschiligidagi kirishildi. PHPni samarali deb hisoblanmagani uchun deyarli noldan boshlab, mavjud C va Perl tillaridan ibrat olib PHP3 talqinini yaratildi. 1999 yilga kelib PHP asosida qurilgan saytlar milliondan oshib ketdi. 2000 yilda esa Zend Technologies shirkati yangi ko’pgina funksiyalarni qo’shgan holda PHP4 sharhlovchisini yaratdi.

PHP – web texnologiya tili. PHPni o’rganish uchun avval HTML va dasturlash tilidan habardor bo’lish talab qilinadi. HTML/CSS va JavaScript larni mukammal bilguvchilar uchun PHPni o’rganish murakkablik tug’dirmaydi. PHPning vazifasi HTML faylini yaratib berish. JavaScript yordamida bajariladigan ko’pgina operatsiyalarni PHP orqali ham amalga oshirish mumkin, ammo e’tibor qilish lozimki, PHP – serverda; JavaScript – mijoz tomonda bajariladi. PHPda yozilgan kod serverning o’zida bajarilib, mijozga HTML shaklida etib boradi. Bu havfsizlik jihatdan ancha maqsadga muvofiq. JavaScript yordamida kod yozish, ma’lumot uzatish va qabul qilishni biroz tezlashtirsa-da, kodni mijoz ko’rish imkoniyatiga ega bo’ladi. Baribir har ikkisini boshqasi bosa olmaydigan o’z o’rni bor, ravshanki bu o’rin PHPda muhimroq va kattaroq.

### **2. PHP TILI ASOSLARI PHP DASTURLARI**

PHP dasturlari ikki usulda bajarilishi mumkin: Web-server tomonidan stsenariy ilovasi va konsol dasturi sifatida.

Bizning maqsadimiz web ilovalarni dasturlash bo‘lgani uchun asosan birinchi usulni ko‘ramiz.

PHP odatda Internet bilan bog‘liq dasturlar yaratish uchun ishlatiladi. Lekin PHP dan komanda satrlar interpretatori, asosan \*nix tizimlarda foydalanish mumkin. Oxirgisi CORBA va COM interfeyslar hamda PHP-GTK kengaytmasi yordamida mumkin. Bu xolda quyidagi masalalarni echish mumkin:

- Interaktiv komanda qatorlari yordamida ilovalar yaratish;
- Kross-platformali GUI ilovalarni PHP-GTK bibliotekasi yordamida yaratish;
- Windows va Linux uchun ba’zi masalalarni avtomatizatsiya qilish

Serverga brouzerning murojjat qilishida yordamida php-stsenariylari bajarilishini ko‘rib chtqamiz. Avval borouzer .php kengaytmali sahifani so‘raydi, so‘ngra web-server dasturni PHP mashinadan o‘tkazadi va natijani html-kod shaklida qaytaradi. Agar standart HTML sahifani olib, kengaytmasini .php ga o‘zgartirilsa va PHP mashinadan o‘tkazilsa, foydalanuvchiga o‘zgartirmasdan qaytaradi. Buf faylga PHP komandani qo‘sish uchun, PHP komandalanini maxsus teglar ichiga olish kerak. Bu teqlarning 4 xil shakli mavjud bo‘lib, ixтиyoriysidan foydalanish mumkin:

1. XML qayta ishlash instruktsiyasi:  
2. <?php  
3. ...  
?>
4. SGML qayta ishlash instruktsiyasi:  
5. <?  
6. ...  
?>
7. HTML stsenariylari qayta ishlash instruktsiyasi:  
8. <script language = "php">  
9. ...  
10. </script>
11. ASP uslubidagi instruktsiya:  
12. <%  
13. ...  
14. %>

Biz XML yoki SGML uslubiga rioya qilamiz.

Xususan biror blok ichida PHPdan chiqish mumkin, faqat keyinchalik yana uning ichiga kirib kodni tugatish sharti bilan, quyidagi konstruktsiya mumkin:

<?

```

if(5<3){
echo("<p>Hello,
world!<p>");
?>
<p>Hello!</p>
// bu qator PHP kodi sifatida qaralmaydi
// va kod bloki bajarilayotgan bo'lsa
chiqariladi <? echo("<p>Hello,
world!<p>");
}
?>

```

PHP da echo komandasi web – sahifalarda uchraydigan har qanday ma'lumotni( matn, HTML ajratuvchi simvoli, son) chiqarish uchun qo'llanadi. Uning ma'nosi misolda lo'rsatilgan.

### ***IZOXLAR***

PHP tilida izoxlarni joylash uchun bir necha usullar mavjud. Eng soddasi ikkilik slesh (//) dan foydalanish, shundan so'ng PHP satrlar oxirigacha yozilganni o'tkazib yuboradi. Bundan tashqari S /\*...\*/ uslubidagi ko'p qatorli izoxlardan foydalanish mumkin. Bir qatorli izoxlar uchun (#) simvoldan foydalanish qulay.( UNIX script tillaridagi izox).

```

<php
echo("<p>Hello</p>"); //
izox
echo("<p>Hello</p>"); #
izox
/*
bu ham izox
*/
?
>

```

Shuni esdan chiqarmaslik lozimki PHP uslubi izoxlari faqat PHP chegaranishlari orasida ta'sir qiladi. Agar PHP bu izoxlar simvollarini chegaranishlari tashqarisida uchratsa, ularni boshqa matnga o'xshab, html- sahifaga joylashtiradi.

Masalan:

```

<php echo("<p>Hello</p>"); //
normal izox
?>

```

```
// bu izox brouzerda ko'rinadi.
<!-- HTML izoxi.

Bu izox HTML kodda ko'rinadi, brouzerda emas -->
Izoxlarni faqat operator oxiriga emas, quyidagicha joylash ham mumkin:
<?

 $a = "Hello, world";
echo strstr($a,"H");
 // bu funktsiyani keyinchalik qarab chiqamiz
?>
```

**O'ZGARUVCHILAR VA KONSTANTALAR** PHP da o'zgaruvchilar dollar (\$) belgisidan boshlanadi. Bu simvoldan ixtiyoriy sondagi harf, raqam va ostiga chizio' simvollari kelishi mumkin, lekin birinchi simvol albatta harf bo'lishi kerak. Shuni esda tutish kerakki, PHPda o'zgaruvchilarning nomlari kalit so'zlardan farqli registrga bog'liqdir.

PHP da o'zgaruvchilarni ta'riflaganda oshkora tipini ko'rsatish shart emas va dastur davomida itta o'zgaruvchi har xil tiplarga ega bo'lishi mumkin.

O'zgaruvchi unga qiymat berilganda initsializatsiya qilinadi va dastur bajarilguncha mavjud bo'ladi.

Ya'ni web-sahifa xolida to so'rov tugamaguncha.

#### **TASHQI O'ZGARUVCHILAR**

Klient so'rovi veb-server tomonidan taxlil qilinib, PHP mashinaga uzatilgandan so'ng, u so'rovga tegishli ma'lumotlarni o'z ichiga olgan va bajarish davomida murojaat qilish mumkin bo'lgan bir necha o'zgaruvchilarni yaratadi. Oldin PHP sizni tizimingiz atrof muxit o'zgaruvchilarini oladi vash u nomdagi va shu qiymatdagi PHP stsenariysi atrofidagi o'zgaruvchilarni yaratadi, toki servedagi stsenariylarga klient tizimi xususiyatlari bilan ishslash mumkin bo'lsin. Bu o'zgaruvchilar **\$HTTP\_ENV\_VARS** assotsiativ massivga joylashtiriladi.

Tabiiyki **\$HTTP\_ENV\_VARS** massivi o'zgaruvchilari tizimga bog'liqdir (chunki ular aslida atrof muxit o'zgaruvchilaridir). Atrof muxito'zgaruvchilari qiymatlarini sizni mashinangiz uchun env (Unix) yoki set (Windows) komandasini yordamida ko'rishingiz mumkin. So'ngra PHP u GET-o'zgaruvchilarning guruxini yaratadi. Ular so'rov satrini taxlil qilishda yaratiladi. So'rov satrini **\$QUERY\_STRING** o'zgaruvchida saqlanadi va so'ralgan URL dagi "?" simvoldan keyingi informatsiyadan iborat. PHP so'rov satrini & simvollari bo'yicha alovida elementlarga ajratadi, va har bir elementda "=" belgisini qidiradi. Agar "=" belgisi topilgan bo'lsa, tenglik chap tomonidagi simvollardan iborat o'zgaruvchi yaratadi. Quyidagi formani ko'ramiz:

```
<form action = "http://localhost/PHP/test.php" method="get">
```

HDD: <input type="text" name="HDD"/><br>

CDROM: <input type="text" name="CDROM"/><br>

```
<input type="submit"/>
```

Agar siz bu formada HDD qatorda "Maxtor", CDROM qatorda "Nec" tersangiz, quyidagi so'rov shaklini xosil qiladi:

<http://localhost/PHP/test.php?HDD=Maxtor&CDROM=Nec>

Bizning misolimizda PHP quyidagi o‘zgaruvchilarni yaratadi: **\$HDD** = "Maxtor" va **\$CDROM** = "Nec".

Siz o‘zingizni scriptingizdagi (bizda – test.php) bu o‘zgaruvchilar Bilan oddiy o‘zgaruvchilar bilan ishlagandek ishlastingiz mumkin. Bizning misolimizda ular ekranga chiqariladi:

```
<? echo("<p>HDD is
$HDD</p>"); echo("<p>CDROM
is $CDROM</p>"); ?>
```

Agar sahifa so‘rovi POST usuli yordamida bajarilsa, POST-o‘zgaruvchilarning guruxi yaratilib, interpretatsiya qilinadi va **\$HTTP\_POST\_VARS** massivga joylashtiriladi.

### **KONSTANTALAR**

Konstantalar PHP da **define()** funktsiyasi yordamida e’lon qilinadi:

```
define(CONSTANT, value)
```

Bu funktsiya birinchi parametri – konstant nomi, ikkinchisi – uning qiymati. Konstantadan foydalanylганда nomi bo‘yicha ilova qilinadi:

```
<? define(CONSTANT1,15);
define(CONSTANT2,"\\x20"); // kod
probela define(CONSTANT3,"Hello");
echo(CONSTANT1); echo(CONSTANT2);
echo(CONSTANT3);
?>
```

Odatga ko‘ra konstantalar nomlari yuqori registr harflari bilan yoziladi. Bu faqat odat bo‘lsa ham unga rioya qilishni maslaxat beramiz, chunki yaxshi odatlarga rioya qilmaydigan dasturchilardan yomon dasturchilar chiqadi. Konstantalar aniqlanganligini **defined()** funktsiyasi yordamida tekshirish mumkin:

```
<?
 define(CONSTANT,"Hello");
if(defined("CONSTANT"))
 { echo("<p>CONSTANT is
defined</p>");
 }
?>
```

### **3. PHP imkoniyatlari**

«PHP da har qanday dastur bajarsa bo’ladi», – degan edi uning yaratuvchisi. Birinchi navbatda PHP tili server tomonidan bajariladigan skriptlar yaratish uchun foydalilanadi va aynan shuning uchun u yaratilgan. PHP tili ixtiyoriy CGI-skriptlari masalalarini echishga va bundan tashqari html formali ma'lumotlarni qayta ishslashga hamda dinamik ravishda html sahifalarini ishlab

chiqishga qodir. Biroq PHP tili foydalilaniladigan boshqa sohalar ham mavjud. Bu sohalarni biz uchta asosiy qismga bo'lamiz:

Birinchi soha – biz yuqorida aytib o'tganimizdek, server tomonidan bajariladigan ilovalar (skriptlar) yaratish. PHP tili bunday turdag'i skriptlarni yaratish uchun juda keng qo'llaniladi. Bunday ish ko'rsatish uchun PHP-parser (ya'ni php-skriptlarni qayta ishlovchi) va skriptlarni qayta ishlovchi web-server, skriptlarni natijasini ko'rish uchun brauzer va albatta php-kodini yozish uchun qanday bo'lsa ham matn muharriri kerak bo'ladi. PHP-parser CGI-dasturlar ko'rinishida yoki server modullari ko'rinishida tarqalgan.

Uni va web-serverni kompyuterimizga qanday o'rnatamiz, biz bu haqida keyinroq ko'rib o'tamiz.

Ikkinci soha – buyruqlar satrida bajariladigan skriptlarni yaratish. Ya'ni PHP tili yordamida biror-bir kompyuterda brauzer va web-serverlardan mustahil ravishda o'zi bajariladigan skriptlarni ham yaratish mumkin. Bu ishlarni bajarish uchun hech bo'limganda PHP-parser (bu holatda biz uni buyruqlar satri interpretatori (CLI, command line interpreter) deb ataymiz) talab etiladi. Bunday ishslash uslubi turli masalalarni rejalshtirish yordamida bajarilishi uchun kerak bo'lgan skriptlar yoki oddiy matnni qayta ishslash uchun kerak bo'lgan masalaga o'xshash ishlaydi.

Va nihoyat oxirgi uchinchi soha – bu mijoz tomonidan bajariladigan GUI-ilovalarni (grafik interfeys) yaratish. Bu soha PHP tilini endigina o'rganayotgan foydalanuvchilar uchun uncha muhim bo'lмаган sohadir. Biroqagarda siz PHP tilini chuqur o'rgangan bo'lsangiz, bu soha siz uchun ancha muhimdir. PHP tilini bu sohaga qo'llash uchun php kengaytmali maxsus yordamchi – PHP-GTK talab etiladi.

Shunday qilib, PHP tilini qo'llanilish sohalari keng va turlichadir. Yuqoridagi masalalarni echa oladigan boshqa turlicha dasturlash tillari ham mavjud, unda nima uchun PHP tilini o'rganishimiz kerak? U til bizga nima beradi? Birinchidan, PHP tili o'rganish uchun juda qulay. PHP tilini sintaksisi asosiy qoidalari va ishslash printsipi bilan yetarlicha tanishib chiqib o'zingizni shaxsiy dasturingizni tuzib ko'rib, so'ngra uni boshqa dasturlash tillarida tuzilgan variantlari bilan solishtirsangiz bunga guvohi bo'lasiz.

Ikkinchidan, PHP tili barcha bizga ma'lum platformalarda, barcha operatsion tizimlarda hamda turlicha serverlarda erkin ishlay oladi. Bu xususiyat juda muhim. Masalan, kimdir Windows operatsion tizimdan Linux operatsion tizimga yoki IIS serverdan Apache serverga o'tmoqchi bo'lsa PHP tilini o'rganishi shart.

PHP dasturlash tilida dasturlashning ikkita hammabop paradigmalari ishlatiladi, bular protsedurali va ob'ektli dasturlash. PHP4 dasturlash tili protsedurali dasturlashni butunlay qo'llab quvvatlaydi, Biroq ob'ektli stildagi dasturlarni ham qo'llasa bo'ladi. PHP5 dasturlash tilining birinchi testlash versiyasida PHP4 dasturlash tilida uchraydigan ob'ektga yo'naltirilgan dasturlash modellarining kamchiliklari to'ldirilgan. Shunday qilib, Hozirda tanish bo'lib ulgurgan ishslash printsipini tanlash kerak.

Agarda PHP tilini Hozirgi imkoniyatlari to'g'risida gaplashadigan bo'lsak, u holda biz PHP tilini birinchi versiyasidan ancha yiroqlashib ketgan bo'lamiz. PHP dasturlash tili yordamida tasvirlar, PDFfayllar, flesh-roliklar yaratish mumkin; Hozirgi vaqtdagi zamonaviy ma'lumotlar bazasini qo'llab quvvatlaydi; ixtiyorli matnli fayl formatlari bilan, hamda XML va fayllar tizimi bilan ishlaydigan funksiyalar ham qo'shilgan. PHP tili turli servislardan o'rtafigi protokollarning o'zaro aloqasini qo'llab quvvatlaydi. Bularga misol papkalarga kirishni boshqarish protokoli LDAP, tarmoq qurilmalari bilan ishlaydigan protokol SNMP, ma'lumotlarni uzatish protokollari IMAP, NNTP hamda POP3, gipermatnlarni uzatish protokoli HTTP va boshqalarni olish mumkin.

PHP dasturlash tilini turli dasturlash tillari o'rtasidagi o'zaro aloqasiga diqqatni qaratsak, bunga Java dasturlash tilini aytib o'tish kerakki, Java dasturlash tili ob'ektlarini PHP tili o'z ob'ektlari sifatida qaraydi. Ob'ektlarga murojaat sifatida CORBA kengaytmasidan foydalaniladi.

Matnli axborotlar bilan ishlash uchun PHP tili o'ziga Perl dasturlash tilidagi tartiblangan ifodalar bilan ishlay oladigan mexanizmlarni (katta bo'lмаган о'згаришларсиз) va UNIX-tizimini meros qilib oladi. XMLhujjatlarini qayta ishlash uchun standart sifatida DOM va SAX, XSLT-transformatsiyasi uchun API dan foydalanishi mumkin.

Elektron tijorat ilovalarini yaratish uchun bir qator to'lovni amalga oshiradigan Cybercash, CyberMUT, VeriSign Payflow Pro hamda CCVS kabi foydali funksiyalar mavjud.

#### **4. Dasturiy vositani sozlash va o'rnatish**

Yuqorida PHP tili imkoniyatlarini, qo'llanilish sohalarini muhokama hildik va tarixini o'rgandik. Endi dasturiy vositani o'rnatishga kerak bo'lgan uskunalar majmuini ko'rib o'tsak. Modomiki, asosiy kursning amaliyoti sifatida biz quyidagi masalalarni ko'rib chiqamiz: mijoz-server texnologiyasi sifatida ishlanadigan masalalar, mos ravishda skriptlar yaratilishida qo'llanilishi, serverlarni qayta ishlash. Bular uchun bizga web-server hamda PHP tili interpretatori kerak bo'ladi. Web-server sifatida, masalan, web-mutaxassislar o'rtasida mashhur bo'lgan Apache serverni olamiz. Dastur natijasini ko'rish uchun web-brauzer kerak bo'ladi, bunga misol Internet Explorer.

#### ***Denver Distributivi***

Biz Yuqorida Linux va Windows platformalari uchun PHP dasturiy vositasini sozlash va o'rnatish bilan etarlicha tanishmiz. PHP dasturiy vositasi va uni ishlashi uchun kerak bo'ladi komponentalarni o'rganishni xohlamaydiganlar uchun PHP dasturining tayyor PHP tilini to'ldiradigan distributivlari mavjud. Bunday distributivlar ichida keng tarqalgani - Denver (<http://dklab.ru/chicken/web/>). Uni o'rnatishni o'rganish uchun web-mutaxassislar saytlariga murojaat qilish kerak. Denvernii o'rnatish juda oddiy hamda unga hech qanday bilim talab etilmasligini aytib o'tish kerak. Bu distributivni PHP tilini endigina o'rganayotgan yosh dasturchilar uchun tavsiya etamiz. Jiddiy masalalarni hal etish uchun esa PHP dasturlash tilini to'lia o'rnatish va sozlash kerak bo'ladi.

#### **Nazorat savollari:**

1. Qanaqa dasturlash tillarini bilasiz?
2. PHP qanday dasturlash tili hisoblanadi;
3. Bu tilni vujudga kelishiga sababchi bo'lgan ehtiyojlar nimadan iborat deb o'ylaysiz?
4. PHP dasturlash tili haqida umumiy ma'lumot bering?
5. PHP dasturlash tilining asosiy tushunchalarini keltiring?
6. PHP dasturlash tili imkoniyatlari haqida gapiring.
7. Server tomonda dasturlash deganda nimani tushunasiz?
8. PHP ni ishga tushirish, Denver paketi va undan foydalanish haqida ma'lumot bering?
9. Boshqa dasturlash tillari bilan PHP dasturlash tilining farqini tushuntiring?
10. PHP tilining qanday afzallliklari bor?

## **17-18-ma'ruza. PHP funksiyalari va ob'yektlari**

**Reja:**

1. PHP funksiyalari va ob'yektlari.
2. PHPda massivlar, satrlar va fayllar bilan ishlash.
3. Formalar bilan ishlash.
4. PHP da xatolarni qayta ishlash

**Kalit so'zlar:** PHP da satr tipi, massiv tipi, ko'p o'lchovli massivlar, kvadrat qavs sintaksisi, massiv initsializasiyasi, foreach sikli, heredoc sintaksisi, tartiblash funksiyalari.

### **1. PHP dasturlash tilida String (satr) tipi**

Satr – bu belgilar to'plamidir. PHP dasturlash tilida belgi bu bir bayt va 256 ta turli belgilar mavjud. PHP dasturlash tili Unicode tipidagi belgilarni qabul qilmaydi. PHP dasturlash tilida amalda satrlarga chegirma mavjud emas, shuning uchun satrlarni ishlatganda uning aniq uzunligi haqida o'ylash shart emas.

PHP dasturlash tilida satrlar uchta turli xil usullarda aniqlanadi:

- bittalik  
qo'shtirnoqlar  
yordamida ('');
- qo'shtirnoqlar  
yordamida (""); □  
heredoc-  
sintaksisi  
yordamida.

#### ***Bittali tirnoqlar***

Satrlarning aniqlashning oddiy usuli – u «'» bittali qo'shtirnoqlar ichida yoziladi. Agarda satr ichida ham bittali tirnoq ishlatalishga to'g'ri kelib qolsa, bittali tirnoqdan oldin «\» belgini qo'yish, ya'ni uni ekranlash shart. Agarda «\» belgi bittali tirnoqdan oldin yoki satrning oxirida bo'lsa, u holda belgini ikkilantirish kerak, ya'ni «//».

Agarda bittali tirnoq ichidagi satr ichida ixtiyoriy belgidan oldin («\» va «'» lardan farqli ravishda) teskari slesh «\» belgisi uchrasa, u holda uni oddiy belgi deb qarab barcha belgilarni o'z holicha ekranga chiqaradi. Shuning uchun teskari slesh «\» belgisini satr oxirida yopiladigan qo'shtirnoqdan avval turganini ekranlash shart.

PHP dasturlash tilida teskari slesh «\» belgisi bilan ifodalanadigan bir qator belgilar majmui mavjud. Ularni ketma-ketlikni boshqaruvchilar deb ataladi hamda ular maxsus vazifalarni bajaradi. Ular haqida keyinroq to'xtalib o'tamiz. O'zgaruvchilar va ketma-ketlikni boshqaruvchilar bittalik qo'shtirnoqlar satri ichida uchrashsa, ular o'rtaсидаги farq ketma-ketlikni boshqaruvchilarni qayta ishlanmaydi.

#### **Misol:**

```
<?php
```

```

echo 'Satrlar majmui';
//Ekranga chiqaradi: ' belgini chiqarish uchun undan oldin \ belgi
qo'yiladi. echo ' Belgini \' chiqarish uchun undan oldin'
 ' //belgini qo'yish kerak';
//Ekranga chiqaradi: Siz shuni o'chirmoqchimisiz
C:*.*? echo ' Siz shuni o'chirmoqchimisiz C:**.*?' ;
//Ekranga chiqaradi: Buni qo'ymang: \n
//yangi qatorga
echo ' Buni qo'ymang: \n yangi qatorga ';
//Ekranga chiqaradi: o'zgaruvchi $expand ham
//$/either qo'yilmaydi echo
'o'zgaruvchi $expand ham $either'
.
 'qo'yilmaydi';
?>

```

## 2. PHP dasturlash tilida Array (massiv) tipi.

PHP dasturlash tilida massiv tipi tartiblangan kartalarga o'xshaydi va qiymatini kalitga o'zlashtiradigan tipdir. Bu tip bir necha yo'naliishlarda optimallashtiriladi, shuning uchun siz uni xususiy massiv, ro'yxat (vektor), xesh-jadvali (kartani amalga oshirish uchun ishlataladi), stek, navbat va boshhalar sifatida foydalanishingiz mumkin. Modomiki, PHP dasturlash tilida bir massivni qiymatini boshqasiga o'zlashtirish uchun daraxtlardan foydalanasiz.

Massivlarni array() konstruktsiyasi yordamida aniqlanadi yoki elementlariga qiymat berish bilan aniqlanadi. **array()** konstruktsiyasi yordamida aniqlash. **array ([key] => value, [key1] => value1, ... )**

PHP dasturlash tilining array() konstruktsiyasi vergul bilan ajratilgan juft parametrlar kalit => qiymat bilan ajratilgan. => belgi mos ravishda qiymat va uning kaliti o'rtasida aloqa o'rnatadi. Kalit butun son bo'lishi mumkin, uning qiymati esa PHP dasturlash tilidagi ixtiyoriy tipni qabul qilishi mumkin. Kalit raqamini biz ko'pincha indeks deb ataymiz. PHP dasturlash tilida indeksslash noldan boshlanadi. Massiv elementining qiymatini olish uchun massiv nomi va kvadrat qavs ichida uning kaliti ko'rsatilishi kerak. Agar massiv kaliti standart butun son bo'lsa, u holda uning qiymatini butun son deb qarasa bo'ladi, aks holda u satr deb qaraladi. Shuning uchun \$a["1"] yozuv \$a[1] yozuvga teng kuchli, \$a["-1"] yozuv esa \$a[-1] yozuvga teng kuchli.

**Misol.** PHP dasturlash tilida massivlar.

```

<?php
$books = array ("php" =>
 "PHP users
guide", 12
=> true); echo
$books["php"];
//ekranga chiqaradi: "PHP users
guide" echo $books[12]; //ekranga
chiqaradi: 1
?>

```

Agarda element uchun kalit berilmagan bo'lsa, u holda kalit sifatida kalitning maksimal qiymatiga bir qo'shib hisoblanadi. Agarda qiymati mavjud kalit ko'rsatilgan bo'lsa, u holda shu kalit qiymatini ekranga chiqaradi. PHP 4.3.0 dasturlash tili versiyasidan boshlab kalitning maksimal qiymati manfiy son deb qaralsa, u holda massivning keyingi kaliti nol (0) bo'ladi.

**Misol.** PHP dasturlash tilida massivlar.

```
<?php
//$arr hamda $arr1 massivlar ekvivalentdir.
$arr = array(5 => 43, 32, 56, "b" => 12);
$arr1 = array(5 => 43, 6 => 32,
 7 => 56, "b" => 12);
?>
```

Agarda TRUE yoki FALSE kalit sifatida qo'llanilsa, u holda uning qiymati mos ravishda integer tipining bir va noliga o'zlashtiriladi. Agar NULL dan foydalanilsa, u holda kalit o'rniga bo'sh satr hosil bo'ladi. Bu bo'sh satrni kalit sifatida foydalansa bo'ladi, ammo uni qo'shtirnoqqa olish kerak bo'ladi. Bu usul bo'sh kvadrat qavsni ishlatish kabi emas. Massivlar yoki ob'ektlar kaliti sifatida foydalanish mumkin ham emas.

### **3. Kvadrat qavs sintaksisi yordamida massivni aniqlash.**

Massivga qiymat berish orqali massiv yaratish mumkin. Biz yuqorida aytib o'tganimizdek, massiv elementi qiymatiga ega bo'lish uchun kvadrat qavs ichiga uning kaliti ko'rsatilishi kerak, masalan, \$book["php"].

Agarda yangi kalit va yangi qiymat ko'rsatsangiz quyidagicha bo'ladi: \$book["new\_key"] = "new\_value" hamda massivga yangi element qo'shiladi. Agarda kalitni ko'rsatmay faqat qiymatni o'zlashtirsak, ya'ni \$book[] = "new\_value", u holda massivga yangi element qo'shiladi va uni kaliti mavjud maksimal qiymatga bir qo'shiladi. Agarda biz qiymat bergen massiv yaratilmagan bo'lsa, u holda biz qiymat bergandan keyin u yaratiladi.

```
<?
$books["key"] = value; //key kaliti bilan value qiymat $books massiviga
qo'shiladi
$books[] = value1; /* 13-kalit bilan value1 qiymati massivga
qo'shiladi, chunki bizda kalitning maksimal qiymati 12 edi. */
?>
```

Massivning aniq bir elementini o'zgartirish uchun uning shu kaliti bilan yangi qiymatga o'zlashtirish kerak. Massiv elementi kalitini o'zgartirish mumkin emas, faqat o'chirish (kalit va elementi juftligini) va yangi qo'shish mumkin xolos. Massiv elementini o'chirish uchun unset() Funksiyasidan foydalanish kerak.

```
<?php
$books = array ("php" =>"PHP users guide", 12 => true);
$books[] = "Book about Perl"; /* 13-kalit(indeks) bilan yangi element
qo'shildi, bu quyidagiga ekvivalent $books[13] = "Book about Perl";
```

```
$books["lisp"] = 123456; /* Bu massivga yangi "lisp" kalitli 123456
qiymatli yangi element qo'shish*\ unset($books[12]); //Bu 12-kalitli
elementni massivdan o'chirish unset ($books); //massivni butunlay
o'chirish
```

?> Bo'sh kvadrat qavsdan foydalanganda kalitning maksimal qiymati massivda mavjud oxirgi qayta indekslangan kalitlar orasidan qidiriladi. Massivni array\_values() funksiyasi yordamida qayta indeksslash mumkin.

**Misol.** Massivni qayta indekslaymiz.

```
<?php
$arr = array ("a","b","c");
/* "a", "b" va "c" qiymatli massivni yaratamiz. Bu erda kalit
ko'rsatilmagan
Biroq mos ravishda ular 0,1,2 bo'ladi. *\ print_r($arr);
//massivni ekranga chiqaramiz (kaliti va qiymatini)
unset($arr[0]); unset($arr[1]); unset($arr[2]);
//massivdan hamma elementini o'chiramiz
print_r($arr); //massivni ekranga chiqaramiz (kaliti va qiymatini)
$arr[] = "aa"; //massivga yangi element qo'shamiz. Uni
indeksi(kaliti) 3 bo'ladi, 0 emas. print_r($arr); $arr =
array_values($arr); //massivni qayta indekslaymiz.
$arr[] = "bb"; //bu elementni kaliti 1 bo'ladi.
print_r($arr);
?>
```

Bu skriptning natijasi quyidagicha bo'ladi:

```
Array ([0] => a [1] => b [2] => c)
Array ()
Array ([3] => aa)
Array ([0] => aa [1] => bb)
```

#### 4. Massivlar initsializatsiyasi

PHP da massivlarni initsializatsiya qilishning 2 usuli mavjud. Birinchisi massiv elementlariga qiymat berishdan iborat:

```
<
 $car[] = "passenger car";
 $car[] = "land-rover";
 echo($car[1]); // chiqaradi "land-rover"
?>
```

Massiv indeksini ochiq ko'rsatish mumkin: <?

```

$car[0] = "passenger car";
$car[1] = "land-rover";
echo($car[1]); // chiqaradi "land-rover"
?>

```

Agar massiv elementlarini e'lon qilishda oshkora indeksatsiyali va indeksatsiyasiz o'zgaruvchilar aralashib kelsa indeksi berilmagan elementga ishlatalgan indekslar ichida eng kattasidan keyin keluvchi ruxsat berilgan indeksni beradi. Masalan agar biz yaratgan massiv elementlari indekslari 10, 20 va 30 bo'lsa va indeks ko'rsatmasdan Yangi element yaratsak, uning indeksi avtomatik ravishda 31 bo'ladi:

```

<?
$car[10] = "passenger car";
$car[20] = "land-rover";
$car[30] = "station-wagon";
$car[] = "victoria";
echo($car[31]);
?>

```

Alternativ usul **array()** konstruktsiyasidan foydalanishdan iborat:

```

<?
$car = array("passenger car", "land-rover");
echo($car[1]); // chiqaradi "land-rover"
?>

```

Indekslarni oshkora ko'rsatish uchun => operator qo'llanadi:

```

<?
$car = array("passenger car", 5 => "land-rover",
 "station-wagon", "victoria");
echo($car[0]); echo("
"); // chiqaradi "passenger car"
echo($car[5]); echo("
"); // chiqaradi "land-rover"
echo($car[6]); echo("
"); // chiqaradi "station-wagon"
echo($car[7]); // chiqaradi "victoria"
?>

```

Massiv indekslari satrlar ham bo'lishi mumkin:

```

<?
$car = array("pc" => "passenger car", "lr" => "land-rover");
echo($car["lr"]); echo("
"); // chiqaradi "land-rover"
echo($car["pc"]); // chiqaradi "passenger car"
?>

```

## 5. Massivlarni ko'rib chiqish uchun foreach sikli

PHP4 da massiv elementlarini ko'rib chiqish uchun **foreach** operatoridan foydalanish mumkin. Bu operator sintaksisi:

```
foreach (array as [$key =>] $value)
{
 statements;
}
```

Busikl ma'nosi sodda: har bir element ko'rilmagan uding indeksi **\$key** o'zgaruvchiga, qiymati bo'lsa

**\$value** o'zgaruvchiga joylashtiriladi. Bu ikki o'zgaruvchilarning nomlari ixtiyoriydir.

**Misol:**

```
<
?

$car = array("passenger car", "land-rover",
 "station-wagon", "victoria");
foreach($car as $index => $val)
{
 echo("$index -> $val
");
}
?
>
```

Sintaksidan ko'rinish turibdiki, **\$key** o'zgaruvchidan foydalanish shart emas, shuning uchun tashlab yuborilishi mumkin:

```
<?
echo(
 "available cars:
 "
);
$car = array("passenger car", "land-rover",
 "station-wagon", "victoria");
foreach($car as $val)
{
 echo("$val\n");
}
echo("");
?>
```

## 6. Ko'p o'lchovli massivlar

Ko'p o'lchovli massivlarni ko'rib chiqish uchun ichki joylashgan **array()** konstruktsiyasidan foydalilanadi. Ko'p o'lchovli massivlarni o'qib chiqish joylangan sikllar yordamida amalgam oshiriladi. Quyidagi scriptda ko'p o'lchovli massiv yaratish va ko'rib chiqish ko'rsatilgan.

**Misol:**

```
<?
$ship = array(
 "Passenger ship" => array("Yacht", "Liner", "Ferry"),
 "War ship" => array("Battle-wagon", "Submarine", "Cruiser"),
 "Freight ship" => array("Tank vessel", "Dry-cargo ship", "Container
cargo ship")
);
foreach($ship as $key => $type)
{
echo(
 "<h2>$key</h2>\n". "\n");
foreach($type as $ship)
{
 echo("\t$ship\n");

}
}
echo("\n");
?>
```

Bu script bajarilish natijasi:

```
Passenger ship
Yacht
Liner
Ferry
War ship
Battle-wagon
Submarine
Cruiser
Freight ship
Tank vessel
Dry-cargo ship
Container cargo ship
```

Endi PHP da mavjud massivlar bilan ishslash funksiyalarini ko‘ramiz. Biz massivlarni tartiblash funksiyalardan boshlaymiz. Lekin avval misollarimizda ko‘p foydalanadigan uchta funksiyani ko‘rib chiqamiz.

### **1. Funksiya count()**

*Sintaksis:*

```
int count(mixed var)
```

Bu funksiya argument sifatida massivni qabul qilib, undagi elementlar sonini qaytaradi.

### **2. Funksiya in\_array()**

*Sintaksis:*

```
boolean in_array(mixed needle, array haystack [, bool strict])
```

Bu funksiya **haystack** massivda **needle** qiymatni qidiradi va agar u mavjud bo'lsa **true** qaytaradi, aks xolda **false** qaytaradi.

### 3. Funksiya **reset()**

*Sintaksis:*

```
mixed reset(array array)
```

Funksiya **reset()**massiv ko'rsatkichini birinchi elementga o'rnatadi va massiv birinchi elementi qiymatini qaytaradi.

Kichkina izox. PHP da har bir massiv joriy elementga ko'rsatkichga ega. Quyidagi **foreach** kabi konstruktsiyalar bilan ishlashda ko'rsatkich haqida o'ylash kerak emas, chunki foreach uni massiv boshiga o'rnatadi.Lekin boshqa massivlar funksiyalari masalan **prev()**, **next()**, massiv ko'rsatkichlarini suradi, bu esa **array\_walk()**, kabi qayta ishlashni ko'rsatkichni turgan joyidan boshlovchi funksiyalar uchun katta ahamiyatga ega.

Endi tartiblash bilan shug'ullanamiz.

## 7. Massivlarni tartiblash funksiyalari

Quyidagi tartiblash funksiyalari mavjud:

### □ **sort()**

Massivni o'sish bo'yicha tartiblash funksiyasi.

*Sintaksis:*

```
void sort(array array [, int sort_flags])
```

Funksiya **array** massivini o'sish bo'yicha tartiblaydi. Majburiy bo'lмаган element **sort\_flags** elementlar qanday tartiblanishi kerakligini ko'rsatadi(tartiblash bayroqlarini belgilaydi). Argumentning mumkin bo'lган qiymatlari quyidagilar:

- ✓ **SORT\_REGULAR** – elementlar normal soldishtiradi  
normalnoe srovnenie elementov

(elementlarni "boricha" solishtiradi)

- ✓ **SORT\_NUMERIC** – elementlarni sonlar sifatida solishtiradi
- ✓ **SORT\_STRING** - elementlarni satrlar sifatida almashtiradi

Umuman olganda bu funksiya ro'yxatlarni tartiblash uchun mo'ljallangan. Ro'yxat deganda kalitlari nuldan boshlangan va bo'shlklarga ega bo'lмаган massiv tushuniladi. Funksiya **sort()** ixtiyoriy massivni ro'yxat deb qaraydi.

**Misol:**

```
<
?
```

```
$arr = array("2", "1", "4", "3", "5");
sort($arr);
for($i=0; $i < count($arr); $i++)
```

```

{
 echo ("$i:$arr[$i] ");
}
// chiqaradi "0:1 1:2 2:3 3:4 4:5"
?>

```

*Natija:*

0:1 1:2 2:3 3:4 4:5

Agar siz satrlarni tartilayotgan bo‘lsangiz, misol uchun massiv quyidagi ko‘rinishga ega bo‘lsa  
`array("one", "two", "abs", "three", "uic", "for", "five");`

Bu ajoyib funksiya quyidagi natijani  
qaytaradi: *Natija:*

0:abs 1:five 2:for 3:one 4:three 5:two 6:uic

Ya’ni satrlarni u alfa-beta tartibda, soddarroq aytganda birinchi harflari alfavitda kelishi bo‘yicha  
tartiblaydi.

- **rsort()**

Massivlarni kamayish bo‘yicha tartiblash.

*Sintaksis:* void **rsort**(array **arr**

[, int **sort\_flags**])

Shunga o‘xshash **sort()** funksiyasi faqat kamayish bo‘yicha tartiblaydi. Oldingi **sort()** funksiyasi  
uchun ko‘rilgan scriptni olamiz, faqat sort(**\$arr**) o‘rniga rsort(**\$arr**) qo‘yamiz. *Natija:*

0:5 1:4 2:3 3:2 4:1

- **asort()**

Assotsiativ massivni o‘sish bo‘yicha tartiblash.

*Sintaksis:* void **asort**(array **arr**

[, int **sort\_flags**])

Funksiya **asort()** berilgan **arr** massivni shunday tartiblaydiki uning qiymatlari alfavit tartibida  
(agar satr bo‘lsa) yoki o‘sish tartibda (sonlar uchunl) tartibda joylashadi. Bu funksiyaning **sort()**  
funksiyasidan muxim farqi shundaki **asort()** funksiyasi qo‘llanilganda kalitlar va ularga mos  
qiymatlar orasida bog‘liklik saqlanadi, **sort()** funksiyasida bo‘lsa bu bog‘liqlik uziladi.

**Misol:**

```
<
?
```

```

$arr = array("a" =>"one", "b" => "two", "c" => "three", "d" => "four");
asort($arr);
foreach($arr as $key => $val)
{
 echo (" $key => $val ");
}

```

?

>

*Natija:* d => four a => one c

=> three b => two

Ko‘rinib turibdiki "kalit-qiyamat" bog‘lanishlari saqlanib qolgan.

Ko‘zda tutilgan bo‘yicha **asort()** funksiyasi massivni alfavit bo‘yicha tartiblaydi. Tartiblashbayroqlari **sort\_flags** qiymatlari **sort()** funksiyasi ta’rifida keltirilgan.

- **arsort()** Assotsiativ massivlarni kamayish bo‘yicha tartiblash.

*Sintaksis:* void **arsort(array arr**

**[, int sort\_flags])**

Bu funksiya **asort()** funksiyasiga o‘xhash, faqat u masivni o‘sish bo‘yicha emas kamayish bo‘yicha tartiblaydi.

- **ksort()**

Massivlarni kalit o‘sishi bo‘yicha tartiblash. *Sintaksis:* int **ksort(array**

**arr [, int sort\_flags])**

Bu funksiyada tartiblash qiymatlar bo‘yicha emas, balkim kalitlar bo‘yicha o‘sish tartibida amalga oshiriladi.

<?

```
$arr = array("a" =>"one","b" => "two","c" => "three","d" => "four");
ksort($arr);
foreach($arr as $key => $val)
{
 echo (" $key => $val ");
}
```

?

>

*Natija:*

a => one b => two c => three d => four

- **krsort()**

Indekslar kamayishi bo‘yicha massivlarni tartiblash.

*Sintaksis:* int **krsort(array arr**

**[, int sort\_flags])**

Xuddi **ksort()** funksiyaga o‘xhash, faqat massivni kalitlar bo‘yicha teskari tartibda (kamayish bo‘yicha) tartiblaydi.

- **array\_reverse()**

Massiv elementlarini teskari joylashtirish.

*Sintaksis:* array **array\_reverse(array arr [, bool preserve\_keys])**

Funksiya **array\_reverse()** elementlari parametrda berilgan **arr** massivi elementlariga nisbatan teskari joylashtirilgan massivni qaytaradi. Kalitlar va qiymatlar orasidagi bog'lanish saqlanib qoladi. Agar majburiy bo'limgan parametr **preserve\_keys** ga **true** berilsa, kalitlar ham teskari tartibda joylashadi.

**Misol:**

```
<?
 $arr = array ("php", 4.0, array ("green", "red"));
 $result = array_reverse
($arr); echo "Massiv:
";
 foreach($result as $key => $val)
 {
 echo ("$key => $val
");
 }
echo("
")
;
 echo "Tartiblangan massiv:
";
$result_keed = array_reverse ($arr,
false); foreach($result_keed as $key
=> $val)
{
 echo ("$key => $val
 ");
}
?>
```

Birinchi xolda:

*Natija:*

Massiv:

0 =>Array

1 =>4

2 =>php

Tartiblangan massiv:

0 =>Array

1 =>4

2 =>php

Agar ikkinchi parametrga **true** qiymat berilsa:

*Natija:*

Massiv:

0 =>Array

1 =>4

2 =>php

Tartiblangan massiv:

2 =>Array

1 =>4

0 =>php

- **shuffle()**

Massiv elementlarini tasodifiy joylashtirish.

*Sintaksis:* void **shuffle(array arr)** **shuffle()** funksiyasi **arr** massivi elementlarini tasodifiy aralashtiradi.

- **natsort()**

Tabiiy tartiblashni bajaradi.

*Sintaksis:* void **natsort(array arr)** Bunday tartiblashni satrlarni tartiblashda uchratgan edik. Tabiiy tartibash deb elementlar tushunarli tartibda joylashga aytildi.

**Misol:**

```
<?
 $array1 = $array2 = array("pict10.gif", "pict2.gif", "pict20.gif",
 "pict1.gif"); echo ("oddiy
tartibash:"); echo ("
");
sort($array1); print_r($array1);
 echo ("
"); echo ("tabiiy tartibash:"); echo
("
"); natsort($array2); print_r($array2);
?>
```

*Natija:*

oddiy tartibash:

Array ( [0] => pict1.gif [1] => pict10.gif [2] => pict2.gif [3] => pict20.gif ) estestvennaya sortirovka:

Array ( [3] => pict1.gif [1] => pict2.gif [0] => pict10.gif [2] => pict20.gif )

**Nazorat savollari:**

1. PHP dasturlash tilida qanday tiplarni bilasiz? 2. PHP dasturlash tilida satr tipini tushuntirib bering?
3. PHP dasturlash tilida satrlar qanday aniqlanadi? 4. PHP da massivlar tushuntirib bering?
5. Massiv tipi qanday aniqlanadi?
6. Massivlarni iinitsializatsiya qilishning qanday usullarini bilasiz?
7. Qanday tartiblash funksiyalarini bilasiz?
8. Turli tartiblash funksiyalarini ishlashga doir misollar bilan tushuntiring?
9. Ko'p o'lchovli massivlar haqida ma'lumot bering?
10. Foreach siklininh ishlash mohiyatini tushuntiring?

## 19-ma'ruza: PHPda global o'zgaruvchilar

### ***Reja:***

1. PHPda global o'zgaruvchilar: POST, GET, COOKIE, SESSION, SERVER, REQUEST, ENV, FILES.
2. O'zgaruvchilarning ko'rinish sohalari.
3. Fayllar ustida amallar.
4. Kataloglar bilan ishslash.

### ***O'zgaruvchilar, o'zgarmaslar va operatorlar***

Har bir dasturlash tilida muhim elementlardan biri bu o'zgaruvchilar, o'zgarmaslar va ular qo'llaniladigan operatorlardir. Bu kattaliklar haqida biz 9-ma'ruzada ham to'xtalib o'tgan edik. PHP dasturlash tili bu elementlarni qanday belgilashi va qayta ishslashini ko'rib chiqamiz.

**O'zgaruvchilar** PHP dasturlash tilida o'zgaruvchilar oldiga dollar belgisi ("\$") qo'yib e'lon qilinadi, masalan, \$my\_var.

O'zgaruvchilar nomlari registrlarni farqlaydi, ya'ni \$my\_var hamda bosh harfli \$My\_var o'zgaruvchilari turli xil o'zgaruvchilardir.

PHP dasturlash tilida o'zgaruvchilar nomi qolgan dasturlash tillari qoidalari kabi e'lon qilinadi: o'zgaruvchi nomi lotin alfaviti bilan boshlanishi va undan keyin harflar yoki tagiga chizilgan belgi yoki rahamlar bo'lishi mumkin.

PHP4 dasturlash tilida bularidan tashqari o'zgaruvchiga qiymat o'zlashtirishning yana bir usuli mavjud: ssilka bo'yicha o'zlashtirish. Ssilka bo'yicha o'zgaruvchiga qiymat o'zlashtirish uchun uni nomi bo'lishi shart, ya'ni u qandaydir o'zgaruvchini taqdim etishi kerak. Bir o'zgaruvchi qiymatini boshqa o'zgaruvchiga Ssilka bo'yicha o'zlashtirish uchun birinchi o'zgaruvchi oldiga ampersand & belgisi qo'yish shart. Bunga Yuqoridagi misolni ko'rib chihamiz, faqat first o'zgaruvchi second o'zgaruvchiga ssilka bo'yicha o'zlashtiriladi:

**Misol.** Ssilkalar bo'yicha o'zlashtirish.

```
<?php
$first = ' Text ';// $first o'zgaruvchiga
// ' Text ' qiymat o'zlashtirildi
$second = &$first;
/* $second.orqali $first o'zgaruvchiga ssilka qilamiz
Endi bu o'zgaruvchilar qiymatlari har doim
tengdir */
// $first o'zgaruvchi qiymatini
// ' New text ' qiymatga o'zgartiramiz
$first = ' New text ';
echo "first nomli o'zgaruvchi qiymati $first ga teng
"; //
$second o'zgaruvchi qiymatini ekranga chiqaramiz
echo "second nomli o'zgaruvchi qiymati " . "$second ga teng"; ?>
```

Bu skriptni natijasi esa quyidagicha bo'ladi:

first nomli o'zgaruvchi qiymati New text ga teng. second  
nomli o'zgaruvchi qiymati New text ga teng.

Ya'ni \$first o'zgaruvchi qiymati o'rniga \$second o'zgaruvchi qiymati o'zlashtirildi.

### **O'zgarmaslar**

Skript bajarilish jarayonida o'zgarmaydigan qiymatli kattaliklarni saqlash uchun o'zgarmaslardan foydalaniladi. Bunday kattaliklar matematik o'zgarmaslar, parollar, fayllarning yo'llari va boshqalar bo'lishi mumkin. O'zgarmaslarning o'zgaruvchilardan asosiy farqi shuki, ularni faqat bir martagina o'zlashtiriladi va uni qiymatini e'lon qilingandan keyin bekor qilib bo'lmaydi. Bundan tashqari o'zgarmaslar oldida dollar belgisi qo'yilmaydi hamda uni oddiy qiymat o'zlashtirish kabi qarash mumkin emas. O'zgarmaslar qanday aniqlanadi? Buning uchun maxsus define() funksiyasi mavjud, uning sintaksisi quyidagichadir:

```
define("O'zgarmas nomi", "O'zgarmas qiymati", [registrga_sezgirligi_kichik])
```

O'zgarmaslar nomi registrga segirligi katta. har bir o'zgarmaslarda uni o'zgartirish mumkin, ya'ni registrga\_sezgirligi\_kichik argumentni qiymati sifatida True qiymati ko'rsatiladi. O'zgarmaslar nomi har doim katta registr bilan yozishga kelishib olingan.

O'zgarmasni qiymatini bilish uchun uni nomini ko'rsatish kerak. O'zgaruvchidan farqi o'zgarmas nomi oldiga \$ belgi qo'yilmaydi. Bundan tashqari o'zgarmasni qiymatini bilish uchun konstanta nomi bilan parametr sifatida constant() funksiyasidan foydalanish mumkin.

**Misol.** PHP dasturlash tilida o'zgarmaslar.

```
<?php
// o'zgarmasni aniqlaymiz PASSWORD define("PASSWORD","qwerty");
// registrланмаган PI o'zgarmasni qiymatini aniqlaymiz 3.14 define("PI","3.14",
True);
// PASSWORD o'zgarmas qiymatini olamiz, ya'ni qwerty echo
(PASSWORD);
// bu ham qwerty ni chiqaradi echo constant("PASSWORD"); echo (password); /*
password ni chiqaradi va biz registrланган o'zgarmas PASSWORD ni
kutgандик.*/ echo pi;
// 3.14 ni chiqaradi, chunki o'zgarmas PI registrланмаган va aniqlangan. ?>
```

Dasturchi tomonidan o'zgaruvchilardan tashqari Yuqorida aytib o'tganimizdek PHP dasturlash tilida mavjud o'zgarmaslar ham interpretator tomonidan aniqlanadi. Masalan, \_\_FILE\_\_ o'zgarmas dastur bajarilish jarayonida fayl nomini (va fayl yo'lini), \_\_FUNCTION\_\_ funksiya nomidan tashkil topadi, \_\_CLASS\_\_ - sind nomi, PHP\_VERSION – PHP dasturlash tili interpretatori versiyasini o'zida saqlaydi. Bunday o'zgarmaslarning barcha ro'yxatini PHP dasturlash tili uchun mo'ljalangan qo'llanmalardan topish mumkin.

### **Amallar.**

O'zgaruvchilar, o'zgarmaslar va ifodalar ustida turli hisoblashlarni bajaradigan bu amallardir. Biz hali bu ifodalar haqida to'xtab o'tganimiz yo'q. Ifodalar qiymatini ushbu amallar yordamida aniqlanadi. O'zgaruvchilar va o'zgarmaslar – bu ifodalarning asosiy va juda sodda shaklidir. Shunday ifodalarni ko'paytirishi mumkin bo'lgan amallar to'plami mavjud. Ularni quyida to'liqroq muhokama qilamiz:

### 9.1-jadval . Arifmetik amallar.

| <b>Belgilanishi</b> | <b>Nomlanishi</b>  | <b>Misollar</b> |
|---------------------|--------------------|-----------------|
| +                   | Qo'shish           | $\$a + \$b$     |
| -                   | Ayirish            | $\$a - \$b$     |
| *                   | Ko'paytirish       | $\$a * \$b$     |
| /                   | Bo'lish            | $\$a / \$b$     |
| %                   | Bo'lishdagi qoldiq | $\$a \% \$b$    |

### 9.2-jadval. Satrli amallar.

| <b>Belgilanishi</b> | <b>Nomlanishi</b>                          | <b>Misollar</b>                                                       |
|---------------------|--------------------------------------------|-----------------------------------------------------------------------|
| .                   | <i>Konkatenatsiya (satrlarni qo'shish)</i> | $\$c = \$a . \$b$<br>(bu $\$c$ satr $\$a$ va $\$b$ satrlardan iborat) |

### 9.3-jadval. O'zlashtirish amallari.

| <b>Belgila<br/>nishi</b> | <b>Nomla<br/>nishi</b> | <b>Izohlar</b>                                                                                                                                                                                                                               | <b>Misollar</b>                                             |
|--------------------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| =                        | O'zlash<br>tirish      | <i>Operatoridan tomonda turgan <math>\\$a = (\\$b = 4) + 5</math>; o'zgaruvchilar ustida bajarilgan (<math>\\$a</math> ga teng, <math>\\$b</math> amallardan hosil bo'lgan natija 4 ga teng) qiymati o'zlashtiriladi.</i>                    |                                                             |
| +=                       |                        | Qisqartirish. O'zgaruvchiga son qo'shiladi va keyin natija o'zlashtiriladi.                                                                                                                                                                  | $\$a += 5;$<br>$(\$a = \$a + 5$<br>ifodaga<br>ekvivalent; ) |
| .=                       |                        | zlashtirish va konkatenatsiya amallari $\$b = "Hammaga"$ ; kombinatsiyasini qisqartirilgan shakli $\$b .= "salom"$ ;<br>(dastavval satrlar qo'shiladi, so'ngra ( $\$b = \$b .$ hosil bo'lgan satr o'zgaruvchiga "salom" ifodaga o'zlashadi). | ekvivalent;) Natijasi:<br>$\$b = "Hammaga salom"$           |

#### 9.4-jadval. Mantiqiy amallar.

| Belgilanishi | Nomlanishi | Izohlar                                                                                                  | Misollar |
|--------------|------------|----------------------------------------------------------------------------------------------------------|----------|
| and          | VA         | \$a VA \$b rost (True)                                                                                   | \$a and  |
| &&           | VA         |                                                                                                          | \$b &&   |
| or           | YOKI       | \$a yoki \$b o'zgaruvchilardan hech bo'limganda bittasi rost bo'lsa (ikkalasi ham rost bo'lishi mumkin). | \$a or   |
|              | YOKI       |                                                                                                          | \$a      |
| xor          | INVERSIY   | O'zgaruvchilardan bittasi rost bo'lsa .                                                                  | \$b      |
|              | A YOKI     | Agarda ikkalasi ham rost bo'lsa inersiyalanadi.                                                          |          |
| !            | INVERSIY   | Agarda \$a=True, u holda !\$a=False va aks \$a xor                                                       |          |
|              | A (NOT)    | holda teskari bo'ladi.                                                                                   | \$b      |
|              |            |                                                                                                          | ! \$a    |

#### 9.5-jadval. Taqqoslash amallari.

| Belgilanishi | Nomlanishi       | Izohlar                                    | Misollar    |
|--------------|------------------|--------------------------------------------|-------------|
| ==           | Tenglik          | O'zgaruvchilar qiymatlari teng.            | \$a == \$b  |
| ====         | Ekvivalentlik    | O'zgaruvchilar qiymatlari va tiplari teng. | \$a === \$b |
| !=           | Tengsizlik       | O'zgaruvchilar qiymatlari teng emas.       | \$a != \$b  |
| <>           | Tengsizlik       |                                            | \$a <> \$b  |
| !==          | Noekvivalentlik  | O'zgaruvchilar ekvivalent emas.            | \$a !== \$b |
| <            | Kichik           |                                            | \$a < \$b   |
| >            | Katta            |                                            | \$a > \$b   |
| <=           | Kichik yoki teng |                                            | \$a <= \$b  |
| >=           | Katta yoki teng  |                                            | \$a >= \$b  |

#### 9.6-jadval. Inkrement va decrement amallari.

| Belgilanishi | Nomlanishi                                                              | Izohlar | Misollar                           |
|--------------|-------------------------------------------------------------------------|---------|------------------------------------|
| ++\$a        | Pre_ \$a qiymati birga oshiriladi va \$a inkrement qiymati qaytariladi. |         | <? \$a=4; echo "4 bo'lishi shart:" |

|       |                 |                                                                       |                                                  |
|-------|-----------------|-----------------------------------------------------------------------|--------------------------------------------------|
| \$a++ | Post- inkrement | \$a qiymati qaytariladi va so'ngra \$a qiymati birga oshiriladi.      | va .\$a++; echo "6 bo'lishi shart:" .+\$a;<br>?> |
| --\$a | Pre- inkrement  | \$a qiymati birga kamaytiriladi va \$a dekrement qiymati qaytariladi. |                                                  |
| \$a-- | Post-dekrement  | \$a qiymati qaytariladi va so'ngra \$a qiymati birga kamaytiriladi.   |                                                  |

### 3. Ma'lumotlar tiplari bilan ishlash

PHP dasturlash tili sakkizta sodda ma'lumot tiplarini qo'llab quvvatlaydi:

To'rttasi skalyar tiplar:

- boolean (mantiqiy);
- integer (butun);
- float (nuqtasi siljiydigan);
- string (satrli).

Ikkitasi aralish

tiplar: □ array  
(massiv);  
• object (ob'ekt).

Ikkitasi maxsus tiplar:

- resource (resurs); □ NULL.

PHP dasturlash tilida o'zgaruvchilar tiplari oshkora e'lon qilinmaydi. Ko'pincha o'zgaruvchi qo'llanilgan konteksdan, ya'ni o'zgaruvchiga o'zlashtirilgan qiymat itpidan mustahil ravishdagi dastur bajarilish jarayonidan interpretator o'zi bu ishni bajaradi. Quyida yuqorida sanab o'tilgan ma'lumotlar tiplarini birma-bir ko'rib chiqamiz.

**Boolean tipi(Bul yoki mantiqiy tip).**

Bu sodda tip qiymatni rost ekanligini ifodalaydi, ya'ni o'zgaruvchi faqat ikkita qiymat qabul qiladi – rost TRUE yoki yolg'on FALSE. Mantiqiy tiplarni aniqlash uchun TRUE yoki FALSE kalit so'zlaridan foydalanamiz. Bu ikkala tiplar registrланмаган.

**Misol.** Mantiqiy tip.

```
<?php
$test = True;
?>
```

Mantiqiy tiplar turli boshqariladigan konstruktsiyalarda (sikllar, shartlar va shunga o'xshash, bular haqida keyinroq aytib o'tamiz) qo'llaniladi. Bir qancha amallar (masalan,

tenglik amali) ham mantiqiy tip qabul qilishi mumkin, ya'ni faqat ikki qiymat rost yoki yolg'on qiymatni qabul qiladi. Ular boshqariladigan konstruktorsiyalarda shartlarni tekshirish uchun qo'llaniladi. Masalan, shartli konstruktorda amallar yoki o'zgaruvchilar qiymati haqiqiyligini tekshiradi va natijadan qat'iy nazar shu yoki boshqa amallarni bajarilishini tekshiradi. Bu erda shart rost yoki yolg'on bo'lishi mumkin, chunki mantiqiy tip amallari va o'zgaruvchilar ko'rsatilgan.

**Misol.** Mantiqiy tiplarning qo'llanilishi.

```
<?php
// '++' amal tenglikka tekshiradi mantiqiy qiymatni qaytaradi if
($know ++ False) { // agar $know qiymat false bo'lsa echo
 "PHP dasturlash tilini o'rgan!";
}
if (!$know) { // xuddi Yuqoridagidek $know qiymati false bo'ladi echo " PHP dasturlash
tilini o'rgan!"; } /* ++ amal $action o'zgaruvchi qiymati bilan "PHP dasturlash tilini
o'rganish!" satrni ustma-ust tushishini tekshiradi. Agar ustma-ust tushsa true qiymatni
qaytaradi, boshqa holda false ni qaytaradi. Agar true ni qaytarsa figurali qavs ichidagi amallar
bajariladi. */ if ($action ++ " PHP dasturlash tilini o'rganish ")
{ echo "O'rganishni boshladim";}
?>
```

**Integer (butun) tipi.**

Bu tip butun sonlar to'plamidan  $Z = \{\dots, -2, -1, 0, 1, 2, \dots\}$  birini qaytaradi. Butun sonlar xohishga qarab oldiga «-» yoki «+» belgilarni qo'yib sanoq sistemasini o'nlik, o'n oltilik yoki sakkizlik tizimlarida ko'rsatilgan bo'lishi mumkin.

Agar siz sakkilik sanoq sistemasidan foydalanayotgan bo'lsangiz, oldindan 0 (nol) rahamini ko'rsatishingiz kerak. O'n oltilik sanoq sistemasida esa raqamlar oldiga 0x belgini qo'yish shart.

```
<?php
o'nlik raham
$a = 1234;
manfiy son
$a = -123;
sakkizlik son (o'nlik sistemasidagi
83 songa ekvivalent)
$a = 0123;
o'n oltilik son (o'nlik sistemasidagi
26 songa ekvivalent)
$a = 0x1A;
?>
```

Butun sonni o'lchami platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati ikki milliard (bu ishorali 32 bitli qiymat) atrofida bo'ladi. Ishorasiz butun sonni PHP dasturlash tili qo'llab quvvatlamaydi.

Agar siz butun son chegarasidan tashqari biror qiymat bersangiz interpretator bu sonni qo'zg'aluvchan vergulli songa o'zgartiradi. Xuddi shunday butun son chegarasidan tashqari

chiqib ketadigan biror amal bajarsangiz ham bu sonni qo'zg'aluvchan vergulli songa o'zgartiriladi.

PHP dasturlash tilida butun sonlarni bo'lish amali mavjud emas. 1/2 ifoda qiymati qo'zg'aluvchan vergulli son 0.5 ga teng. Siz natijangizni butun tipga standart qoida asosida yoki round() funksiyasidan foydalangan taqdirda o'zgartirishingiz mumkin. O'zgaruvchini aniq bir tipga o'zgartirish uchun uning oldiga qavs ichida kerakli tipni yozish kerak bo'ladi. Masalan, \$a=0.5 o'zgaruvchini butun tipga o'zgartirish uchun (integer)(0.5) yoki (integer) \$a ko'rinishda yoki qisqartirilgan (int)(0.5) ko'rinishda yozish kerak bo'ladi. Bunday oshkora yangi tipga o'tish imkoniyati barcha ma'lumotlar tiplari uchun o'rinci bo'ladi (albatta, har doim ham qiymatni bir tipdan boshqasiga olib o'tish shart emas). Biz keltirilgan barcha tiplarni chuqur o'rganishimiz shart emas, chunki PHP dasturlash tili konteksdan mustahil ravishda o'zi bu ishlarni bajaradi.

### ***Float (qo'zg'aluvchan vergulli son) tipi.***

Qo'zg'aluvchan vergulli sonlar (ular ikki karra aniqlik yoki haqiqiy sonlardir) quyidagi sintaksislardan yordamida aniqlanishi mumkin:

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Qo'zg'aluvchan vergulli sonni o'lchami ham platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati ~1.8e308 aniqlik bilan 14 xonali raqam atrofida bo'ladi. **Resource (resurslar) tipi.**

Resurs – bu tashqi resursga (masalan, ma'lumotlar bazasi bilan bog'lanish ) ssilka orqali bog'langan maxsus o'zgaruvchidir. Resurslar maxsus funksiyalar (masalan, mysql\_connect(), pdf\_new() va shunga o'xshashlar) yordamida yaratiladi va foydalaniladi.

### ***Null tipi.***

Maxsus NULL qiymati o'zgaruvchini qiymatga ega emasligi haqida ogohlantiradi.

O'zgaruvchi NULL qiymat qabul qiladi, agarda:

- unga o'zgarmas NULL (\$var = NULL) o'zlashtirilgan bo'lsa; □  
unga hech qanday qiymat berilmagan bo'lsa;
- u unset() funksiya yordamida tozalangan bo'lsa.

NULL tipi faqat bitta qiymati mavjud – registriga sezgirligi kichik NULL kalit so'zidir.

### ***Masalaning echilishi.***

Endi bo'limning boshida qo'yilgan masalaga qaytsak. U turli sabablar bo'yicha har xil odamlarga tuzilgan maktubni jo'natishdan iborat edi. Bu masalani hal etish uchun o'rganilgan vositalardan – o'zgaruvchilar, amallar, o'zgarmaslar, satrlar va massivlardan foydalanishiga harakat qilamiz. Ko'rsatilgan maktub qabul hiluvchiga bog'liq ravishda murojaat va holati o'zgaradi, shuning uchun tabiiy ravishda bu kattalikni o'zgaruvchi deb belgilaymiz. Bundan tashqari hodisalar va odamlar ko'p, shuning uchun massiv o'zgaruvchi tipidan foydalanish qulay. Maktub matni har doim o'zgarmas, shuning uchun uni o'zgarmas deb berish maqsadga muvofiqdir. Juda uzun va qo'pol satrlarni yozmaslik uchun satrlar konkatenatsiya (qo'shish) amalidan foydalanamiz. Shunday qilib, quyidagiga ega bo'lamiz:

```

<?
// bizning yozuvimiz
// o'zgarmas bo'lsin.
define("SIGN","Hurmat bilan, Azamat");
// odamlar va hodisalar massivini beramiz
$names = array("Ivan Ivanovich",
 "Petr Petrovich",
 "Semen Semenovich");
$events = array(
 "f" => "ochiq eshiklar kuni",
 "o" => "ko'rgazmaning ochilishi",
 "p" => "bitiruvchilar bali");

// taklifnoma matnnini tuzamiz.
$str = "Hurmatli, $names[0]";
$str .= "
 Sizni taklif etamiz
".
$events["f"];
$str .= "
" . SIGN; echo $str; //
matnni ekranga chiqaramiz.
?>

```

Shunday qilib, bu bo'limda biz PHP dasturlash tilining asosiy sintaksisi bilan tanishib chiqdik, turli tipdagi o'zgaruvchilar, o'garmaslar va amallar bilan ishlashni, PHP dasturlash tilidagi mavjud tiplarini o'rgandik. Massivlar va satrlar ma'lumot tiplari haqida gap ketganda ularni chuqur va qismlarga ajratib o'rgandik. Bu konstruktsiyalar foydalanishga qulay va soddadir. Bular haqida keng ma'lumotlar keyingi bo'limlarda keltirilgan. Masalaning echilishi bor bilimlarga asoslangan holda sodda echilgan, shuning uchun echim amaliyotda qo'llashga juda yaqin kelmaydi. Keyingi bo'limlarda bu kamchiliklarni to'g'rilaymiz va elektron maktubni umumi shablonini yaratamiz.

#### **4. Alternativ sintaksislar**

PHP dasturlash tili o'zining bir nechta if, while, for, foreach hamda switch boshqariladigan strukturalari uchun alternativ sintaksisni taqdim etadi. har bir holatda ochiladigan qavs ikki nuqtaga (:), yopiladigani esa mos ravishda endif;, endwhile; va hokazolarga o'zgartiriladi.

Masalan, if shart operatori sintaksisini quyidagicha tfodalash mumkin:

```
if (ifoda) : bajariladigan_blok endif;
```

Ma'nosi o'zgarmasdan qoladi: agar if shart operatori dumaloq qavsi ichidagi shart rost bo'lsa, ikki nuqtadan «::» to endif; buyruqigacha barcha kod bajariladi. Bunday sintaksisdan foydalanish html-kod ichida qurilgan php-kod uchun quaydir.

**Misol.** Alternativ sintaksisdan foydalanish.

```
<?php
$names = array("Karim","Salim","Sodi?");
if ($names[0]++"Karim"):
?>
```

Salom, Karim!

<?php endif ?>

Agarda else hamda elseif konstruktsiyalaridan foydalanilsa, u holda ham alternativ sintaksisdan foydalansa bo'ladi:

```
<?php
if ($a ++ 5):
 print "a o'zgaruvchi 5 ga
teng"; print "..."; elseif ($a
++ 6):
 print "a o'zgaruvchi 6 ga teng ";
 print "!!!";
else:
 print "a o'zgaruvchi na 5 ga va na 6 ga teng "; endif;
?>
```

#### **Nazorat savollari:**

1. PHP kodni tuzilishini tushuntirib bering.
2. PHP da qanday ma'lumot tiplaridan foydalaniadi?
3. PHP da ifodalar qanday e'lon qilinadi?
4. Alternativ sintaksislар deganda nimani tushunasiz?
5. Boshqa dasturlash tillari bilan PHP dasturlash tilining tilining farqini tushuntiring?
6. PHP tilining qanday afzalliliklari bor?

## 20-ma'ruza. PHPda obyektga yo'naltirilgan dasturlash. MVC texnologiyasi

### Reja:

1. PHP da sinflar va ob'yektlar.
2. MVC texnologiyasi haqida.
3. Model. View. Controller.

**Kalit so'zlar:** Funksiyalar, argumentlar, ro'yhatlar, o'zgaruvchilar, ichma-ich funksiyalar, klasslar va ob'yektlar, xatoliklar, o'zgaruvchilarni initsiallashtirish.

### 1. Funksiyalar

Funksiyalar nima uchun kerak? Bu savolga javob berish uchun, funksiya o'zi nima ekanligini tushunib olish lozim bo'ladi. Dasturlashda, xuddi matematikadagi kabi, unga bog'liq ko'pgina argumentlarning uning ko'pgina ma'nolarida aks etishidir. Demak, funksiya argumentning har bir ma'nolari jamlanmasi uchun uning bajargan ishi natijasi sifatida qandaydir ma'no qaytaradi. Funksiyalar nima uchun kerak, buni misollar bilan oydinlashtirishga qarakat qilamiz. Dasturlashdagi funksiyaga klassik misol – bu sonning faktorial ahaliyatini hisoblab beruvchi funksiya. Demak, biz unga son beramiz, u esa bizga uning faktorialini qaytaradi. Bunda biz faktorialini olishni xohlagan har bir son uchun aynan bir xil kodni qaytaravermaymiz – bu songa teng bo'lgan argumentli funksiyani chaqirishning o'zi kifoya qiladi.

#### Natural son faktorialini hisoblash funksiyasi

**Misol:** <?php function fact(\$n){ if (\$n++0) return  
1;  
else return \$fact = \$n \* fact(\$n-1);  
}  
echo fact(3);  
//echo (3\*2) deb yozish mumkin edi; lekin son katta bo'lsa, echo  
fact(50);  
//Funksiyadan foydalanish qulayroq, echo (50\*49\*48\*...\*3\*2) deb yozgandan;  
?>

Shu yo'l bilan biz biron-bir ma'lumotga bog'liqlik zaruriyati tug'ilgan amalni bajarganimizda, bu holda ham biz aynan shunday amallarni bajarishimiz oshirishimiz lozim bo'ladi, faqat boshqa boshlang'ich ma'lumotlardan foydalanamiz, funksiyalar mexanizmidan foydalanish-funksiya tanasi ko'rinishidagi amallar blokini taxt qilish, o'zgaruvchan ma'lumotlarni esa – uning parametrlari sifatida foydalanish qulayroq bo'ladi.

Funksiya topshirig'i (e'loni) umumiy tarzda qanday bo'lishini ko'ramiz. Funksiya quyidagi sintaksis yordamida aniqlanadi:

```
function Funksiya_nomi (1-parametr, 2-parametr, ... N-parametr) { Amallar bloki
 return "Funksiyaga aylanuvchi ma'no";
}
```

Agar php-dasturda to'g'ridan-to'g'ri yozilsa, hech narsani ishlab bo'lmaydi. Birinchidan, funksiya nomi funksiya parametrlari nomlari (1-parametr, 2-parametr va b.) PHP da nomlanish qoidalariga muvofiq kelishi kerak (unda yaxshisi kirillcha simvollarni ham ishlatmagan ma'qul). Funksiya nomlari registrga nisbatan sezuvchan bo'ladi. Ikkinchidan, funksiya

parametrlari – tilning o'zgaruvchan qismlari, shuning uchun ularning har birining nomlari oldidan \$ belgisi turishi lozim bo'ladi. Parametrlar ro'yxatida hech qanday ko'p nuqtalarini qo'yish mumkin emas. Uchinchidan, amallar bloki so'zi bilan birga funksiya tanasida istalgan to'g'ri PHP-kod mavjud bo'lishi kerak (parametrlarga muvofiq bo'lishi majburiy emas). Va nihoyat, return kalit so'zidan so'ng tartibli php-ifoda kelishi lozim (ma'noga ega bo'lgan qandaydir simvollar). Bundan tashqari, funksiyada qaytariluvchi ma'no kabi parametrlar bo'lmasligi ham mumkin. Funksiyani to'g'ri e'lon qilishga misol – yuqorida keltirilgan faktorialni hisoblash funksiyasi.

Funksiya chaqirish qanday amalga oshadi? Funksiya nomi va yumaloq qavslar ichida uning parametrlari ma'nolari ro'yxati ko'rsatiladi, agar shundaylari mavjud bo'lsa:

### Misol:

```
<?php
 Funksiya_nomi ("1-parametr_uchun_ma'no", "2-parametr_uchun_ma'no ",...);
/* Funksiyani chaqirishga misol – Funksiyani chaqirish faktorialni hisoblash Yuqorida bor, 3
soni faktorialini hisoblash uchun u erda biz fact(3) deb yozganmiz; u erda fact – chaqiriluvchi
Funksiya nomi, a 3 -$n nomli uning parametri ma'nosি *\ ?>
```

Funksiyani qachon chaqirish mumkin? Bu halati savol bo'lib tuyulishi mumkin. Funksiyani uni aniqlangandan keyin chaqirish mumkin, ya'ni function f\_name(){...} blokidan pastda istalgan dastur qatorida. PHP3 da bu aynan shunday. Lekin PHP4 da bunday talab yo'q. Hamma gap interpretator olingen kodni qanday qayta ishlashida. Birgina istisno shartli ravishda aniqlanadigan Funksiyadan tashkil topadi (shartli operatorlar yoki boshqa Funksiyalar ichida). Funksiya shu tarzda aniqlangan taqdirda, uni aniqlash uni chaqirishdan oldin bajariladi.

### Misol. Shartli funksiya ichida funksiyani aniqlash <?

```
$make = true;
/* bu erda Make_event() ni chaqirish mumkin emas; Chunki u hali mavjud emas, lekin
Save_info() ni chaqirish mumkin *\ */
Save_info("Sobir","Sodiqov", "Men PHP kursini tanladim"); if
($make){
//Make_event() Funksiyasini aniqlash function
Make_event(){ echo "<p> Python
 ni
o'rganmoqchiman";
}
}
//endi Make_event() ni chaqirish mumkin
Make_event();
//Save_info funksiyasini aniqlanadi function
Save_info($first, $last, $message){ echo
"
$message
";
echo "Ism: ". $first . " ". $last . "
";
}
Save_info("Murod","Yoqubov", "Men Lisp ni tanladim");
//Save_info ni bu erda ham chaqirish mumkin
?> Agar funksiya dastur ichida aniqlangan bo'lsa, uni keyin qayta aniqlash yoki o'chirib
tashlash mumkin emas. Funksiya nomlariga registr ta'sir qilmasligiga qaramasdan, yaxshisi
funksiyani aniqlash paytida berilgan nom bilan chaqirish mumkin bo'ladi.
```

**Misol.** Funksiya ichidagi funksiyani aniqlash

```
<?php
/* ma'lumotlarni saqlash, ya'ni DataSave() funksiyasini chaqirish mumkin emas.
Uning to'g'riliqi tekshirilmasdan oldin, ya'ni DataCheck() funksiyasi chaqirilmasdan oldin bu
mumkin emas.*\
DataCheck();
DataSave(); function
DataCheck(){
//ma'lumotlar to'g'riliqini tekshirish
 function DataSave(){
//ma'lumotlarni saqlaymiz } }
?>
```

Funksiya argumentlari, ularning ma'nolari va ishlatalishini batafsil ko'rib chiqamiz.

## 2. Funksiyalarning argumentlari

Har bir funksiyada, avval aytganimizday, argumentlar ro'yxati bo'lishi mumkin. Bu argumentlar yordamida funksiyaga har xil ma'lumotlar beriladi (masalan, faktoriali hisoblanishi kerak bo'lganson ma'nosi). Har bir argument o'zgaruvchi va konstantaga ega bo'ladi.

Argumentlar yordamida ma'lumotlar funksiyaga uch xil turli usullar bilan o'tkazilishi mumkin. Bu argumentlarni ma'nosiga ko'ra (o'zgarmas holatda foydalaniladi), ilovalarga ko'ra va o'zgarmas holatda argumentlarga ma'no berishga ko'ra o'tkazish. Bu usullarni atroflicha ko'rib chiqamiz. Argument funksiyaga ma'nosiga ko'ra o'tkazilsa, funksiya ichidagi argument ma'nosining o'zgarishi uning funksiya tashqarisidagi ma'nosiga ta'sir qilmaydi. Funksiyaga uning argumentlarini o'zgartirishga yo'l qo'yish uchun ularni havolalarga ko'ra o'tkazish kerak. Buning uchun argument nomi oldidan funksiyani aniqlashda ampensand “&” belgisini yozish kerak.

**Misol.** Argumentlarni havolasiga ko'ra o'tkazish

```
<?php
//qo'shimcha qilishi mumkin bo'lgan funksiyani yozamiz checked so'zi qatoriga function
add_label(&$data_str){
 $data_str .= "checked";
}
$str = "<input type=radio name=article ";
//bunday qator mavjud bo'lsin echo
$str .">>
";
//forma elementini keltiradi – belgilanmagan radio knopkasini add_label($str);
//Funksiyani chaqiramiz echo
$str .">>
";
//bu endi belgilangan radio knopkani keltiradi
?>
```

Funksiyada tinch holatda foydalanilayotgan argumentlar ma'nosini aniqlash mumkin. Ayni paytdagi ma'noning o'zi konstant ifoda bo'lishi, o'zgartirish va sinf vakili yoki boshqa funksiya chaqiruviga bo'lmasligi lozim.

Bizda informatsion xabar tuzuvchi funksiya, unga berilgan parametr ma'nosiga muvofiq tarzda o'zgaruvchi imzo bor. Agar parametr ma'nosiga berilmagan bo'lsa, "Tashkiliy qo'mita" imzosidan foydalaniladi.

**Misol.** Tinch holatdagi argument ma'nosiga

```
<?php
function Message($sign="Tash.qo'mita."){
//bu erda parametr sign ayni paytda "Tash.qo'mita" ma'nosiga ega echo
"Keyingi yig'ilish ertaga bo'lib o'tadi.
";
echo "$sign
";
}
Message();
//Parametrsiz Funksiyani chaqiramiz. Bu holda imzo – Bu Tashkiliy qo'mita
Message("Hurmat bilan Kamoliddin");
//Bu holda imzo "Hurmat bilan Kamoliddin." bo'ladi ?>
```

Bu skript ishining natijasi quyidagicha:

Keyingi yig'ilish ertaga bo'lib o'tadi.  
Tashkiliy qo'mita.  
Keyingi yiqilish ertaga bo'lib o'tadi. Hurmat  
bilan Kamoliddin.

Agar funksiyaning bir necha parametrlari bo'lsa, tinch holatda ma'no beriluvchi bu argumentlar funksiya aniqlanishida boshqa barcha argumentlardan keyin yozilishi kerak. Aks holda, agar bu argumentlar funksiyani chiqarish paytida ko'zdan qochirilsa xato yuzaga kelishi ehtimoli bor.

Masalan, biz katalogga maqola tavsifini kiritmoqchimiz. Foydalanuvchi maqolaga uning nomlanishi, muallifi va qisqa tavsif kabi xarakteristikalarini keltirishi lozim bo'ladi. Agar Foydalanuvchi maqola muallifi ismini kiritmadi, u Murod Yoqubov deb olaylik.

```
<?php
function Add_article($title, $description, $author="Murod Yoqubov"){ echo
"Maqolani katalogga kiritamiz: $title, "; echo "muallif $author"; echo "
qisqa
tavsif: "; echo "$description <hr>"; }
Add_article("Informatika va biz","Bu maqola informatikaga oid ...", "Zayniddin
Saidov");
Add_article("Xarakterlar kim", "Bu maqola xarakterlar haqida ..."); ?>
```

Skript ishi natijasi sifatida quyidagilarni olamiz:

Katalogga maqola kiritamiz:  
Informatika va biz, Muallif  
Murod Yoqubov.  
Qisqa tavsif:  
Bu maqola informatikaga oid...

Katalogga maqola kiritamiz: Xarakterlar  
kim,  
Muallif Odil Ziyaev.

Qisqa tavsif:

Bu maqola xarakterlar haqida...

Agar biz quyidagicha yozsak:

```
<?php
function Add_article($author="Odil Ziyaev", $title, $description){ //... avvalgi
misoldagi kabi amal
}
Add_article("Xarakterlar kim", "Bu maqola xarakterlar haqida..."); ?>
```

Natija quyidagicha bo'ladi:

```
Warning: Missing argument 3 for add_article() in
c:\users\nina\tasks\func\def_bad.php on line 2
```

### **3. O'zgaruvchan uzunlik argumentlari ro'yxatlari**

PHP4 da argumentlarning o'zgaruvchan soni bilan funksiya tuzish mumkin. Ya'ni biz uni necha argumentlar bilan chaqirilishini bilmasdan, funksiya tuzamiz. Bu kabi funksiya yozish uchun hech qanday maxsus sintaksis kerak bo'lmaydi. Hammasi uning ichiga o'rnatilgan funksiyalar func\_num\_args(), func\_get\_arg(), func\_get\_args() yordami bilan qilinadi.

**func\_num\_args()** funksiyasi argumentlar sonini qaytaradi. Bu funksiya faqat foydalanuvchi funksiyasini aniqlash mobaynida foydalanishi mumkin. Agar u funksiyadan tashqarida paydo bo'lsa, interpreterator ogohlantirish beradi.

**Misol.** func\_num\_args() funksiyasidan foydalanish

```
<?php function
DataCheck() { $n =
func_num_args();
echo "Funksiya argumentlari soni $n";
}
DataCheck();
//qatorni keltiradi "0 Funksiya argumentlari soni"
DataCheck(1,2,3);
//qatorni keltiradi "3-Funksiya argumentlari soni"
?>
```

**func\_get\_arg** funksiyasi (argument\_raqami to'laligicha) argumentni o'zgarishlar ro'yxatidan argumentlar funksiyasiga qaytaradi, uning tartib raqami func\_get\_arg parametri bilan beriladi. Funksiya argumentlari noldan boshlab hisoblanadi. func\_num\_args() kabi bu funksiya faqat biron-bir funksiyani aniqlashda foydalaniladi.

Argument raqami funksiyaga o'zgargan argumentlar sonidan ortib ketishi mumkin emas. Aks holda ogohlantirish umumlashtiriladi va func\_num\_args() Funksiyasiga False qiymat qaytadi.

Ma'lumotlarni tekshirish uchun funksiyaga uning argumentlarini tuzamiz. Agar funksiyaning birinchi argumenti – butun son, ikkinchisi – qator bo'lsa, tekshirish muvaffaqiyatli o'tdi, deb hisoblaymiz.

**Misol.** Ma'lumotlar tipini, uning argumentlarini tekshirish <?

```

function DataCheck(){ $check =true;
 $n = func_num_args();
/* Funksiyaga o'zgargan argumentlar sonini tekshiramiz, birinchi o'zgargan argument butun
sonmiyo'qmi */ if ($n>=1) if (!is_int(func_get_arg(0))) $check = false;
/* tekshiramiz, ikkinchi o'zgargan argument qatormiyo'qmi */ if
($n>=2)
 if (!is_string(func_get_arg(1)))
 $check = false; return
$check;
}
if (DataCheck(123,"text")) echo
"Tekshiruv to'g'ri o'tdi
";
else echo "ma'lumotlar shartlarni qondirmaydi
"; if
(DataCheck(324))
echo "Tekshiruv to'g'ri o'tdi
";
else echo "ma'lumotlar shartlarni qondirmaydi
";
?>
```

Dastur natijasi quyidagicha bo'ladi.

Ma'lumotlar shartlarni qoniqtirmaydi. Tekshiruv to'g'ri o'tdi.

**func\_get\_args()** funksiyasi argumentlar ro'yxatidan tashkil topgan massiv qaytaradi. Massivning har bir elementi argumentga, funksiya o'zgarishiga to'g'ri keladi. Agar funksiya foydalanuvchi funksiyasi aniqligidan tashqarida foydalanilsa ogohlantirish umumlashtiriladi.

Avvalgi misolni ko'chiramiz, bu funksiyadan foydalanamiz.

Funksiyani harakatlantiruvchi juft argument butun son ekanligini tekshiramiz:

### **Misol:**

```

<? function
DataCheck(){ $check
=true;
 $n = func_num_args();
//Funksiyaga o'zgargan argumentlar soni
$args = func_get_args();
//Funksiya argumentlari massivi
for ($i=0;$i<$n;$i++){ $v =
$args[$i]; if ($i % 2 ++ 0){
 if (!is_int($v)) $check = false;
//tekshiramiz, juft argument butunmiyo'qmi
}
}
return $check;
} if (DataCheck(array("text", 324)))
echo "Tekshiruv to'g'ri o'tdi
";
else echo "Ma'lumotlar shartlarni qoniqtirmaydi
";
?>
```

Bundan, func\_num\_args(), func\_get\_arg() va func\_get\_args() funksiya kombinatsiyasi funksiyalar o'zgaruvchan argumentlar ro'yxatiga ega bo'la olishi uchun foydalaniladi. Bu funksiyalar faqat PHP4 ga kiritilgan. PHP3 da shunday natijaga erishish uchun, argument sifatida massiv funksiyasidan foydalanish mumkin bo'ladi. Masalan, har bir toq funksiyalar parametri butun sonligini tekshiruvchi skriptni quyidagicha yozish mumkin:

```
<?
function DataCheck($params){
 $check =true;
 $n = count($params);
 //Funksiyaga o'zgargan argumentlar soni
 for ($i=0;$i<$n;$i++){ $v = $params[$i];
 if ($i % 2 != 0){
 //tekshiramiz, toq argument butunmi-yo'qmi
 if (!is_int($v)) $check = false;
 }
 }
 return $check;
}
if (DataCheck("text", 324)) echo
"Tekshiruv to'g'ri o'tdi
";
else echo "Ma'lumotlar shartlarni qoniqtirmaydi
"; ?>
```

#### 4. Funksiyalar ichida o'zgaruvchilardan foydalanish

Quyidagi o'zgaruvchilarning turlaridan foydalanish mumkin:

1. Global o'zgaruvchilar;
2. Statistik o'zgaruvchilar;
3. Qaytariluvchan ma'nolar;
4. Havolani qaytarish;

#### ***Global o'zgaruvchilar***

Funksiyalar ichida undan tashqarida berilgan o'zgaruvchilardan foydalanish uchun, bu o'zgaruvchilarni global deb e'lon qilish kerak. Buning uchun funksiya tanasida uning nomlarini global kalit so'zidan keyin keltirish lozim bo'ladi:

#### **Misol.** Global o'zgaruvchilar

```
global $var1, $var2; <? $a=1;
function Test_g(){ global $a;
$a = $a*2;
echo '$a=',$a Funksiya ishida natija;
}
echo 'Funksiyadan tashqarida $a=',$a,' ';
Test_g(); echo "
";
echo Funksiyadan tashqarida $a=',$a,' ';
Test_g();
?>
```

Bu skript ishidan quyidagi natijalarni olamiz:

\$a=2 Funksiyadan tashqarida, \$=2 Funksiya ishida natija  
\$a=2 Funksiyadan tashqarida, \$=4 Funksiya ishida natija  
O'zgaruvchi global deb e'lon qilinganda, aniq global o'zgaruvchi uchun havola tuziladi. Buning uchun bunday yozuv quyidagiga ekvivalent (GLOBALS massivi mavjud ko'rinish sohalariga muvofiq barcha global o'zgaruvchilarni o'z ichiga oladi):

```
$var1 = & $GLOBALS["var1"];
$var2 = & $GLOBALS["var2"];
```

Bundan kelib chiqadiki, \$var1 o'zgaruvchini o'chirish \$\_GLOBALS["var1"] global o'zgaruvchiini o'chirib tashlamaydi.

### ***Statistik o'zgaruvchilar***

O'zgaruvchilardan faqat funksiya ichida foydalanish uchun bunda uning ma'nosini saqlagan holda va funksiyadan chiqqandan so'ng, bu o'zgaruvchilarni statistik deb e'lon qilish kerak. Statistik o'zgaruvchilar faqat funksiyalar ichida ko'rindi va dasturni yuklash funksiya doirasidan tashqariga chiqsa o'z ma'nosini yo'qotmaydi. Bu o'zgaruvchilarni e'lon qilish static kalit so'zi yordamida amalga oshiriladi:

```
static $var1, $var2;
```

Har qanday ma'no statistik o'zgaruvchi sifatida talqin qilinishi mumkin, faqat havola emas.

**Misol.** Statistik o'zgaruvchilardan foydalanish <?

```
function Test_s(){ static $a = 1; //ifoda yoki
havolani o'zlashtirib bo'lmaydi
 $a = $a*2;
echo $a;
}
```

Test\_s(); //2 chiqadi echo \$a; //hech narsa chiqmaydi, zero \$a faqat funksiya ichida kirish yo'lagi bor Test\_s(); // \$a=2 Funksiya ichida, shuning uchun funksiya ishi natijasi 4 soni bo'ladi ?>

### ***Qaytariluvchan ma'nolar***

Yuqorida misol qilib keltirilgan barcha funksiyalar biror-bir amal bajarishgan.

Bunday hollardan tashqari, har qanday funksiya o'z ishi natijasi sifatida qandaydir qiymat qaytaradi. Bu return tasdig'i yordamida qilinadi. qaytariluvchan qiymat har qanday turda, shu jumladan, ro'yxat va ob'ektlar bo'lishi mumkin. Interpretator funksiya tanasida return komandasiga uchraganda, u darhol uni bajarishni to'xtatadi va funksiya chaqirilgan qatorga o'tib ketadi.

Masalan, inson yoshini qaytaruvchi funksiya tuzamiz. Agar inson vafot etmagan bo'lsa, yosh joriy yilga muvofiq hisoblanadi.

```
<?php
/* agar ikkinchi parametr true kabi hisoblab chi?ilsa, u vafot etgan sanaday ko'rib chiqiladi, */
function Age($birth, $is_dead){ if ($is_dead) return is_dead-birth; else return date("Y")-$birth;
} echo Age(1971, false); //vo`vedet 33
echo Age(1971, 2001); //vo`vedet 30
?>
```

Bu misolda return funksiyasidan foydalanmasa ham bo'ladi, shunchaki uni chiqarish funksiyasini echo ga almashtiriladi. Aksincha, agar biz funksiya biror-bir qiymat qaytaradigan qilsak (bu misolda inson yoshi), biz dasturda o'zgaruvchini bu funksiya qiymatini istalgan o'zgaruvchiga o'zlashtirishimiz mumkin.

```
$my_age = Age(1981, 2004);
```

Funksiya ishi natijasida faqat bitta qiymat qaytarilishi mumkin. Bir necha qiymatni qiymatlar ro'yxati qaytarilgan taqdirda olish mumkin (bir o'lchamli massiv). Biz inson yoshini kunigacha aniqlikda olmoqchimiz, deb hisoblaylik.

```
<?php
function Full_age($b_day, $b_month, $b_year)
{
 $y = date("Y");
 $m = intval(date("m"));
 $d = intval(date("d"));
 $b_month = intval($b_month);
 $b_day = intval($b_day);
 $b_year = intval($b_year);

 $day = ($b_day > $d ? 30 - $b_day + $d : $d - $b_day);
 $tmpMonth = ($b_day > $d ? -1 : 0);
 $month = ($b_month > $m + $tmpMonth
 ? $b_month + $tmpMonth - $m : $m+$tmpMonth - $b_month);
 $tmpYear = ($b_month > $m + $tmpMonth ? -1 : 0); if
 ($b_year > $y + $tmpYear)
 {
 $year = 0; $month = 0; $day = 0;
 }
else
 {
 $year = $y + $tmpYear - $b_year;
 }
return array ($day,$month,$year);
}
$age = Full_age("29","06","1986");
echo "Siz $age[2] yosh, $age[1] oylar va $age[0] kunlar";
?>
```

Funksiya bir necha qiymatlarni ularni dasturda qayta ishlash uchun qaytarganda, bir amal bilan ma'noni birdaniga bir necha o'zgaruvchilarni o'zlashtirishga imkon beruvchi list() til konstruktsiyasidan foydalanish qulay bo'ladi. Masalan, yuqorida misolda funksiyani, uning qiymatiga o'zgartirish kiritmay qayta ishlash quyidagicha bo'lishi mumkin:

```
<?
//Full_age() Funksiya kiritish
list($day,$month,$year) = Full_age("07",
 "08","1974");
```

```

echo "Sizning yoshingiz $year, $month oy va
 $day kun";
?>
list() konstruktsiyasini umuman o'zgaruvchini o'zlashtirish uchun istalgan massiv
elementi qiymatidan foydalanish mumkin.

```

**Misol.** list() dan foydalanish <?

```

$arr = array("first","second"); list($a,$b) =
$arr;
//o'zgaruvchi $a o'zlashtiriladi, birinchi massiv qiymati, $b – ikkinchi echo $a," ",$b;
//«first second» qatori keltiriladi
?>

```

**Havolani qaytarish**

Funksiya o'z ishi natijasida shuningdek havolani biror-bir o'zgaruvchiga qaytarishi mumkin. Bu funksiyani qanday o'zgaruvchi havolaga o'zlashtirish kerakligini aniqlash uchun foydalaniladi. Funksiyadan havola olish uchun, e'lon oldidan ampersand (&) belgisini yozish kerak bo'ladi va har safar funksiya chaqiruvi paytida uning nomi oldidan ham ampersand (&) yozish kerak bo'ladi. Ko'pincha funksiya havolani biror-bir global o'zgaruvchiga (yoki uning qismini – havolani global massiv elementiga), havolani statistik o'zgaruvchiga (yoki uning qismini) yoki havolani argumentlardan biriga qaytaradi, agar u havola bo'yicha berilgan bo'lsa.

**Misol.** Havolani qaytarish <?

```

$a = 3; $b = 2; function
& ref($par){ global $a,
$b;
if ($par % 2 == 0) return $b;
else return $a; }
$var =& ref(4);
echo $var, " i ", $b, "
"; //2 va 2 keltiriladi
$b = 10;
echo $var, " i ", $b, "
"; //10 va 10 keltiriladi ?>

```

Havola sintaksisidan foydalanishda bizning misoldagi \$var o'zgaruvchi o'zgaruvchining \$b qiymati \$ref qaytarilgan funksiyasiga ko'chirilmaydi, bu o'zgaruvchiga havola tuziladi. Demak, endi \$var va \$b teng kuchli o'zgaruvchilar va ular bir paytda o'zgartiriladi.

## 5. Funksiyaning o'zgaruvchilari

PHP funksiyalar o'zgaruvchilariga ko'maklashadi. Bu degani, agar o'zgaruvchi nomi oddiy qavslar bilan tugasa, PHP shu kabi nomli funksiyani qidiradi va uni bajarishga harakat qiladi.

**Misol.** Funksiyalar o'zgaruvchilaridan foydalanish <?

```

/* Ikkita oddiy funksiya tuzamiz: Add_sign – qatorga imzo qo'shadi va Show_text
-matn qatorini chiqarib beradi*/
function Add_sign($string,
$sign="Hurmat bilan, Murod"){
 echo $string . " ".$sign;
}
function Show_text(){ echo "Xabarni
pochtadan jo'natish
"; }

```

```

$func = "Show_text"; //ma'noga ega o'zgaruvchi tuzamiz, u funksiya nomiga teng
Show_text
$func(); //u Show_text funksiyani chaqiradi
$func = "Add_sign"; //ma'noga ega o'zgaruvchi tuzamiz, u funksiya nomiga teng
Add_sign
$func("hammaga salom
");

//bu funksiyani chaqiradi Add_sign "hammaga salom" parametrli ?>
Bu misolda Show text Funksiya shunchaki matn qatorini chiqaradi. Agar echo maxsus

funksiyasi mavjud bo'lsa, nega buning uchun alohida Funksiya tuzish kerak, deyish mumkin.

Gap shundaki, echo(), print(), unset(), include() kabi funksiyalardan funksiyalar

o'zgaruvchilari sifatida foydalanib bo'lmaydi. Ya'ni biz yozsak: <?
$func = "echo "; $func("TEXT");
?>
Interpretator xatoni ko'rsatadi:

Fatal error: Call to undefined function: echo() in

c:\users\nina\tasks\func\var_f.php on line 2
Shuning uchun yuqorida keltirib o'tilgan istalgan funksiyalardan o'zgaruvchilar funksiyasi

sifatida foydalanish uchun yuqoridagi misoldagi yo'lni tutdik. 6. Ichki joylashgan (ichma-ich) funksiyalar

Foydalanuvchi tomonidan aniqlanadigan funksiyalar haqida gapirganda ichki joylashgan

funksiyalar haqida gap ketmasligi mumkin emas. Yuqorida biz echo(), print(), date(), include()

kabi ichki joylashgan funksiyalar bilan tanishdik. Bundan tashqari date() funksiyadan boshqa

barcha funksiyalar PHP dasturlash tili konstruktsiyasiga ega. Ular PHP dasturlash tili

yadrosiga joylashgan bo'lib, hech qanday modullar va qo'shimcha o'zgartirishlar talab

etmaydi. Ammo shunday funksiyalar mavjudki, ularga turli fayl bibliotekalari va mos ravishda

modullarni yuklamasdan iloji yo'q. Masalan, MySql ma'lumotlar bazasi bilan ishlaydigan

funksiyalardan foydalanish uchun shunday kengaytmali fayllarni qo'llab quvvatlaydigan

komponentalari kerak. Oxirgi vaqtarda bu funksiyalardan foydalanish uchun qo'shimcha

komponentalar kerak emas, chunki ularning barchasi Hozirda PHP dasturlash tili yadrosiga

kiritilgan.
```

## 7. Sinflar va ob'ektlar

Ob'ektga yo'naltirilgan dasturlashning asosiy tushunchalari – sinflar hamda ob'ektlardir. Bu  
tushunchalarni quyidagicha tushunish mumkin: ob'ekt – bu dasturda qo'llaniladigan tushuncha  
yoki biror fizik predmet haqida ma'lumot beradigan strukturalangan o'zgaruvchidir, sinflar esa  
bu ob'ektlarning tavsifi va ular ustida bajariladigan qarakatlardir.

PHP dasturlash tilida sinflar quyidagi sintaksis yordamida aniqlanadi: class  
Sinf\_nomi{ var \$xususiyat\_nomi; /\* xususiyatlar ro'yxati \*/ function  
metod\_nomi(){  
/\* usullarning tanasi \*/ }  
/\*usullar ro'yxati\*/ }

Sinf ob'ektlari xususiyatlari nomi var kalit so'zi yordamida e'lon qilinadi, berilgan sinf  
ob'ektlariga qo'llanilgan usullar Funksiya sifatida ishlataladi. Sinf tanasi ichida this kalit so'zi  
yordamida taqdim qilinayotgan joriy sinfga murojaatni amalgalash mumkin.

Masalan, biz maqola kategoriyasini tasvirlovchi sinf tuzishimiz kerak. har bir maqolaning  
nomi, muallifi va qisqa mazmuni kabi xususiyatlari bor. Biz maqola bilan qanday amal  
bajarmoqchimiz? Biz sanab o'tilgan xususiyatlarga ma'no berishimiz, maqolani brauzerda  
ko'rsatishimiz kerak bo'ladi. Shunda bu sinfning ifodalanishi quyidagicha holatda bo'ladi: <?  
class Articles { //Maqola sinfini tuzamiz var \$title; var \$author; var \$description;

```

//maqola atributi ma'nosini o'zlashtiruvchi usul function
make_article($t, $a, $d){
 $this->title = $t;
 $this->author = $a;
 $this->description = $d; }
//sinf nusxasini ifodalash uchun usul
function show_article(){ $art = $this-
>title . "
" .
 $this->description .
 "
Muallif: " . $this->author;
 echo $art; }
} ?>

```

Shunday qilib “maqola” turidagi fizik ob'ektlarni tasvirlash uchun biz uch o'zgartuvchidan tashkil topgan, maqola xarakteristikasini o'zida jamlagan Articles nomli sinf va muayyan maqola tuzish va uni tasvirlash uchun ikkita funksiya tuzdik.

Ma'lumki, PHP bilan ishlash davriy holatda HTML rejimida yuklanishi mumkin. Bu holda dastur bir necha kodning bo'laklari(bloklar)dan tashkil topadi. Sinfni ifodalash php-kodning har xil bloklari bo'yicha va qolaversa har xil fayllar bo'yicha tar?atilmasligi kerak. Ya'ni quyidagicha yozsak:

```

<?php
class Articles { //Sinfni tasvirlashning boshi var
$title;
?>
<?php //sinfni tasvirlashning davomi function
show_article(){ //usulning tarkibi
}
} //sinfni tasvirlashning yakuni
?> bunda dastur tartibli
ishlaydi.

```

Sinf nomi masalasida ayrim narsalarni e'tiborda tutish kerak. Sinfning nomi PHP tilidagi ob'ektlar nomlanishi qoidalariga javob berishi lozim, lekin bir qator nomlar borki, texnik mutaxassislar tomonidan o'z maqsadi uchun zaxira qilinadi. Birinchi navbatda bu nomlar “\_” quyi chiziqdandan boshlanuvchilardir. Sinflar va funksiyalar tuzish uchun bu kabi nomlarni ishlatmaslik kerak. Bundan tashqari stdClass nomi zaxira qilingan, zero u PHP surilgichi ichida ishlatalidi.

## **8. O'zgaruvchilarni initsiallashtirish**

Ba'zan ayrim sinf atributlariga ma'noni sinf ishtirokchisini tuzish bilanoq o'zlashtirish kerak bo'ladi. Biz maqola sinfini tuzganimizda, sinf atributlari (xususiyatlari) ma'nolarini o'zlashtirish uchun maxsus funksiya make\_article() dan foydalandik. Umuman olganda, biz to'g'ri yo'l tutmadik, chunki “velosiped ixtirosi” bilan shug'ullanndik. Sinf atributlarining boshlang'ich ma'nolarini berish uchun maxsus ikkita standart usul mavjud. PHP4da ma'noni var operatori yoki konstruktor funksiyasi yordamida initsiallashtirish mumkin. var yordamida faqat konstant ma'nolarni initsiallashtirish mumkin. Konstant bo'limgan ma'nolarni berish uchun ob'ekt sinfdan ajrab chiqqanda o'z-o'zidan ishga tushuvchi konstruktor funtsiyasidan foydalaniladi. Konstruktor-funksiya u ifodalangan butun sinfga mos keluvchi nomga ega bo'lishi kerak.

Misol. “maqola” deb nomlangan ob'ektni tuzishda uning xususiyatlarini quyidagicha belgilash mumkin: mualliflar – “Kamolov” satriga teng, nomlanish va qisqa mazmun - \$\_POST global massivi elementlariga mos, maqola nashri – mazkur sanada.

```

<?
class Articles { //maqola sinfini tuzish var
$title;
 var $author = "Kamolov"; var $description; var $published; //sinf
atributi ma'nosini o'zlashtiruvchi usul function Articles(){
 $this->title = $_POST["title"];
 $this->description = $_POST["description"];
 $this->published = date("Y-m-d");
} } ?>

```

PHP3 va PHP4 da konstrukturlar har xil ishlashini hisobga olish kerak. Funksiya PHP3 da, agar u sinfniki kabi nomga ega bo'lsa, konsrtuktorga aylanadi, PHP4 da esa – agar u ifodalangan sinfniki kabi nomga ega bo'lsa shunday bo'ladi. Bir sinf boshqasini kengaytirganda va xususiyatlarning va baza sinflar usullarining ergashishida usullar orasidagi farq ko'rinish turibdi. Lekin biz bu haqida biroz keyinroq gapiramiz. PHP5da sinf konstruktori `_construct` deb nomlanadi. Bundan tashqari, PHPda destruktorlar – ob'ektini yo'q qilishda o'z-o'zidan ishga tushuvchi Funksiyalar paydo bo'ldi. PHP5 da Funksiya-destruktor `destruct` deb nomlanishi kerak bo'ladi.

## 9. Ob'yektlar

### *Object (ob'ektlar) tipi*

Ob'ektlar – ob'ektga yo'naltirilgan dasturlashdan kirib kelgan ma'lumot tipidir. Ob'ektga yo'naltirilgan dasturlash tamoyiliga ko'ra, sinf – aniq xossalarga ega va ular bilan ishlaydigan metodli ob'ektlar to'plami. Ob'ekt esa mos ravishda sinf nusxasidir. Masalan, dasturchilar – bu dasturni tuzuvchi, kompyuter adapiyotlarini o'rganadigan odamlar sinfi va bundan tashqari hamma odamlar qatori ism va familiyasi mavjud. Endi agarda biror aniq dasturchi – Azamat Boboevni olsak, u holda uni shu xossaga ega bo'lgan dasturchi sinfini ob'ekti sifatida qarash mumkin va u ham dastur tuzadi, hamda ismi mavjud va boshqalar.

PHP dasturlash tilida ob'ekt metodiga murojaat -> amalidan foydalilanadi. Ob'ektni initsializatsiya qilishda ob'ektni o'zgaruvchan nusxasini yaratadigan `new` ifodasidan foydalilanadi.

```

<?php //odam sinfini yaratamiz. class
Person
{
 //PHP dasturlash tilini o'rganadigan odam metodi function
know_php() {
 echo "Endi men PHP dasturlash tilini bilaman!";
}
}
$bob = new Person; //odam sinfini ob'ektini yaratamiz.
$bob -> know_php(); //uni PHP tiliga o'rgatamiz.
?>

```

Sinf – bu ob'ekt tipidagi ma'lumotlarning bir turidagi ifodalanishidir. Sinflar real o'zgaruvchilar uchun shablon vazifasini o'taydi. Kerakli tipdagi o'zgartuvchi `new` operatori yordamida sinfdan tuziladi. Ob'ektni tuzib, biz barcha usullarni qo'llashimiz va barcha sinf ifodasida ko'rsatib o'tilgan xususiyatlarni olishimiz mumkin bo'ladi. Buning uchun quyidagicha sintaksisdan foydalilanadi: `$ob'ekt_nomi->xususiyat yoki _nomi$ob'ekt_nomi->usuuning_nomlanishi(argumentlar ro'yxati)`. Xususiyatlar yoki usular nomlari oldidan \$ belgisi qo'yilmaydi.

**Misol:** Ob'ekt usullari va xususiyatlariga erkin kirish (dostup)

```

<?php
$art = new Articles; //ob'ekt tuzamiz $art

```

```

echo ($art ->title); //ob'ekga nomlanish beramiz $art $another_art = new
Articles; //ob'ekt tuzamiz $another_art
$another_art->show_article(); //ob'ektning brauzerdagi ifodasi uchun usulni chaqiramiz ?>
Sinfning har bir ob'ekti aynan bir xil xususiyatlar va usullarga ega bo'ladi. Demak, $art
ob'ektda va $another_art ob'ektda title, description, author xususiyatlari va Articles(),
show_article() usullari mavjud. Lekin bular ikki xil ob'ektlar. Ob'ektni fayllar sistemasidagi
direktoriya deb hisoblaymiz, uning xarakteristikasi esa – bu direktoriyadagi fayllar singari
bo'lsin. Aniqki, har bir direktoriyada bir xil fayllar yotishi mumkin, lekin shunday bo'lsa-da,
ular har xil direktoriyalarda saqlanayotgani uchun har xil hisoblanishi mumkin. Xuddi
shuningdek, xususiyatlar va usullar ham, agar ular turli ob'ektlarga qo'llaniladigan bo'lsa, har
xil hisoblanadi. Yuqori bosqichdagi direktoriyadan kerakli faylni olish uchun bu faylga yo'lni
batafsil yozib chiqamiz. Sinflar bilan ishlash mobaynida biz chaqirishni istagan Funksiyaning
nomini to'liq yozishimiz kerak bo'ladi. PHP dagi Yuqori bosqich direktoriyalariga global
o'zgaruvchilarning bo'sh o'rni bo'ladi, yo'l esa -> taqsimlovchisi yordamida ko'rsatiladi. Shu
tarzda $art->title va $another_art>title nomlari ikki xil turli o'zgaruvchilarni anglatadi. PHP da
o'zgaruvchi nom oldidan faqat bitta dollar belgisiga ega bo'ladi, shuning uchun $art->$title
ko'rinishida yozish mumkin emas. Bu konstruktsiya $art ob'ektining title xususiyatiga
murojaat sifatida ko'rib chiqilmaydi, $title o'zgartuvchi ko'rinishida berilgan nomli xususiyat
sifatida ko'rildi (masalan, $art->"").
```

**Misol:** Xususiyatlar ma'nosini o'rnatish

```
<?php
$art->title = " Internet ga kirish"; //ob'ekt xususiyati ma'nosini shunday o'rnatish mumkin
$art->$title = "Internet ga kirish"; //ob'ekt xususiyati ma'nosini bunday o'rnatib bo'lmaydi
$property = "title";
$art->$property = "Internet ga kirish"; //ob'ekt xususiyati ma'nosini shunday o'rnatish mumkin
?>
```

Sinfni tuzib, bu sinfning ob'ekti qanday nomga ega bo'lishini bila olmaymiz, qolaversa
ob'ektlar juda ko'p bo'lishi va ularning barchasi har xil nomga ega bo'lishi mumkin. Sinfni
yuzaga chiqarish ichida ob'ektga qanday munosabatda bo'lishni bilmaymiz. Sinf yuzaga
chiqishi ichida funksiyalar va o'zgaruvchilarga erkin kirish uchun, \$this o'rindosh
o'zgaruvchisidan foydalanish kerak. Masalan, \$this->title shunday sind ob'ektining title ini
qaytaradi. Ba'zan bu o'zgaruvchini “mening xususiy mulkim” (xususiyatga munosabat
tariqasida) deb o'qishni taklif qilinadi.

### Nazorat savollari:

1. PHP dasturlash tilida funksiyalarni tushuntirib bering.
2. PHP da funksiyalar argumentlari qanday e'lon qilinadi? 3.  
PHP da o'zgaruvchilar e'lonini tushuntiring.
4. Ichma-ich funksiyalar nima va ularning ahamiyati haqida gapiring?
5. PHP dasturlash tilida sinflar va ob'ektlar nima?
6. Funksiyani chaqirish qanday amalga oshiriladi?
7. PHP adsturlash tilida funksiyalar nima uchun kerak? 8. Funksiyalar ichida qanday
turdag'i o'zgaruvchilardan foydalanish mumkin?
9. PHP dasturlash tilida klasslar va ob'yektlar qanday e'lon qilinadi?
10. Sinfning o'byektlari qanday hususiyatlarga ega bo'lishi mumkin?
11. PHP dasturlash tili konstruktsiyasiga ega bo'lмаган funksiyani ayting va u haqida
ma'lumot bering?

## **21-22-ma'ruza. PHPda MySQL ma'lumotlar bazasini boshqarish tizimi bilan ishlash.**

### **Reja:**

1. PHPda ma'lumotlar bazasi bilan ishlash (CRUD) amallari.
2. MySQL ma'lumotlar bazasini boshqarish tizimi.
3. Jadval yaratish. Jadvaldagi ma'lumotlarni o'qish, o'zgartirish, o'chirish.
4. PHP da AJAX texnologiyasi bilan ishlash.

**Kalit so'zlar:** Ma'lumotlar bazasi, MySql dasturi, PHP da MySql bilan bog'lanish, fayllar, fayllar bilan ishlash, MB amallari.

### **1. Ma'lumotlar bazasi haqida tushuncha**

Ushbu bo'lim PHP va MySql MBBS o'rtasidagi hamkorlik usullari bilan tanishishga mo'ljallangan. Asosiy e'tibor ma'lumotlar bazasi bilan bog'lanishni o'rnatish, so'rovlar jo'natish funksiyalari va javoblarni (mysql\_connect, mysql\_query, mysql\_result, mysql\_num\_rows, mysql\_close) qayta ishlashga qaratiladi.

Misol sifatida virtual tarix muzeyi ma'muriyati uchun web-interfeys tuzish masalasini ko'raylik. PHP distributivida MySQL ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan Funksiyalar mavjud. Bunda bu funksiyalarning MySql dagi ba'zi bir ma'lumotlar bazasini tasvirlash va to'ldirish maqsadida webinterfeyslarni tuzish imkonini beruvchi funksiyalar bilan tanishamiz. Ma'lumotlar bazasiga ma'lumotlarni qo'shish uchun web-interfeys bilan ishlashda bu ma'lumotlarni shunchaki html-formaga kiritish va ularni serverga jo'natish kerak bo'ladi.

Namoyish etishda bu interfeysni virtual muzey eksponatlari haqidagi ma'lumotlar saqlanadigan

Artifacts jadvallari uchun tuzamiz. Artifacts kolleksiyasidagi har bir eksponat quyidagi xarakteristika yordamida tasvirlanishini eslatib o'tamiz:

- nom (title);
- muallif (author);
- ifoda (description);
- o'rindosh nom (alternative);
- tasvir (photo).

Nomlanish va o'rindosh nomlanish uzunasiga 255 belgidan kam satr (ya'ni VARCHAR(255)), tasvirlash – matnli maydon (TEXT turiga mansub) hisoblanadi, "muallif" va "tasvir" maydonlarida esa Persons kolleksiyasidan muallifning identifikatorlari va Images kolleksiyasidan eksponat tasvirlariga muvofiq mavjud bo'ladi.

### **2. Ma'lumotlar bazasi interfeysi**

Ma'lumotlar bazasidagi mavjud jadval strukturasini (ya'ni uning maydonlari jamlanmasini) html-formada tasvirlash uchun quyidagi tarkibiy topshiriqlarni rejalashtirish mumkin:

- MB bilan ulanishni o'rnatish; □ MB ishini tanlash;

- Jadval maydonlari ro'yxatini olish;
- html-formada maydonlarni tasvirlash.

Bundan keyin formaga kiritilgan ma'lumotlarni ma'lumotlar bazasiga kiritish mumkin.

### **3. Ma'lumotlar bazasi bilan bog'lanish (MySQL dasturi misolida)**

#### *Aloqa o'rnatish*

Ma'lumotlar bazasi bilan aloqa o'rnatish uchun **mysql\_connect** funksiyasidan foydalaniladi. mysql\_connect sintaksisi mysql\_connect resursi ("server qatori", "username", "password")

Bu Funksiya MySQL serveri bilan aloqa o'rnatadi va bu aloqaga ko'rsatkich qaytaradi yoki muvaffaqiyatsiz chiqqanda FALSE ko'rsatadi. Odatda quyidagi parametrlar qiymati e'lon qilinadi:

```
server = 'localhost:3306'
username = server jarayoni egasidan foydalanuvchi ismi
password = bo'sh parol
Server bilan ulanish, agar u bungacha mysql_close() yordamida yopilmagan bo'lsa, skriptni amalga oshirish tugallanishida baza bilan aloqa yopiladi.
```

#### **Misol:**

```
<?
$conn = mysql_connect("localhost", "nina","123") or die("Ulanishni
amalga oshirib bo'lmaydi: ". mysql_error()); echo "Ulanish amalga
oshdi"; mysql_close($conn);
?>
mysql_connect amali shell>mysql -u nina -
p123 buyrug'i bilan teng kuchli.
```

### **4. Ma'lumotlar bazasi ustida bajariladigan amallar**

#### *Ma'lumotlar bazalarini tanlash*

MySQL da ma'lumotlar bazasini tanlash use buyrug'i yordamida amalga oshiriladi:  
mysql>use book;

PHP da buning uchun mysql\_select\_db funksiyasi mavjud.

mysql\_select\_db: sintaksisi  
mantiqiy mysql\_select\_db (database\_name qatori);

Bu funksiya TRUE qiymatni ma'lumotlar bazasini muvaffaqiyatlama tanlanganda qaytaradi va FALSE ni esa – aksincha bo'lganda.

**Misol:** Book ma'lumotlar bazasini tanlash

```
<?
$conn = mysql_connect("localhost","user","123") or die("Ulanishni
amalga oshirib bo'lmaydi: ".mysql_error()); echo "Ulanish amalg
oshdi"; mysql_select_db("book");
?>
```

### **Jadval maydonlari ro'yxatini olish**

PHP da ma'lumotlar bazasi bilan bog'lanilgandan so'ng, undagi jadvallar ro'yxatini olish mumkin. Bu funksiya - mysql\_list\_fields.

mysql\_list\_fields sintaksisi

```
mysql_list_fields (database_name qatori, table_name qatori)
```

mysql\_field\_name Funksiyasi so'rov amalga oshirilishi natijasida olingan maydon nomini qaytaradi.

mysql\_field\_len      funkisiyasi      maydon      uzunligini      qaytaradi.      mysql\_field\_type Funksiyasi      maydon      tipini qaytaradi, mysql\_field\_flags Funksiyasi esa probel bilan yozilgan maydon bayroqlari ro'yxatini qaytaradi. Maydon tiplari int, real, string, blob va h. bo'lishi mumkin. Bayro?lar not\_null, primary\_key, unique\_key, blob, auto\_increment va h. bo'lishi mumkin.

Bu barcha buyruqlar sintaksisi bir xil:

```
mysql_field_name (result qatori, butun field_offset)
resursi; mysql_field_type (result qatori, butun
field_offset) resursi; mysql_field_flags (result qatori,
butun field_offset) resursi; mysql_field_len (result
qatori, butun field_offset)
```

Bu erda result – bu so'rov natijasi identifikatori (masalan, mysql\_list\_fields yoki mysql\_query Funksiyalar bilan jo'natilgan so'rov), field\_offset esa – natijadagi maydonning tartib raqami. mysql\_num\_rows(result resursi) buyrug'i result ning ko'pgina natijalari qatorlari miqdorini qaytaradi.

**Misol:** Artifacts (eksponatlar kollektiyasi) jadvali maydonlari ro'yxatini olish.

```
<?
$conn = mysql_connect("localhost","user","123") or die("Aloqa o'rnatib
bo'lmaydi: ".mysql_error()); echo "Aloqa o'rnatildi";
mysql_select_db("book");
$list_f = mysql_list_fields ("book","Artifacts",$conn);
$n = mysql_num_fields($list_f);
for($i=0;$i<$n; $i++){
 $type = mysql_field_type($list_f, $i);
 $name_f = mysql_field_name($list_f,$i);
 $len = mysql_field_len($list_f, $i);
```

```

$flags_str = mysql_field_flags ($list_f,
$i); echo "
Maydon nomi: ". $name_f;
echo "
Maydon turi: ". $type; echo
"
Maydon uzunligi: ". $len;
echo "
Maydon bayroqlari qatori: " . $flags_str .
"<hr>"; }
?>

```

Natija sifatida taxminan quyidagilarni olish mumkin (albatta, jadvalda ikkita maydon bo'lganda):

- Maydon nomi: id
- Maydon turi: int
- Maydon uzunligi: 11
- Maydon bayroqlari qatori:
- not\_null primary\_key auto\_increment
- Maydon nomi: title
- Maydon turi: string
- Maydon uzunligi: 255 □ Maydon bayroqlari qatori:

### ***HTML-formada maydonlar ro'yxatining aks etishi***

Maydon haqidagi ma'lumotni html-forma elementida aks ettiramiz. BLOB turidagi elementlarni textarea ga o'tkazamiz (TEXT turida biz tuzgan description maydoni BLOB tipiga ega), raqamlar va qatorlarni <input type=text> matnli qatorlarida aks ettiramiz, avtoinkrement belgisiga ega elementni esa umuman aks ettirmaymiz, chunki uning ma'nosi o'z-o'zidan o'rnatiladi.

Buning uchun explode funksiyasidan foydalaniladi:

explode: sintaksisi

explode massivi (separator qatori, string qatori , int limit)

Bu funksiya string qatorini separator taqsimlagichi yordamida qismlarga bo'ladi va olingan qatorlar massivini qaytaradi.

Bizning holatda taqsimlagich sifatida probel “ ” ni olish kera, bo'lish uchun boshlang'ich qator sifatida esa – maydon bayroqlari qatorini.

**Misol.** Artifacts jadvaliga ma'lumot kiritish uchun forma (index.php)

```

<?php
$dblocation = "localhost";
$dbname = "book";
$dbuser = "root";
$dbpasswd = "";
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
@mysql_select_db($dbname,$dbcnx);
if(isset($_POST['save_hide'])) {
$title=($_POST['title']);
$query = "INSERT INTO Artifacts (title) VALUES ('$title')";
if(@mysql_query($query))
{

```

```

echo "<HTML><HEAD><META HTTP-EQUIV='Refresh' CONTENT='0;
URL=index.php'><\HEAD><\HTML>";
} else { print mysql_error(); error("Ma'lumotni bazaga ezishda
xatolik"); }
} else {
?>
<TABLE width="100%" align=center border=0><TR><TD>
<form method="POST" name="save" action="" enctype="multipart\form-data"
>
<input type=hidden name=save_hide value=save_hide>
<TABLE width="100%" align=center border=0><TR>
<TD>Eksponat nomini kiriting <\TD>
<TD><INPUT class="input" size="35" name="title" value=""><\TD>
<\TR><\TABLE>
<input type="submit" value="Saqlash" class="button" G`>
<\form><\TD><\TR><\TABLE>
<? } ?>

```

***Ma'lumotlar bazasiga ma'lumotlar yozish***

Ma'lumki, ma'lumotlarni jadvalga yozish uchun SQL tilidagi INSERT buyrug'i ishlataladi:

```
mysql> INSERT INTO Artifacts SET title='Eksponat nomi';
```

PHP skriptda bunday buyruqdan foydalanish uchun mysql\_query() funksiyasi mavjud. mysql\_query sintaksi mysql\_query resursi (query qatori)

mysql\_query() SQL-so'rovni MySQL ma'lumotlar bazasining ma'lumotlar bazasiga jo'natadi. Agar ochiq aloqa bo'lmasa, funksiya parametrsiz mysql\_connect() funksiyasiga o'xshash holda MBBT bilan bog'lanishga urinadi.

So'rov natijasi buferlanadi.

## 5. Fayllar bilan ishlash

PHP tilida fayllar bilan ishlash jarayonida quyidagi holatlarga duch kelish mumkin:

### Fayllarni ochish

**Fayl** biror ma'lumot saqlash fizik qurilmasidagi baytlar ketma ketligidir. Har bir fayl joylashuvini ko'rsatuvchi absolyut yo'lga ega. Yo'l ajratuvchisi sifatida Windows da to'g'ri slesh (/) yoki teskari slesh (\) ishlatilishi mumkin. Boshqa operatsion tizimlarda faqat to'g'ri slesh ishlatiladi.

Fayllarni serever fayl tizimida ochish **fopen** funksiyasi yordamida amalga oshiriladi:

```
int fopen(string filename, string mode [, int use_include_path])
```

Birinchi argument **filename** – fayl nomi yoki unga olib boruvchi absolyut yo'l. Agar absolyut yo'l ko'rsatilmasa, fayl katalogda joylashgan bo'lishi kerak.

Ikkinchи argument **mode** fayl qaysi amallar uchun ochilganligini ko'rsatadi va quyidagi qiymatlarga ega bo'lishi mumkin:

**r** (faylni faqat o'qish uchun ochish; ochilgandan so'ng fayl ko'rsatkichi fayl boshiga o'rnatiladi);  
**r+** (faylni o'qish va yozish uchun ochish; ochilgandan so'ng fayl ko'rsatkichi fayl boshiga o'rnatiladi); **w** (Yozish uchun Yangi bo'sh fayl yaratish; agar shu nomli fayl mavjud bo'lsa, undagi hamma

ma'lumot o'chiriladi); **w+** (Yozuvlarni o'qish uchun Yangi bo'sh fayl yaratish; agar shu nomli fayl mavjud bo'lsa, undagi

hamma ma'lumot o'chiriladi); **a** (Yozuv qo'shish uchun faylni ochish, ma'lumotlar fayl oxiriga yoziladi); **a+** (Yozuv qo'shish va o'qish uchun faylni ochish, ma'lumotlar fayl oxiriga yozilad);

**b** (Ikkilik fayl Bilan ishslash usulini( o'qishsh va yozish) ko'rsatuvchi bayroq; faqat Windows da ko'rsatiladi).

Uchinchi shart bo'lмаган argument **use\_include\_path** fayllar **include\_path** katalogida izlash kerakligini belgilaydi. (**include\_path** parametr php.ini faylda o'rnatiladi).

Fayl muvaffaqiyatli ochilganda, **fopen** funksiyasi fayl descriptorini qaytaradi, aks xolda – **false** qaytaradi. **Fayl Descriptori** ochilgan faylga ko'rsatkich bo'lib, opertsion tizim tomonidan shu fayl Bilan amallarni qo'llash uchun ishlataladi. Funksiya tomonidan qaytarilgan fayl descriptorini keyinchalik shu fayl bilan ishlaydigan hamma funksiyalarda ko'rsatish lozim.

Quyida keltirilgan kod, C:/WWW/HTML/file.txt faylini o'qish uchun ochadi:

```
<?
$file =
=fopen("c:/www/html/file.txt","r");
if(!file)
{
 echo("Fayl ochish xatoligi");
}
?>
```

Ikkilik fayl, masalan rasmni ochish shu tariqa **b** bayrog'i bioan bajariladi:

```
<?
$file =
=fopen("c:/www/html/river.jpg","rb");
if(!file)
{
 echo("Fayl ochish xatoligi");
}
?>
```

### *Fayllarni akslantirish*

Ochilgan fayldagi ma'lumotlarni brauzerda **fpassthru** funksiyasi yordamida akslantirish mumkin:

**int fpassthru (int file)**

Argument **file** fayl descriptori emasdir.

```
<?
```

```

$file =
=fopen("c:/www/html/pavlovo.jpg","rb");
if(!file)
{
 echo("Fayl ochish xatoligi");
}
else
{
 fpassthru($file);
}
?>

```

Matnli fayllarni akslantirish uchun Yana bir **readfile** akslantirish qunktsiyasi mavjuddir:

**readfile (string filename)**

Shuni ta'kidlash lozimki argument sifatida bu funksiya fayl nomini emas, uning descriptorini qabul qiladi:

```

<? readfile
("file.txt");
?>

```

### *Fayllarni berkitish*

Fayl Bilan ishni tugallagandan so'ng uni yopish kerak, Fayllarni yopish **fclose** funksiyasi yordamida amalga oshiriladi:

**int fclose (int file)**

Argument **file** yopish kerak bo'lgan fayl descriptori.

### *Fayllardan o'qish*

Ochiq fayldan qatorni **fread** funksiyasi yordamida o'qish mumkin: **string fread ( int file, int length )**

Bu funksiya fayl simvollaridan iborat **length** uzunlikdagi **file** descriptorli qatorni qaytaradi.

```

<?
$file =
 fopen("c:/www/html/file.txt","r");
if(!file)
{
 echo("Fayl ochish xatoligi");
}
else
{
 $buff = fread ($file,100);
print $buff;
}
?>

```

Fayldan o'qish uchun **fgets** funksiyasidan foydalanish ham mumkin:

**string fgets ( int file, int length )**

Bu funksiya **length - 1** bayt uzunlikdagi satrni o‘qiydi va qaytaradi. Yangi satr yoki fayl oxiriga etilganda o‘qish to‘xtatiladi. Fayl oxiriga etilganda funksiya bo‘sh satr qaytaradi.

HTML teglarini tashlab yuborgan xolda faylni o‘qish uchun **fgetss** funksiyasi qo‘llanadi: **string fgetss (int file, int length [, string allowable\_tags])**

Shart bo‘lmagan uchinchi parametr **allowable\_tags** tashlab yuborilmasligi kerak bo‘lgan teglar ro‘yxatidan iborat satrni o‘z ichiga oladi. Bu satrda teglar vergul bilan ajratiladi.

Agar fayldagi bor ma’lumotlarni massivga yozish kerak bo‘lsa, **file** funksiya si qo‘llanadi: **array file (string filename [, int use\_include\_path])**

Funksiya **filename** nomli faylni o‘qiydi va har bir elementi o‘qilgan fayldagi satrga mos keluvchi massiv qaytaradi. Quyidagi misolda funksiya yordamida fayl o‘qiladi va undagi bor ma’lumot brauzerga chiqariladi.

```
<?
 $file_array = file("file.txt");
if(!$file_array)
{
 echo("Fayl ochish xatoligi");
}
else
{
 for($i=0; $i < count($file_array); $i++)
 {
 printf("%s
", $file_array[$i]);
 }
}
?>
```

Bu funksiya qulayligi shundaki uning yordamida fayldagi satrlar sonini hisoblash mumkin:

```
<?
 $file_array = file ("file.txt");
if(!$file_array)
{
 echo("Fayl ochish xatoligi");
}
else
{
 $num_str = count($file_array);
echo($num_str);
}
?>
```

Shuni e’tiborga olish kerakki **file** funksiyasini faqat kichkina fayllarni o‘qish uchun qo‘llash kerak. Kengaytmasi \*.csv bo‘lgan fayllarni o‘qish uchun **fgetcsv** funksiyasi qo‘llanadi: **array fgetcsv ( int file, int length, char delim)**

Funksiya fayldan satr o‘qiydi va **delim** simvoli bo‘yicha uni ajratadi. Parametr delim albatta bir simvoldan iborat satr bo‘lishi kerak, aks xolda satrlarning faqat birinchi simvoli inobatga olinadi. Funksiya xosil bo‘lgan massivni yoki fayl oxiriga etilgan bo‘lsa false qiymatni qaytaradi. Bo‘sh satrlar tashlab yuborilmaydi, aksincha bir element bo‘sh satrdan iborat massiv qaytaradi. Parametr **length** satrlar maksimal uzunligini qaytaradi, **fgets** funksiyasidagi kabi. CSV formati MSExcel fayllari saqlanuvchi formatlardan biridir. Quyidagi misolda MSExcel da yaratilgan, foydalanuvchilar parollarini o‘z ichiga olgan file.csv o‘qiladi.

```
<?
$count = 1;
$file = fopen ("file.csv","r");
while ($data = fgetcsv ($file, 1000, ","))
{
 $num = count ($data);
 $count++;
 for ($i=0; $i < $num; $i++)
 {
 print "$data[$i]
";

 }
}
fclose ($file);
?>
```

### *Fayllarga yozish*

Fayllarga yozish **fputs** va **fwrite** bir xil funksiyalari bilan amalga oshiriladi:

```
int fputs (int file, string string [, int length
]) int fwrite (int file, string string [, int
length])
```

Birinchi argument yozuv amalga oshiralayotgan fayl descriptori. Ikkinci argument faylga yozilishi kerak bo‘lgan satr. Uchinchi shart bo‘lmagan argument satrda yozilishi kerak bo‘lgan simvollar soni. Uchinchi argument ko‘rsatilmasa hamma satr yozilishi kerak.

Bu misolda "file.txt" fayliga "Hello, world!" qator yoziladi.

```
<?
$file = fopen ("file.txt","r+");
$str = "Hello,
world!";
if (!$file
)
{
 echo("Fayl ochish xatoligi");
}
else
{
 fputs ($file, $str);
```

```
 }
 fclose ($file);
?>
```

### **Fayllardan nusxa olish, nomini o‘zgartirish va fayllarni o‘chirish**

Fayldan nusxa olish **copy** funksiyasi yordamida amalga oshiriladi: **int copy ( string file1, string file2)**  
**copy** funksiyasi **file1** nomli fayldan **file2** nomli fayl nusxa oladi. Agar fayl **file2** mavjud bo‘lsa, u qaytadan yoziladi.

Fayl nomini o‘zgartirish **rename** funksiyasi yordamida amalga oshiriladi: **int rename ( string old, string new)**

Bu funksiya fayl **old** nomini **new** nomiga almashtiradi.

Agar fayl Yangi nomi boshqa fayl tizimida joylashgan bo‘lsa **rename** funksiyasi faylga Yangi nom bermaydi.

Faylni o‘chirish **unlink** funksiyasi yordamida amalga oshiriladi: **int unlink ( string filename)**

### **Fayllar atributlari**

Fayl atributlari haqida qo‘srimcha ma’lumot olish uchun quyidagi funksiyalardan foydalanishingiz mumkin. **file\_exists** funksiyasi fayl mavjudligini tekshiradi va fayl mavjud bo‘lsa true, aks xolda false qiymat

qaytaradi:

**bool file\_exists ( string filename)** **filectime** funksiyasi faylga oxirgi murojaat vaqtini qaytaradi:  
**int filectime ( string filename)** **filemtime** funksiyasi faylning oxirgi o‘zgartirish vaqtini qaytaradi: **int filemtime ( string filename)** **file\_size** funksiyasi fayl xajmini baytlarda qaytaradi: **int file\_size ( string filename)** **file\_type** funksiyasi fayl tipini qaytaradi: **string file\_type ( string filename)**

Bu funksiya qaytaradigan satr quyidagi fayl tiplaridan biriga tegishli bo‘ladi:

- char (maxsus simvolli qurilma);
- dir (katalog);
- fifo (nomlangan kanal);
- link (simvolli ilova);
- block (maxsus blokli qurilma);
- file (oddiy fayl);
- unknown (tip ma’lum emas).

Fayl harakteristikalarini qaytaruvchi funksiyalardan foydalanish ko‘p resurslarni talab qilgani uchun, bunday funksiyalarni chaqirishda unumdorlikni yo‘qotmaslik uchun, PHP fayl haqidagi ma’lumotni keshlaydi. Bu keshni **clearstatcache** funksiyasi yordamida tozalash mumkin:

```
<?
clearstatcache(
); ?>
```

## **Fayllar bo'yicha ko'chish**

Fayldan ma'lumot o'qilganda joriy pozitsiya ko'rsatkichi navbatdagi o'qilmagan simvolga suriladi. Ko'rsatkich xolatini boshqarishga imkon beruvchi bir necha funksiyalar mavjuddir. Joriy pozitsiya ko'rsatkichini fayl boshiga keltirish uchun **rewind** funksiyasi qo'llanadi: **int rewind ( int file ) file** argumenti fayl descriptoridir.

Ko'rsatkich joriy pozitsiyasini **ftell** funksiyasi yordamida aniqlash mumkin: **int ftell ( int file )**

Ko'rsatkichni fayl ixtiyoriy joyiga **fseek** funksiyasi yordamida o'rnatish mumkin: **int fseek ( int file, int offset [, int whence ] )**

Funksiya **fseek** fayl ko'rsatkichini **offset** siljishli baytga o'rnatadi (fayl boshidan, oxiridan yoki joriy pozitsiyadan **whence paorametri qiyamatiga qarab**). Argument **file** fayl descriptoridir. Argument **whence** qaysi joydan **offset** siljish hisoblanishi kerakligini aniqlaydi va quyidagi qiyatlardan biriga teng bo'lishi mumkin:

**SEEK\_SET** (pozitsiyani fayl boshidan hisoblaydi);

**SEEK\_CUR** (pozitsiyani ko'rsatkich joriy pozitsiyasidan otschitivaet pozitsiyu otnositelno tekishi polojeniya ukazatelya);

**SEEK\_END** (otschitivaet pozitsiyu otnositelno kontsa fayla).

Ko'zda tutilgan bo'yicha argument **whence** qiymati **SEEK\_SET**.

Ko'rsatkich fayl oxirida ekanligini **feof** funksiyasi yordamida aniqlash mumkin: **int feof ( int file )**

Agar ko'rsatkich fayl oxirida joylashgan bo'lsa, funksiya **true** qaytaradi aks xolda **false**.

**feof** funksiyasini fayl o'qishda ishlatish qulaydir:

```
<?
 $file = fopen
("file.txt","r"); if ($file)
{
 while(!feof($file))
 {
 $str =
fgets($file);
 echo $str; echo
("
");
 }
 fclose ($file);
}
else
{
 echo("Fayl ochish xatoligi");
}
?>
```

Bu funksiya yordamida faylda satrlar sonini aniqlash qulay:

```
<?
 $file = fopen
("file.txt","r"); if ($file)
```

```

{
 $counter = 0;
while(!feof($file))
{
 $str = fgets ($file);
 $counter++;
}
echo($counter);
fclose ($file);
}
else
{
 echo("Fayl ochish xatoligi");
}
?>

```

### **Kataloglar bilan ishlash**

Joriy kattalogni o‘rnatish uchun **chdir** funksiyasi qo‘llanadi: **int chdir ( string directory)**

Bu funksiya bilan quyidagicha ishslash mumkin:

**chdir("./tmp/data");** // absolyut yo‘l bo‘yicha o‘tish **chdir("./js");** // joriy kattalog ost kattalogiga o‘tish **chdir("../");** // ajdod kattaloga o‘tish **chdir("~/data");** // o‘tamiz /home/foydanuvchi/data (Unix uchun) Joriy kattalogni aniqlash uchun **getcwd** funksiyasidan foydalanish mumkin: **string getcwd ( string path)**

Kattalogni ochish uchun path parametri Bilan berilgan kattalogni ochuvchi **opendir** funksiyasidan foydalaniladi:

**int opendir ( string path)**

Kattalog ochilgandan so‘ng, uni **readdir** funksiyasi Bilan o‘qish mumkin: **string readdir ( int dir)**

Bu funksiya kattalogdagi elementlar nomlarini qaytaradi. Undan tashqari kattaloglarda "." va ".." elementlari mavjud. Birinchi element joriy ikkinchisi ajdod kattaloga ko‘rsatadi. Joriy kattalogni nomini "." sifatida ko‘rsatib ochish mumkin: **\$dir = opendir (".");**

Kattalog bilan ish tugagandan so‘ng uni yopish kerak. Kattalog yopilishi **closedir** funksiyasi Bilan amalga oshiriladi:

**void closedir (\$dir)**

Quyida joriy kattalogdagi fayllarni o‘qish va chiqarishga misol keltirilgan.

<?

```

$dir = opendir (".");
echo "Files:\n";
while ($file = readdir ($dir))
{
 echo "$file
";
}
closedir ($dir);
?>

```

Bu funksiya "." va ".." qiymat ham qaytaradi. Agar bu kerak bo'lmasa, bu qiymatlarni quyidagicha tashlab yuborish mumkin:

```
<?
$dir = opendir (".");
while ($file = readdir
($dir))
{
 if (($file != ".") && ($file != ".."))
 {
 echo "$file
";

 }
}
closedir ($dir);
?>
```

Ko'rilgan funksiyalarga misol sifatida, c:/temp katalogidagi bir sutka davomida murojaat qilinmagan hamma fayllarni o'chiruvchi script yaratamiz. Fayllarni o'chirish funksiyasi bu holda rekursiv chaqiriladi.

```
<? function delTemporaryFiles
($directory)
{
 $dir = opendir ($directory);
while (($file = readdir
($dir)))
{
 if(is_file ($directory."/".$file))
 {
 $acc_time = fileatime ($directory."/".$file);
 $time = time();
 if (($time - $acc_time) > 24*60*60)
 {
 if (unlink ($directory."/".$file))
 {
 echo ("Fayllar muvaffaqiyatli o'chirilgan");
 }
 }
 }
 else if (is_dir ($directory."/".$file) && ($file != ".") &&
($file != ".."))
 {
 delTemporaryFiles ($directory."/".$file);
 }
}
closedir ($dir);
}
```

```
 delTemporaryFiles ("c:/temp");
?>
```

Kattaloglarni yaratish **mkdir** funksiyasi yordamida amalga oshiriladi:

```
bool mkdir (string dirname, int mode)
```

Bu funksiya dirname nomli i mode.murojaat huquqlari bilan kattalog yaratadi. Kattalog yaratilmasa false qaytaradi. Murojaat xquqlari faqat UNIX kattaloglari uchun beriladi, chunki Windows da bu argument ta'sir ko'rsatmaydi. Quyida c:/temp direktoriyasida test v kattalogini yaratish misoli ko'rilib.

```
<?
$flag = mkdir ("c:/temp/test",
0700); if($flag) {
 echo("Kattalog muvaffaqiyatli yaratilgan");
}
els
e
{
 echo("Kattalog yaratish xatosi ");
}
?>
```

Kattalogni **rmdir** funksiyasi yordamida o'chirish mumkin: **bool rmdir ( string dirname)**

Endi yaratilgan /test kattalogini o'chiramiz:

```
<?
$flag = rmdir
("c:/temp/test");
if($flag) {
 echo("Kattalog muvaffaqiyatli o'chirilgan");
}
els
e
{
 echo("Kattalogni o'chirish xatosi ");
}
?>
```

Funksiya **rmdir** faqat bo'sh kattaloglarni o'chiradi. Bo'sh bo'limgan kattaloglarni o'chirish uchun,

funksiya yaratib c:/temp kattalogini undagi papkalar va fayllari bilan birga o'chiramiz:

```
<? function full_del_dir
($directory)
{
 $dir = opendir($directory);
while(($file = readdir($dir)))
{

```

```

if (is_file ($directory."/".$file))
{
 unlink ($directory."/".$file);
}
else if (is_dir ($directory."/".$file)
&& ($file != ".") && ($file !=
".."))
{
 full_del_dir ($directory."/".$file);
}
closedir ($dir);
rmdir
($directory);
echo("Kattalog muvaffaqiyatli o'chirilgan");
}
full_del_dir ("c:/temp")
?>

```

Funksiyani rekursiv chaqirganda argumentlar sifatida joriy va ajdod kattaloglarga ko'rsatuvchi  
"." va

".." yozuvlarni uzatmang, chunki bu holda siz ma'lumotlaringizni yo'qotishingiz mumkin. Bu yozuvlarni shartli operator yordamida o'tkazib yuboring.

### ***PUTva POST usullari***

HTTP PUT va HTTP POST usullari fayllarni serverga yuklash uchun mo'ljallangan.

Web-serverdagи ma'lumot bilan ishslash uchun HTTP Protokolda uchta usul mavjud: GET, PUT va POST. GET usuli Web-sahifalarni olish uchun qo'llanadi, bunda forma hamma o'zgaruvchilari URL ga uzatiladi. Ko'pgina Web-serverlarda URL maksimal uzunligi cheklanganligi uchun (odatda 1024 dan oshmaydi), katta xajmdagi ma'lumotlarni uzatishda GET usulini qo'llab bo'lmaydi.

PUT usuli serverdagи ma'lumotni Yangilash uchun qo'llanadi va HTTP PUT so'rov natijasi serverda saqlanishini talab qiladi. So'rov quyidagi ko'rinishga ega:

PUT /path/filename.html HTTP/1.1

Bu holda Web-server bu so'rov natijasini Web-serverning URL nomlar fazosida /path/filename.html ko'rinishda saqlashi lozim. Ko'zda tutilgan bo'yicha Web-server o'zi bunday so'rovlarni bajarmaydi, balkim ularni qayta ishslash uchun CGI-stsenariy beradi. Apache da PUT so'rovlarni qayta ishlovchi stsenariy tayinlash uchun httpd.conf fayldagi Script direktivasini o'zgartirish mumkin, masalan quyidagicha:

Script PUT /cgi-bin/put.cgi Bu shuni bildiradiki PUT-so'rovlarni CGI-script put.cgi qayta ishlaydi.

Odatda fayllarni serverga yuklash uchun HTTP POST usulidan foydalaniladi. Bu usul formadan ma'lumotlar katta xajmini uzatishga imkon beradi va so'rov tanasidagi hamma forma o'zgaruvchilarni saqlaydi.

### **Nazorat savollari:**

1. Ma'lumotlar bazasi deganda nimani tushunasiz?

2. Qanaqa ma'lumotlar bazasi dasturlarini bilasiz va ularni imkoniyatlari haqida gapiring?
3. PHP da MySql bilan bog'lanish funksiyasini misollar yordamida tushuntirib bering?
4. PHP da MySql so'rovlarini amalga oshirishga misol keltiring?
5. Turli ma'lumotlar bazasi dasturlarida ishlash jarayonida har biriga tegishli bo'lgan kamchilik va afzalliklarni ayting?
6. Fayllar bilan ishlash jarayonlarini tushuntiring?
7. Fayllarni serverga yuklash uchun mo'ljallangan qanday usullar mavjud?
8. Web-serverdagi ma'lumot bilan ishlash uchun qanday usullar mavjud va ularning ishslash mohiyatini tushuntiring?
9. PUTva POST usullari usullarining mohiyatini tushuntiring?
10. PHP dasturlash tilida fayllarni yozish, ochish, berkitish, nusxa ko'chirish qanday amalga oshiriladi?

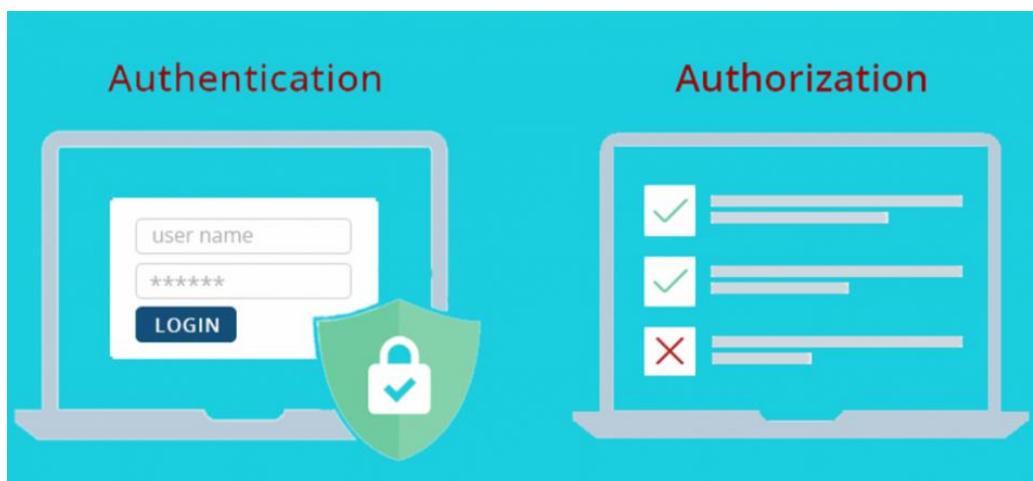
## 23-ma’ruza. PHP da autentifi-kasiya va avtorizasiyani amalga oshirish usullari

1. PHP da registrasiya va autentifikasiya sahifasini ishlab chiqish.
2. Foydalanuvchining sahifalarga kirish ruxsatini belgilash usullari.

Autentifikatsiya va avtorizatsiya atamalari xavfsizlik va tizimga kirish vaqatida bir-biri bilan hamohang ishlatiladi. Ikkala atama ham Internet bilan xizmat ko'rsatish infratuzilmasining asosiy bo'laklari sifatida bog'langan juda muhim mavzular. Biroq, bu ikkita atama mutlaqo boshqacha tushunchalar bilan bir-biridan farq qiladi. Endi ushbu atamalar nima ekanligini tushunishga harakat qilib ko'raylik.



**Autentifikatsiya** sizning shaxsingizni tasdiqlashni anglatadi, **avtorizatsiya** tizimga kirish huquqini anglatadi. Oddiyroq qilib aytganda, autentifikatsiya - bu o'zini tekshirish jarayoni, avtorizatsiya - bu kirish huquqini tekshirish jarayoni.



### Identification (Identifikatsiya)

Dastlab ikki atamani tushunishdan oldin yana bir muhim atamalardan biri bo'lgan atama **Identifikatsiya** nima ekanligini bilib olish lozim. Buni oddiy tarzda tushuntirishga harakat qilaman. Ko'pgina veb-saytlarga kirganingizda, foydalanuvchi nomini kiritasiz (*login*).

Agarda siz yangi hisob qaydnomasi (*profile*) yaratmoqchi bo'lsangiz, sizdan identifikatsiyalash uchun foydalanuvchi nomini tanlashingiz so'raladi. Kirish paytida siz kiritadigan foydalanuvchi nomi bu "Identifikatsiya" dir. Bu shunchaki sizning shaxsingizni tasdiqlash usulidir. Yana ham oddiyroq aytadigan bo'lsak siz kiritadigan foydalanuvchi nomi bu barcha ilovalar yoki veb saytlarda so'raladigan login, e-mail yoki telefon raqamini tushunish mumkin.

### Autentifikatsiya (Authentication)

Autentifikatsiya bu sizning foydalanuvchi identifikatoringizni tekshirish uchun foydalanuvchi nomi / foydalanuvchi identifikatori va parol kabi ma'lumotlarini tekshirish bilan bog'liq jarayonni o'z ichiga oladi. Shundan so'ng tizim sizning ma'lumotingizdan foydalanayotganligingiz yoki yo'qligingizni tekshiradi. Global yoki lokal tarmoqlarda bo'lsin, tizim login parollar orqali foydalanuvchi identifikatorini tasdiqlaydi. Odatda autentifikatsiya qilish foydalanuvchi nomi va parol orqali amalga oshiriladi, garchi autentifikatsiya qilishning boshqa usullari bo'lsa ham.

Oddiyroq aytadigan bo'lsak, Siz login va parolingiz orqali tizimga kirishga urunishingiz bu autentifikatsiya deyiladi.

Autentifikatsiya qilish omillari tizim har kimga biron bir narsaga kirish huquqini berishdan oldin shaxsini tasdiqlash uchun foydalanadigan ko'plab turli elementlarni aniqlaydi. Shaxsnинг identifikatori shaxs nimani bilishi mumkinligini aniqlaishi mumkin va xavfsizlik to'g'risida gap ketganda, tizimda kimgadir ruxsat berish uchun kamida ikkita yoki uchta autentifikatsiya qilish omillarini tekshirish kerak. Xavfsizlik darajasiga qarab, autentifikatsiya qilish omillari quyidagilarda bir biridan farq qilishi mumkin.

**Single- Factor Authentication** (Bir faktorli autentifikatsiya): Bu autentifikatsiya jarayonining eng oddiy shakli bo'lib, foydalanuvchiga veb-sayt yoki tarmoqda muayyan tizimga kirish huquqini berish uchun parolni talab qiladi. Shaxs identifikatorini tekshirish uchun faqat bitta ma'lumotlardan foydalanib tizimga kirishni amalga oshirishi mumkin. Masalan, foydalanuvchi nomiga tegishli parolnigina talab qilish orqali faqat bitta faktorli autentifikatsiya yordamida login ma'lumotlarini tekshirishi mumkin.

**Two- Factor Authentication:** Ushbu autentifikatsiya ikki bosqichli tekshirish jarayonini talab qiladi, bu nafaqat foydalanuvchi nomi va parolni, balki faqat foydalanuvchi biladigan ma'lumotni ham talab qiladi. Foydalanuvchi nomi va parolni maxfiy ma'lumotlar bilan birgalikda ishlatalish xakerlarga muhim va shaxsiy ma'lumotlarni o'g'irlashni ancha qiyinlashtiradi.

**Multi- Factor Authentication** (ko'p faktorli autentifikatsiya): Bu autentifikatsiyaning eng ilg'or usuli bo'lib, foydalanuvchilarga tizimga kirish huquqini berish uchun mustaqil autentifikatsiya kategoriyalardan ikki yoki undan ko'p darajadagi xavfsizlikni talab qiladi. Autentifikatsiya qilishning ushbu shakli har qanday ma'lumotlarga ta'sir qilishni bartaraf etish uchun bir-biridan mustaqil bo'lgan omillardan foydalanadi. Moliyaviy tashkilotlar, banklar va huquqni muhofaza qilish idoralarida ko'p faktorli autentifikatsiyadan foydalanish odatiy holdir.

### Authorization (Avtorizatsiya)

Avtorizatsiya sizning shaxsingiz tizim tomonidan muvaffaqiyatli tasdiqlanganidan so'ng amalgा oshiriladi. Shuning uchun sizga ma'lumot, fayllar, ma'lumotlar bazalari, fondlar va boshqa manbalarga to'liq kirish huquqini beradi. Ammo avtorizatsiya sizning kirish huquqingizni aniqlagandan keyingina manbalarga kirish huquqini tasdiqlaydi. Boshqacha qilib aytganda, avtorizatsiya - bu autentifikatsiya qilingan foydalanuvchining muayyan manbalardan foydalana olishini aniqlatuvchi jarayon hisoblanadi. Autentifikatsiya orqali xodimning identifikatori va parollarini tekshirilib, tasdiqlanganidan so'ng, keyingi qadam qaysi xodimning qaysi qavatga kirish huquqiga ega ekanligi va bu avtorizatsiya orqali nimalarni amalga oshirilishini aniqlash imkonini beradi.

Tizimga kirish autentifikatsiya va avtorizatsiya bilan himoyalangan va ular ko'pincha bir-biri bilan birgalikda ishlataladi. Garchi ikkalasi ham bundan keyin turli xil tushunchalarga ega bo'lishsada, ular veb-servis infratuzilmasi uchun, ayniqsa tizimga kirish huquqi haqida gap ketganda juda muhimdir. Har bir atamani tushunish juda muhim va xavfsizlikning muhim jihatи hisoblanadi.

### Ro'yxatdan o'tish bo'limini hosil qilish

Frontend qismi hosil bo'lgandan so'ng, forma elementlari orqali kiritilgan ma'lumotlarni post usuli yordamida qabul qilib olib, u ma'lumotlarni to'g'riligini tekshirgandan so'ng, ma'lumotlar bazasiga saqlaymiz.

## Register

Login

FISH

Email

Telefon

Parol

Qayta parol

Already have an account? [Login to your account!](#)

[Create an Account](#)

```
include "/config/connect.db.php";
if(isset($_POST['username'])){
 $username=str_replace("","",$_POST['username']);
 if($username==""){ unset($username);}
}
if(isset($_POST['email'])){
 $email=str_replace("","",$_POST['email']);
 if($email==""){ unset($email);}
}
if(isset($_POST['password'])){
 $password=md5($_POST['password']);
 if($password==""){ unset($password);}
}
if(isset($_POST['password2'])){
 $password2=md5($_POST['password2']);
 if($password2==""){ unset($password2);}
}
if(isset($_POST['fullname'])){
 $fullname=str_replace("","",$_POST['fullname']);
 if($fullname==""){ unset($fullname);}
}
```

```

if(isset($_POST['phone'])){
 $phone=str_replace(" ","$_POST['phone']");
 if($phone==""){ unset($phone);}
}

$datetime=date('Y-m-d H:i:s');

if($password==$password2){

 if(isset($username) && isset($password) && isset($email) &&
 isset($fullname) && isset($phone)){
 $result=$mysqli->query("INSERT INTO `user`(`user_name`,
 `user_full_name`, `user_reg_date`, `user_email`, `user_phone`,
 `user_status`, `user_password_hash`, `user_image`,
 `user_role`) VALUES('$username','$fullname','$datetime',
 '$email','$phone',10,'$password',2)");

 if($result){
 header("Location: index.php");
 }
 }
}

```

## Kirish bo'limini hosil qilish

Kirish bo'limi Frontend qismnini hosil qilib bo'lgandan so'ng, foydalanuvchini login va parolini to'g'rilingini tekshirish qismini ko'rib chiqamiz

**Login**

Username:

Password:

Don't have an account? [Create an Account now!](#)

[Parolni unutdingizmi?](#)

**Kirish**

```

<?php
session_start();
include "/config/connect.db.php";
if(!isset($_SESSION['user'])){
 if($_POST){
 if(isset($_POST['username'])) $username=$_POST['username'];
 if(isset($_POST['password'])) $password=md5($_POST['password']);
 if(isset($username) && isset($password)){
 $result=mysqli->query("select * from user where user_name='".$username.' and
user_password_hash="'.$password.'"");
 if($result->num_rows>0){
 $user_info=$result->fetch_array(MYSQLI_ASSOC);
 }
 $_SESSION['user']=$user_info['user_full_name'];
 header("Location: index.php");
 }
 }
}
?>

```

Foydalanuvchi tizimga kirgandan so'ng, kirganligini PHPda tekshirish scripti quyidagicha:

```

if(isset($_SESSION['user'])){
//kodlar
}

• Foydalanuvchi tizimga kirgandan so'ng, tizimdan chiqish uchun PHPda tekshirish scripti
quyidagicha:
<?php
session_start();
if(session_destroy()){ ?>
<script>
window.location="/index.php";
</script>
<?php }?>

```

## **25-ma’ruza. Veb saytlarni yaratishda CMS texnologiyasini qo’llash**

### **REJA:**

1. CMS (kontentlarni boshqarish tizimlari) texnologiyasi haqida.
2. Kontentlarni boshqarish tizimlaridan foydalanishning afzallik va kamchiliklari.
3. CMS xususiyatlari. Mavjud zamонави CMS turlari.

### **CMS (kontentlarni boshqarish tizimlari) texnologiyasi haqida**



Barchaga ma'lumki, butunjahon tarmog'idagi veb-sahifalar kundan kunga ko'payib bormoqda. Bu holat O'zbekistonda, ya'ni Uznetda ham jadallik bilan amalga oshmoqda. Kimdir o'z shaxsiy sahifasini yaratса, kompaniya va firmalar korporativ saytlarini yaratmoqda. Yana kimdir o'z kundaliklarini, ya'ni bloglarini olib boradilar.

Shu bilan birga, yirik portallar faoliyati ham rivojlanmoqda. Bularning hammasi esa biror dasturiy vosita yordamida yaratiladi. Kimdir HTML tilida, kimdir PHP da, yana kimlardir CMS deb

ataluvchi tizimlar yordamida veb-saytlarni yaratmoqda. Xo'sh, CMS degani o'zi nima? Bu tizimni nima ekanini tushunish uchun, keling, avvalo, veb-sahifalar qanday yaratilishi haqida bilib olamiz.

1989-yilda Tim Berners-Li tomonidan World Wide Web, ya'ni Bununjahon o'rgimchak to'ri ixtiro qilinganidan boshlab uning har bir foydalanuvchisi oddiy bo'lsa-da, o'z veb-sahifasini yarata olish imkoniyatiga ega bo'ldi. Lekin, buning uchun HTML (Hyper Text Markup Language) tilini bilish shart edi. Turli xil IT-kompaniyalari tomonidan bu ishni osonlashtiruvchi bir qancha dasturiy paketlar, jumladan Adobe Dreamweaver va Microsoft FrontPage kabilar yaratildi. Bunday dasturlardan ba'zilari sayt yaratishning barcha bosqichlarini (HTML-kodni yozishdan bu kodni serverga yuklashgacha) boshqarishni o'z zimmasiga oladi. Shu bilan birga, ko'pgina hollarda (aynan oddiy sahifalarni yaratishda) foydalanuvchi bu kodni bilishi shart bo'limgan. Sababi bu dasturlarda veb-sahifalarni yaratishning vizual usullaridan foydalanilgan.

Biroq, HTML-redaktorlarining barcha afzalliklariga qaramay, saytni boshqarish vaqt o'tgani sayin va yangi ma'lumotlar qo'shilishi bilan qiyinlashib boraverdi. Chunki axborot texnologiyalari rivojlanishi bilan veb-saytlar ham rivojlanib bordi. Sayt xaritasini yaratishdek oddiy funksiyani qo'shish uchun ham veb-yaratuvchilar tomonidan ancha-muncha mehnat talab etilar edi. Axir buning uchun barcha ilovalarni ko'rib chiqish, sahifalar orasidagi bog'liqliklarni tekshirish, menyuni yaratish, forumni qo'shish kabi ishlarni bajarish lozim edi. Bu esa doimiy diqqatni va juda ko'p vaqtini talab etardi.

Sayt yangilanishi muammosini, ya'ni yangilanishning avtomatlashtirishini hal qilish maqsadida yirik axborot kompaniyalari, jumladan, Newsweek va Time yangi dasturiy ta'minot yaratish ishlarini boshlashga majbur bo'ladi. Bu dasturiy ta'minotlar kontentni (ma'lumotlarni) boshqarish tizimlari (CMS – Content Management System) deb ataldi.

## **Statik va dinamik veb-sahifalar**

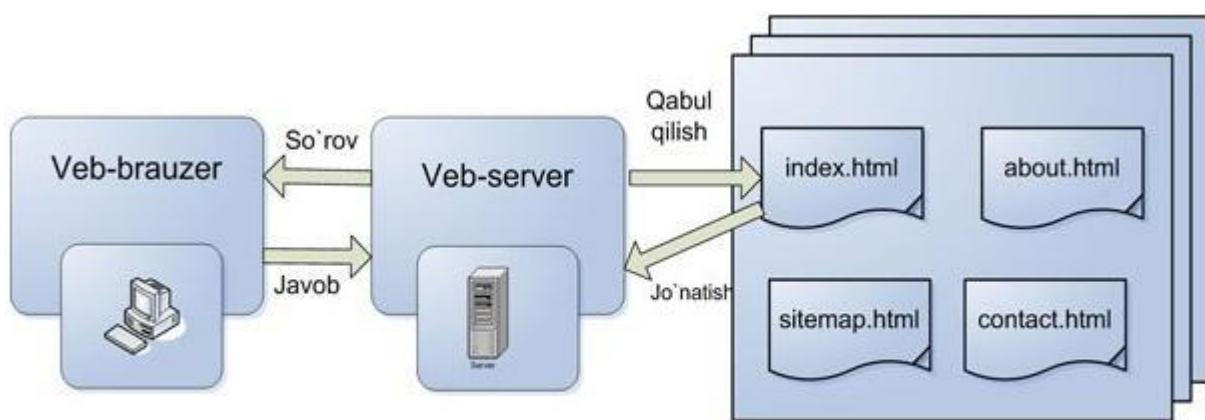
Endi savol tug'ilishi mumkin: oddiy HTML kodida yozilgan sahifa bilan CMS nimasi bilan farq qiladi? Avvalombor, HTML da yozilgan sayt statik holda bo'lsa, zamonaviy saytlar dinamik ko'rinishga ega.

Oddiy sayt qanday prinsip asosida ishlashini ko'rib chiqamiz. Foydalanuvchi biror-bir, masalan, page.html sahifasiga murojaat qildi deylik. Veb-server o'zidagi veb-sahifalar katalogini ko'rib chiqadi va agar page.html bo'lsa, foydalanuvchiga uni ko'rsatadi, aks holda esa – 404-chi xabarni ("Fayl topilmadi") yetkazadi.

Dinamik sayt esa biroz boshqacha ishlaydi. Foydalanuvchi biror sahifaga murojaat qiladi, biroq bu murojaat diskdag'i aniq bir faylga emas, balki sayt dvijokiga (agar dvijok PHP tilida yozilgan bo'lsa,

odatda bu fayl index.php bo'ladi) qaratilgan bo'ladi. Dvijok foydalanuvchiga kerakli ma'lumotni biror manbadan (masalan, ma'lumotlar bazasi yoki diskdag'i fayldan) oladi, uni HTML-sahifaga aylantiradi va veb-serverga uzatadi, u esa o'z navbatida, foydalanuvchi brauzeriga jo'natadi.

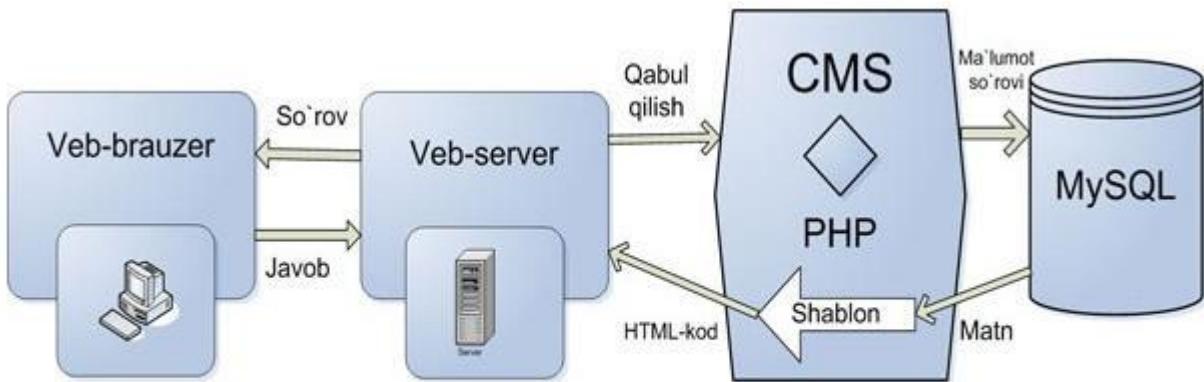
Unda yana bir savol tug'iladi: agar bitta faylningina so'rab, uni brauzerga uzatish mumkin bo'lsa, buncha murakkablashtirishning o'zi nimaga kerak? Shuning uchunki, bu saytni boshqarish qulay bo'lsin. Aytaylik, siz 100 sahifalik (bu unchalik ko'p emas) oddiy sayt yaratdingiz. Ertami-kech siz bu sayt dizaynini, ko'rinishini o'zgartirmoqchi bo'lasiz. U holda esa siz shu 100 sahifani qo'lda o'zgartirib chiqishingizga to'g'ri keladi. Ustiga-ustak, siz avvaliga bu 100 sahifani kompyuteringizdagi lokal serverga yuklab olishingiz, undan so'ng tahrirlab, kerakli o'zgartirishlarni kiritishingiz va oxirida yana serverga yuklastingiz kerak bo'ladi.



### Veb-serverning CMS siz holda qanday ishlashi

Ana endi dinamik sayt qanday ishlashini ko'rib chiqaylik. Asosiy voqe'a – "HTML-sahifani yaratish". Ya'ni dvijok kontentni – matnli ma'lumotni (u rasm va jadvallar bilan ham bo'lishi mumkin) oladi va uni dizaynga "kiyintiradi". Demak, saytning dizaynini o'zgartirish uchun siz faqatgina tayyor grafik mavzuni o'rnatishingiz yoki bir-ikkita faylni – sayt shabloni fayllarini o'zgartirishingiz kerak.

CMS tizimi sahifalarni dinamik ko'rinishga keltiradi: ular foydalanuvchi sahifani ko'rmoqchi bo'lgandagina amalda yaratiladi. Bu degani, sahifalar foydalanuvchi tomonidan bevosita ko'rileyotganda yangilanishi yoki o'zgartirilishi mumkin.



Veb-serverning CMS bilan qanday ishlashi

### Kontentlarni boshqarish tizimlaridan foydalanishning afzallik va kamchiliklari.

Bu tizimlarning asosiy jihat shundaki, dinamik veb-sahifalarni oson yarata olish va ularni turli xil axborot bilan to'ldirish imkoniyatini berishdir. CMS tizimlari odatda murakkab bo'lib, ular o'z ichiga xabarlarni tarqatish xizmatlari (RSS), forumlar va internet-do'konilarni oladi. Shu bilan birga, ularni o'zgartirish ham oson.

CMS nafaqat saytdagi turli ma'lumotlarni boshqarishni avtomatlashtiradi, balki shu bilan birga, sayt tuzilishining texnik tomonlari bilan tanish bo'limganlar, ya'ni yozuvchi va jurnalistlarga saytda o'z materiallarini bevosita foydalanuvchi interfeysi orqali qo'shish imkoniyatini beradi. Shu narsa tushunarli bo'ldiki, endi foydalanuvchilarga saytda o'z materiallarini ko'rish uchun HTML ni ham, saytdagi ma'lumotlar yangilanishining texnologiyasini ham bilishi shart emas.

CMS tizimi, shuningdek, veb-sayt yaratishni turli xil mutaxassislar orasida taqsimlab chiqish imkoniyatini beradi. Shunga ko'ra, veb-dizayner o'z diqqatini saytning ko'rinishiga (dizaynni mukammallashtirish va shablonlarni o'rnatishga) qaratsa, texnik yo'nalishda bo'limgan mutaxassislar uni ma'lumotlar (matn, tasvir va animatsiyalar) bilan to'ldirishga harakat qiladi.

Bu ham hammasi emas. Ma'lumki, dvijok – CMS ning bir qismi xolos. U dizayn shabloni asosida sayt ma'lumotlarini chiqarib beradi. Bundan tashqari, CMS da saytdagi ma'lumotlarni boshqarish (ular sayt bo'limlarini va kategoriylarini yaratishga va ma'lumotlarni ular orasida bo'lib chiqishga imkon beradi) va bu ma'lumotlarni tahrirlash vositalari ham mavjud. Shu bilan birga, foydalanuvchi sayt ma'lumotlari aynan qayerda (ma'lumotlar bazasida, diskda HTML-fayl ko'rinishida yoki umuman tashqi serverda) saqlanayotganini bilishi shart emas.

CMS qo'llanilishi bilan ko'pgina qiyinchiliklar bartaraf etilishi mumkin. Yuqorida misol sifatida ko'rilgan sayt xaritasi ham avtomatik ravishda tuziladi, buning uchun foydalanuvchidan hech qanday mehnat talab etilmaydi. Shu bilan birga, CMS saytdagi qo'shimcha funksiyalarni, masalan,

suratlar galereyasi yoki ro'yxatdan o'tish oynasini qo'llash imkonini beradi – shu kabi funksiyalar CMS kengaytmalari orqali amalga oshiriladi. Faqatgina u yoki bu kengaytmani o'rnatish kerak xolos.

## **CMS ga o'tishning o'ziga xos qiyinchiliklari**

Xo'sh, CMS larning shuncha afzalliklari bo'lгanda nima uchun barcha sayt egalari Joomla! yoki Drupal kabi CMS larga otishga intilmayapti? Bunga bir qancha sabablarni ko'rsatish mumkin.

Birinchi sabab, inertsiya bo'yicha ketish yoki boshqacha aytganda, o'zi bilgan narsaga o'rganib qolish. Inson HTML ni ishlatishga o'rganib qoladi va CMS ga o'tish saytni boshqarishni avtomatlashtirish bo'yicha qancha qulayliklar keltirmasın, biror narsani o'zgartirishni xohlamaydi. HTML dan CMS ga o'tish-yillar ichida amalga oshishi mumkin.

Ikkinci sabab, mavjud minglab HTML-sahifalarni CMS ma'lumotlar bazasiga moslab o'zgartirish kerak bo'ladi. Aksariyat hollarda buni qo'lda bajarish kerak bo'ladi. Katta sayt uchun bu jarayon ancha ko'p vaqtini oladi, bu esa saytni vaqtinchalik faoliyatining to'xtashiga va balki, kompaniya tomonidan ko'rildigan moliyaviy zararga olib keladi. Albatta, CMS ga o'tish saytni boshqarish bo'yicha ancha vaqt tejalishini hisobga olsa, kelajakda bu o'zini oqlaydi.

Endi, CMS ga o'tishga halaqit beruvchi faqatgina bitta to'siq qoladi – texnik holat. Deyarli barcha zamonaviy CMS ishlashi uchun xostingda PHP interpretatori va MySQL ma'lumotlar bazasi serveri bo'lishi kerak. Besh-olti-yil avval shu kabi funksiyalarni taqdim etuvchi xosting-provayderlar kam edi. Hozirga kelib, vaziyat yaxshi tarafga o'zgardi va PHP hamda MySQL ni qollaydigan xostingni O'zbekistonning o'zida topish qiyinchilik tug'dirmaydi.

## **CMS xususiyatlari. Mavjud zamonaviy CMS turlari**

Hozirgi davrga kelib, CMS asosiga qurilgan bir qancha tizimlar yaratilgan. Ularga misol qilib, Drupal, PHP-Nuke, WebDirector, Joomla!, NetCat, Slaed, Microsoft CMS, WordPress, PHPShop kabilarni keltirish mumkin. Bu tizimlarning ba'zilari umuman tekin bo'lsa, ba'zilaridan esa haqqini to'labgina foydalanish mumkin.

CMS ni tanlash bir tomondan oson va jo'ndek ko'rinsada, bu juda jiddiy masala. Internetda ularning turli yo'nalishdagi o'nlab (agar yuzlab bo'lmasa): yuz dollarlab turadiganidan to ochiq kodli bepul turigacha, ma'lum bir sohagagina taalluqlisidan ixtiyoriy murakkablikdagi sayt qurish imkonini beradiganigacha, oson boshqariladiganidan administrator uchun katta qiyinchiliklar keltiradiganigacha turini topishingiz mumkin. Xullas, agar CMS lar bilan ishslash bo'yicha hech qanday tajribangiz bo'lmasa, keraklisini tanlash qiyinlashib ketishi tayin.

Dastlab keling, pullik CMS larni ishlatmaslikka harakat qilamiz. Axir katta pul to'lab, CMS ni ishlatishdan besamara foydalanishga naxojat. Shu yerda bir narsani aytib o'tish kerak: bu bilan o'zimizga qiyin qilamiz. Boshqa tarafdan esa, qanchalik murakkab bo'lsa, shunchalik qiziq bo'lishi aniq, shunday emasmi?

Endi CMS larga yana bir talab qo'yamiz: universallik. Bu degani, biz ishlatmoqchi bo'lgan CMS ixtiyoriy konfiguratsiyadagi kompyuterlarda ham ishlay olishidir. Shuningdek, saytni lokal serverdan haqiqiy serverga ko'chirganimizda yoki bevosita serverning o'zida yaratayotganimizda CMS bilan hech qanday qiyinchiliklar tug'ilmasligi kerak. Bu talabga esa, afsuski, ko'pchilik bepul CMS lar javob bera olmaydi.

Barcha dasturiy ta'minotlarga qo'yiladigan talablardan yana biri – xavfsizlik (axir siz haftalab-oylab yaratgan saytingizni shunchaki qiziqish uchun bir kunda buzishlarini xohlamasangiz kerak?). Shuning uchun CMS larning xavfsizlik darajalari doimiy yangilanib turishi shart.

Natijada, talablarimizga deyarli to'liq javob beradigan quyidagi: PHP Nuke, Drupal, Joomla!, Slaed CMS tizimlarigina qoladi. Endi ana shu CMS larni batafsilroq ko'rib chiqamiz va o'zimizga kerakli ba'zi xulosalar chiqaramiz.

**PHP Nuke.** Eng birinchi CMS lardan biri bo'lib, afsuski, ancha vaqtlardan buyon yangilanmay kelmoqda. Biroq, uning kodi ko'plab adashlari uchun asos bo'lib xizmat qilmoqda (masalan, DotNetNuke, PostNuke). Bu bir-biriga nom jihatdan yaqin bo'lgan tizimlar foydalanuvchi diqqatiga navbatma-navbat chiqib turadi. Funksionallik jihatidan tizim ancha mukammal bo'lib, uning uchun ko'plab qo'shimcha modullar yaratilgan. Masalan, xabarlar lentasi, forum, maqolalar, so'rovlar tashkil qilish, statistika olib borish va boshqalar. Ammo, saytlarning tuzilishi jihatidan bir xillik kuzatiladi. Shu bilan birga, bu tizim yuklanishi qiyin bo'ladi, ya'ni qo'shimcha modullar sahifa yuklanishini sekinlashtiradi. Bu esa ko'plab foydalanuvchilarga yoqmasligi tayin. Yuklanish nafaqat sahifaga, shuningdek, serverga ham tushadi, bu esa xosting xizmatini taqdim etuvchilarga yoqmaydi. Agar sizni shunday vaziyat qoniqtirsa, aynan shu CMS ni tanlashingiz mumkin. Murojaat qilish uchun rasmiy sayti: [www.nukefiles.ru](http://www.nukefiles.ru)

**Drupal.** Turli xil o'zgartirishlar orqali o'zimizga moslab olish bo'yicha eng mukammal CMS deb aytish mumkin. Shu yerda shuni aytish kerakki, "Tomchi" (bu tizim nomi ingliz tilidan aynan shunday tarjima qilinadi) aslida, CMS emas, balki "CMS ni tuzuvchi tizim". Uning yordamida siz tayyor bloklardan kerakli CMS ni yig'asiz. Shuni aytish mumkinki, funksionallik va dizayn jihatidan o'zingizni qanoatlantiruvchi CMS ni hosil qilasiz. Standart modullar to'plamiga blog, forum, xabarlar lentasi, fayllarni yuklash, ovoz berish, qidiruv kabilar kiradi. Bunda ko'rindigan kamchilik bitta: PHP va MySQL ni mukammal bilish lozim. Agar bu qisqartmalarni tushunmagan

bo'lsangiz, keyingi CMS ga o'tavering, aks holda esa bu tizimga jiddiyroq e'tibor bering. Rasmiy sayti: [www.drupal.org](http://www.drupal.org). Texnik yordamni [www.drupal.ru](http://www.drupal.ru) saytidan ham olishingiz mumkin.

**Slaed CMS.** Bugungi kunda ancha ommabop CMS bo'lib, turli xil saytlarda ishlataladi. Ikki xil varianti mavjud: pullik va tekin (OpenSlaed). Ikkovi ham bitta yadroda qurilgan (bu degani xavfsizlik darajasi bir xil) va faqatgina funksionalligiga ta'sir qiluvchi modullar soni bilan farqlanadi. Bepul variantida biror mavzudagi nisbatan kichik portalni yaratishingiz mumkin. Ammo, bundan ortig'ini emas. Dizayni ham unchalik katta qiyinchiliklarsiz o'zgartirilishi mumkin. Tizim aslida Rossiyada yaratilgani uchun rus tilidagi yordam ([slaed.net](http://slaed.net)) haqida qayg'urmasa ham bo'ladi.

**Joomla!** Bugungi kuda eng mukammal dvijoklardan biri (agar bepullari ichida eng mukammali bo'lmasa) bo'lib, uning uchun shunchalik ko'p qo'shimcha modullar yaratilganki, ular yordamida o'zingizning saytingizga deyarli ixtiyoriy funksionallikni berishingiz mumkin. Bu dizaynga ham tegishli bo'lib, uni o'zgartirish uchun ham juda ko'p shablonlar mavjud. Biroq, bu yerda bir kamchilik ham bor: shablonlar tizimi shunday tuzilganki, sahifalarni yaratish va to'ldirish jarayonida sayt bo'limchalari kichik-kichik qismlar bo'lib, ajralib qolishi mimkin. Ya'ni, bu qismlar jadvallar asosida yaratilgan. Bu esa saytingizga o'z shaxsiy ko'rinish berishingizda bir qator noqulayliklar tug'diradi: yuqorida tilga olingan qismchalar kutilmagan joyda chiqib qolib, jahlingizni chiqaradi, shuningdek, ularni yo'qotish uchun vaqtingiz ketadi. Umuman olganda, agar sizni tayyor shablonlar qoniqtirsa (ular esa juda ko'p va xilma-xil), u holda bu tizim sizga juda mos keladi. Agar siz Joomla! ni tanlab o'ziga xos original sayt yaratmoqchi bo'lsangiz ozgina terlashingizga to'g'ri keladi. Rasmiy yordam sayti: [joomla.org](http://joomla.org). Shuningdek, ruscha yordamni norasmiy bo'lsada, ancha yaxshi bo'lgan [joomlaportal.ru](http://joomlaportal.ru) va [joom.ru](http://joom.ru) saytlaridan ham olishingiz mumkin.

Shunday qilib, yuqorida keltirilgan tavsiyalar asosida siz o'z saytingizni tez va oson ravishda yaratishingiz mumkin. Ammo hech qaysi CMS sizga tayyor holdagi saytni taqdim etmaydi. Buning uchun sizdan harakat va mehnat talab etiladi. Yaratayotgan saytingiz boshqalarnikiga o'xshamasligi, dizayn jihatdan takrorlanmas va original bo'lishi uchun esa shaxsiy fantaziyangiz va qobiliyatingizni ishga solishingiz kerak.

## **26-27-ma’ruza. Framework platformasida web-dasturlash. YII2 frameworkka kirish.**

Reja:

1. Framework platformasida web-dasturlash asoslari.
2. YII2 frameworkka kirish. YII frameworkni o‘rnatish.
3. YII2 framework strukturasi. Kirish skriptlari. Model, View, Controller. Ilovalar. Komponentlar. Modullar, filterlar va vidjetlar.

### **Framework platformasida web-dasturlash asoslari**

Hozirgi kunga Respublikamizda kompyuter texnologiyalari, internet kirib bormagan birorta ham shahar yoki aholi punktlari qolmagan. Joylarda qurilayotgan zamonaviy umumta’lim maktablari, kasb-hunar kolejlari va akademik litseylar internet tarmog‘iga ulangan kompyuterlar bilan ta’minlangan. Ushbu ta’lim maskanlarida bilim olayotgan talaba yigit va qizlar katta qiziqish bilan ushbu texnologiyalar asoslarini o‘rganmoqdalar. Davlatimiz rahbari va hukumatimiz tomonidan AKT sohasini rivojlantirishga bo‘lgan diqqat-e’tibor sababli Respublikamizda internet xizmati hajmi, internetdan foydalanuvchilar soni, xalqaro kanallarga chiqish tezligi yildan-yilga o‘sib bormoqda.

Hozirgi vaqtga kelib, yoshlar o‘rtasida Web-dasturlash tillariga qiziqish kuchayib bormoqda. Shu ma’noda, yosh dasturchilar oldida katta Web-loyihalarni amalga oshirish uchun qaysi Frameworkdan foydalanish kerak degan savol ko‘ndalang turadi. Frameworklar nima?

Web-dasturlash sohasida ish yurituvchilar PHP dasturlash tili bilan bir vaqtning o‘zida HTML, CSS, JavaScript kabi dasturlash tillaridan ham foydalanadilar. Biroq so‘nggi 10 yillikka kelib, ish unumdorligini oshirish, vaqt ni maksimal darajada tejash, loyihaning xavfsizligini oshirish, mijozning talabini tez va sifatli bajarish maqsadida web-dasturlash sohasida yangi Framework tushunchasi paydo bo‘ldi.

Framework — bu PHP tilida yozilgan va uning ishlashi MVC konsepsiysi asosida qurilgan (MVC —Model-View-Controller).

Mode — Frameworkning bu qismi MB (ma’lumotlar bazasi)ni ulash, ular bilan ishlashga mo‘ljallangan.

View — saytning bizga ko‘rinib turadigan qismidir.

Controller — Frameworkni nazorat qiladi va chaqirilgan yo‘l bo‘yicha ruxsat oladi (View, Model va boshqa), fayllarni ulash uchun xizmat qiladi.

Framework Web-loyiha (application) uchun ‘karkas’ vazifasini bajaradi. Hozirgi kungacha juda ko‘plab Frameworklar yaratildi, ular mingdan ortiq ([www.blogerator.ru](http://www.blogerator.ru)). Frameworklar kundan-

kun takomillashib bormoqda. PHP Frameworklar asri keldi desak ham bo‘ladi. Aslida, frameworklar evolyutsiyasi qachon boshlangan?

Frameworklarning

paydo

bo‘lishi

Eng birinchi frameworklardan biri — bu CakePHP ([www.blogerator.ru](http://www.blogerator.ru)). Biroq CakePHP o‘sha vaqtida hozirgi Frameworklardan anchagina farq qilar edi. Bu Frameworkda OOP (Object Oriented Programming — Obyektlarga Asoslangan Dasturlash)dan deyarli foydalanimagan. Ular MVC konsepsiya asosida ishlamas edi, shuning uchun ham ortiqcha yuklanish talab etuvchi kodlari ko‘p bo‘lgan. Undan so‘ng 2007-yilda yana bir kuchli — Zend Framework 1.0 yaratildi ([www.blogerator.ru](http://www.blogerator.ru)). Zend Framework 1.0 funksionalligini Framework Zend 1.11 tenglashtirish mumkin. Keyinchalik bu Frameworklardagi kamchiliklar bartaraf etildi, ortiqcha kodlar olib tashlandi. Hozir esa Ajax yordami bilan sahifalarni ortiqcha yuklanishlar va qotib qolishlarsiz yuklab olish imkoniyati tug‘ildi. 2011-yildan Frameworklarning yangi davri boshlandi. Chunki bu paytga kelib, Frameworklar PHP5.3 versiya asosida tuzila boshlandi (PHP 5.3 2009-yil avgust oyida chiqarilgan).

2013-yilga kelib, Frameworklarga talab va ulardan foydalanuvchilar soni yanada oshdi. Fikrimizning isboti sifatida, 2012–2013 yillarda PHP dasturlash tiliga talabning oshganini iyul–avgust oylaridagi o‘zgarishlar misolida ko‘rish mumkin.

Iyul oyi:

| Position Jul 2013 | Position Jul 2012 | Delta in Position | Programming Language | Ratings Jul 2013 | Delta Jul 2012 | Status |
|-------------------|-------------------|-------------------|----------------------|------------------|----------------|--------|
| 1                 | 1                 | =                 | C                    | 17.628%          | -0.70%         | A      |
| 2                 | 2                 | =                 | Java                 | 15.906%          | -0.18%         | A      |
| 3                 | 3                 | =                 | Objective-C          | 10.248%          | +0.91%         | A      |
| 4                 | 4                 | =                 | C++                  | 8.749%           | -0.37%         | A      |
| 5                 | 7                 | ↑↑                | PHP                  | 7.186%           | +2.17%         | A      |
| 6                 | 5                 | ↓                 | C#                   | 6.212%           | -0.46%         | A      |
| 7                 | 6                 | ↓                 | (Visual) Basic       | 4.336%           | -1.36%         | A      |
| 8                 | 8                 | =                 | Python               | 4.035%           | +0.03%         | A      |
| 9                 | 9                 | =                 | Perl                 | 2.148%           | +0.10%         | A      |
| 10                | 11                | ↑                 | JavaScript           | 1.844%           | +0.39%         | A      |

Avgust oyi:

| Position Aug 2013 | Position Aug 2012 | Delta in Position | Programming Language | Ratings Aug 2013 | Delta Aug 2012 | Status |
|-------------------|-------------------|-------------------|----------------------|------------------|----------------|--------|
| 1                 | 2                 | ↑                 | Java                 | 15.978%          | -0.37%         | A      |
| 2                 | 1                 | ↓                 | C                    | 15.974%          | -2.96%         | A      |
| 3                 | 4                 | ↑                 | C++                  | 9.371%           | +0.04%         | A      |
| 4                 | 3                 | ↓                 | Objective-C          | 8.082%           | -1.46%         | A      |
| 5                 | 6                 | ↑                 | PHP                  | 6.694%           | +1.17%         | A      |
| 6                 | 5                 | ↓                 | C#                   | 6.117%           | -0.47%         | A      |
| 7                 | 7                 | =                 | (Visual) Basic       | 3.873%           | -1.46%         | A      |
| 8                 | 8                 | =                 | Python               | 3.603%           | -0.27%         | A      |
| 9                 | 11                | ↑↑                | JavaScript           | 2.093%           | +0.73%         | A      |
| 10                | 10                | =                 | Ruby                 | 2.067%           | +0.38%         | A      |

([www.blogerator.ru](http://www.blogerator.ru))



PHP dasturlash tiliga talabning ko‘payishi esa o‘z navbatida, PHP Frameworklarning qulayligi, ishonchliligi, funksionalligi oshishiga sabab bo‘ldi.

2014-yil avgust oyining oxirlariga kelib, PHP 6.0 versiya yaratildi ([www.php.net](http://www.php.net)). Bu esa web-loyihalarni yaratishdagi xatoliklarni kamaytirish va ish unumdorligini oshirish uchun Frameworklarni yanada takomillashtirish imkonini yaratdi.

Hozirda Web-dasturchilar tomonidan CodeIgniter, Yii, Medoo, Flight, Pop PHP, Laravel, Symfony, Zend, CakePHP, Kajona va boshqa ko‘plab qulay va ko‘p funksiyali PHP Frameworklardan foydalanimoqda.

CodeIgniter — eng ko‘p qo‘llaniladigan MVC tizimli Frameworklardan biri. EllisLab kompaniyasi, Paul Burdick va Rick Ellis tomonidan yaratilgan. Birinchi marotaba 2006-yil 28-fevralda ommaga havola etilgan. 2011-yil 28-yanvarga kelib, CodeIgniter2.0 yaratildi. Bu Framework ikki bo‘limli: CodeIgniter Core va CodeIgniterReactor ([www.ru.wikipedia.org](http://www.ru.wikipedia.org))dan iborat.

2014-yil 6-mayda CodeIgniter2.2.0 versiya yaratildi ([www.ellislab.com](http://www.ellislab.com)).

Mazkur versiya boshqa frameworklardan ko‘ra ancha tez va yengil hisoblanadi. MySql, PostgreSQL, MSSQL, SQLite, Oracle kabi ma’lumotlar bazasini quvvatlaydi. O‘rganish juda oson. Sozlash va o‘rnatish uchun ko‘p vaqt talab qilmaydi. Deyarli mikroframework desak ham bo‘ladi. Biroq ma’lumotlarni bazadan Modelga avtomatik generatsiya qilsh imkonи mavjud emas ([www.uzmob.net](http://www.uzmob.net)).



Yii Frameworkning asoschisi Qiang Xue bo‘lib, u Yii Frameworkni 2008-yil 1-yanvarda yaratdi. Oradan bir yil o‘tib, dekabr oyida Yii framework 1.0 ni ommaga havola qildi. Yii’ni yaratishdan oldin u ko‘p ishlataladigan barcha frameworklarni o‘rganib chiqdi. Natijada, qolgan Frameworklarda yo‘l qo‘yilgan kamchiklarni o‘z Frameworkida bartaraf etdi. Hozirda Yii Framework’ga talab kuchli va u o‘z o‘rnida boshqa Frameworklar bilan raqobatlasha, ulardan bemalol ustunlik qila oladi ham. Yii yuqori samarador, komponentlarga asoslangan PHP framework. Yii Web-dasturlarni yaratishni sezilarli darajada tezlashtiradi ([www.yiiframework.com](http://www.yiiframework.com)).

Yii — Yes It Is (ha bu o‘sha) degan so‘zning qisqartmasidir. Yii’da yaratilgan Web-dasturlarning ishlashi uchun kamida PHP 5.1.0 versiyasi o‘rnatilgan Web server kerak bo‘ladi. Bundan tashqari, Yii’da ishslash uchun OOP (Object Oriented Programming — Obyektlarga asoslangan dasturlash) ni tushunish dasturchi uchun juda muhim, chunki Yii to‘liq OOP’ga asoslangan Frameworkdir. Yii umumiy Framework hisoblangani uchun ham u yordamida ixtiyoriy web-dasturlarini yaratish mumkin. Yii yengil, ko‘p resurs talab qilmaganligi va murakkab keshlash mexanizmlariga ega bo‘lganligi uchun ham katta trafikli dasturlarni, masalan, portallar, forumlar, kontentni boshqarish tizimlari (CMS), elektron kommersiyani yaratishga mos keladi ([www.opennet.uz](http://www.opennet.uz)).

The screenshot shows the official website for the Yii PHP Framework. At the top, there's a navigation bar with links for demos, guide, class reference, wiki, extensions, forum, live chat, and login. Below the header is the Yii logo and the tagline "The Fast, Secure and Professional PHP Framework". A section titled "Three steps to build your application rapidly:" lists three steps: 1. You create the database; 2. Yii generates the base PHP code; 3. You customize the code to fit your exact needs. There are two buttons: "Take the Tour" and "Download Yii". Below this, there are three icons labeled "Fast", "Secure", and "Professional", each with a brief description. The "Fast" icon says "Yii only loads the features that you need. It has powerful caching support. It is explicitly designed to work efficiently with AJAX.". The "Secure" icon says "Security comes as standard with Yii. It includes input validation, output filtering, SQL injection and Cross-site scripting prevention.". The "Professional" icon says "Yii helps you develop clean and reusable code. It follows the MVC pattern, ensuring a clear separation of logic and presentation.". On the left side, there's a "Latest News" section with three entries: "Yii 2.0 alpha is released" (Dec 1, 2013), "We are very pleased to announce the alpha release of Yii Framework version 2. You can download it from [yiiframework.com](#).", "Yii 1.1.14 is released" (Aug 11, 2013), and "We are very pleased to announce that Yii Framework version 1.1.14 is released. You can download it at [yiiframework.com/download](#).", followed by a link to "Website deface details". On the right side, there's a section titled "Modelling Use Cases & UML" with a link to "www.adaptevents.nl" and a description of "Smart use cases in de praktijk Sander Hoogendoorn, 15-16 april, NL".



Zend — juda mustahkam, dasturchidan OOP chuqur bilishi talab qiladigan ko‘plab modul va kutubxonalarga ega, JS framework, DOJO va uning kutubxonasi o‘rnatilgan. Ixtiyoriy hajmdagi Web loyihalarni yaratish imkonini beradi. 2014-yil 15-aprelda yangi ZendFramework 2.3.1 versiyasi ishlab chiqildi.



Symfony — 2005-yilda ishlab chiqilgan Framework bo‘lib, juda kuchli MVC Framework hisoblanadi. Dasturchilar orasida anchagina ommalashgan. 2011-yil lider Frameworklar qatoriga kirgan va hozirgacha mashhurligini yoqotmagan ([www.blogerator.ru](http://www.blogerator.ru)).



Medoo — bu Framework boshqalaridan farqli ravishda, eng yengil (hajm jihatdan kichik) Framework hisoblanadi. Modooning xotira hajmi bor-yo‘g‘i 10,9 Kb. Uning barcha imkoniyatlari shu hajmli birgina fayldan tashkil topgan. Juda ajoyib mikroframework va miniloyihalardir ([www.w3talks.org.ru](http://www.w3talks.org.ru)).

POP php — har qanday Framework ham o‘ziga yarasha murakkab, o‘rganish qiyin. Biroq POP Php ish tajribasidan qat’iy nazar (professional dasturchi yoki havaskor), har qanday dasturchi va qiziquvchilarga mo‘ljallangan. O‘rganish va loyihani yaratish boshqa Frameworklarga nisbatan oson bo‘lgan, ochiq kodli Frameworkdir ([www.w3talks.org.ru](http://www.w3talks.org.ru)).

The Lightest PHP database framework to accelerate development

Version: 0.9 [10.9KB]

DOWNLOAD

Love Medoo?

134 Likes 89 Tweets 81+1 Shares 441 Stars 140 Forks

#### WHY MEDOO

Be a PHP Ninja.

DOWNLOAD THE LATEST VERSION

POW!

Who can use Pop PHP?

Pop is built for users of all levels. Some PHP frameworks are pretty complex and intense. They can intimidate the beginner and even confuse the moderate PHP developer. Pop can easily be used by anyone with any experience level.

Come one, come all. Pop has a manageable learning curve to help beginners get their feet wet with a PHP framework, but also offers robust and powerful features for the more advanced PHP developer.

Why use Pop PHP?

Pop promotes today's standards in PHP. Pop keeps up with the current standards in the PHP community. It's a fully object-oriented framework that can easily be scaled to accommodate an application of any size. Currently, Pop supports PHP 5.3+.

Lightweight and fast. As previously mentioned, some PHP frameworks are pretty heavy, and that in and of itself can be a deterrent. All of Pop's many robust components and features weigh in at less than 2 MB.

Contact | Terms | Privacy | License

© 2009 - 2014 Merc 10 Media, LLC. All Rights Reserved.

Laravel — deyarli barcha professional Web-dasturchilar bu Framework haqida xabardor. Hozirgi kunda eng ommabop Frameworklardan biri bo‘lgan Laraver dastlab yaratilgan versiyalari MVC konsepsiya asosida ishlamagan. Ma’lumotlar bazasi bilan ishslash, kesh bilan operatsiyalar, sessiyalar, modul va kutubxonalar, sodda mexanizmlar formalar va HTML kodlar ham ORM (Object Relational Mapping) struktura bo‘yicha ish 2011-yil 24-noyabrga kelib, 2-versiya, 2013-yil 28-mayda esa Laravel 4 yaratildi.

Laravel - The PHP framework

laravel.com

# THE PHP FRAMEWORK FOR WEB ARTISANS.

YOUR LIFE AS A DEVELOPER JUST GOT A WHOLE LOT EASIER.

QUICKSTART    LARACon 2014

```

1 /**
2 * Display the specified resource.
3 *
4 * @param int $id
5 * @return Response
6 */
7 public function show($id)
8 {
9 $post = Post::with('comments')->where('id', $id)->first();
10
11 return View::make('post.show')->with('post', $post);

```

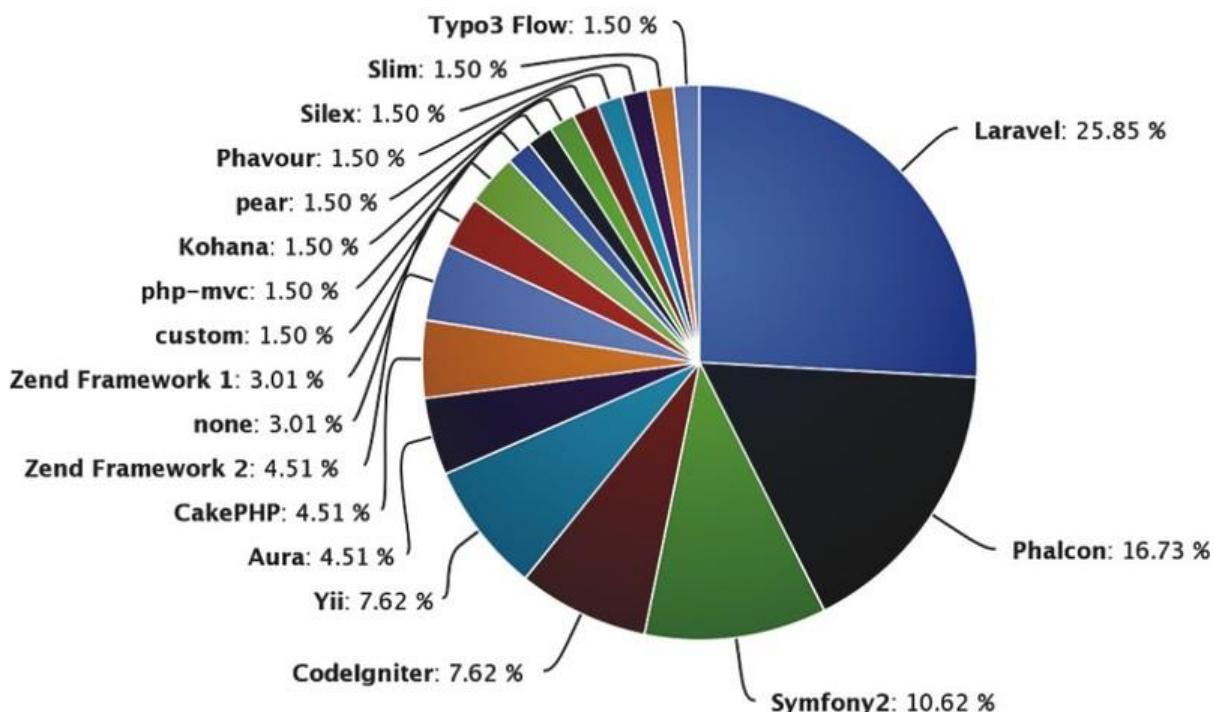
LARAVEL    WELCOME DOCUMENTATION API GITHUB FORUMS TWITTER

**RESTful Routing** Use simple Closures to respond to requests to your application. It

**Command Your Data** Ships with the amazing Eloquent ORM and a great migration system.

**Beautiful Templating** Use native PHP or the light-weight Blade templating engine. Blade

**Ready For Tomorrow** Build huge enterprise applications, or simple JSON APIs. Write powerful

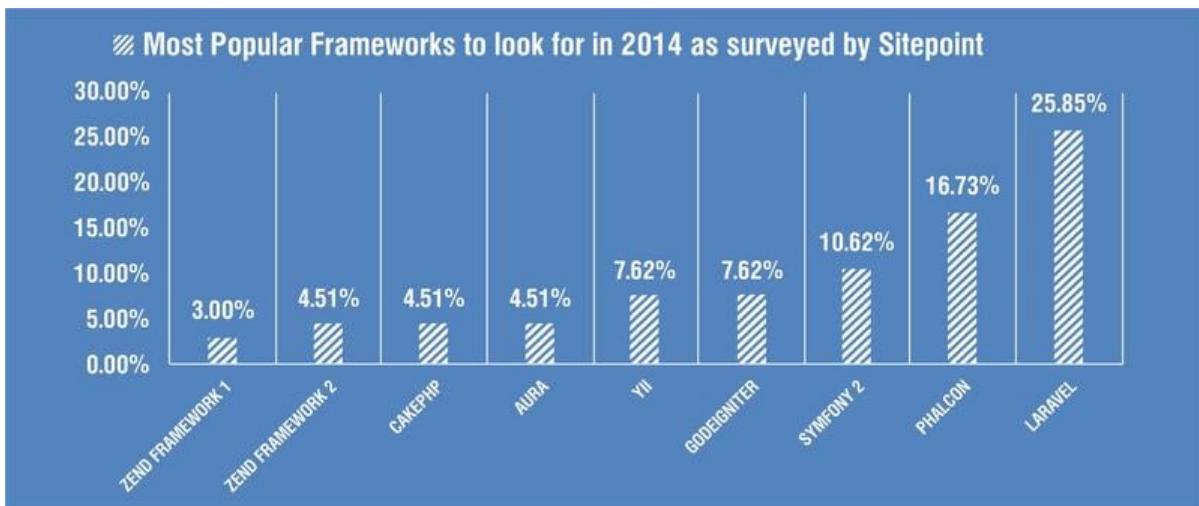


Ommabop

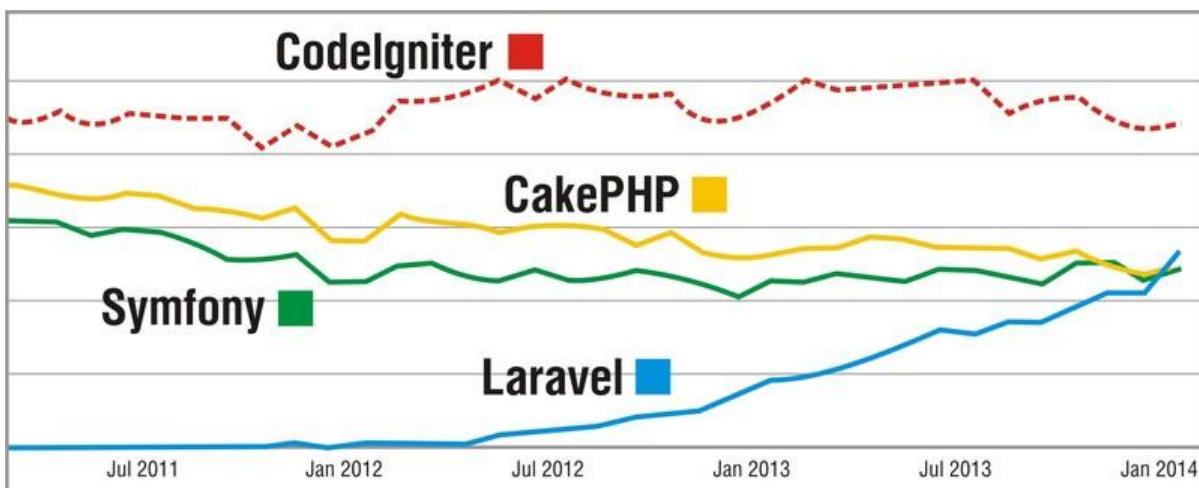
([www.highchart.com](http://www.highchart.com))

frameworklar

2013-yil



Ommabop Frameworklar 2014-yil



Eng ommabop 5ta Frameworklar

| (www.phpframeworks.com)         | Zend F                                                | Codeigniter                                    | Kohana         | Symfony                          | Yii 1.1                                                                                                                  |
|---------------------------------|-------------------------------------------------------|------------------------------------------------|----------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Ta'rif                          | PHP va OOPni bilishni talab etadi. Ishonchli, yengil. | Deyarli mikroframework. Juda yengil, mustahkam | Tez, ishonchli | Kuchli himoya, view tizimi qulay | Zend va Symfony dan o'rnatishni nisbatan osonroq. View tizimi yaxshi, yengil, ixtiyorli Web-dasturlarini yaratish mumkin |
| Ishlash rejimi                  | MVC, OOP                                              | MVC                                            | MVC            | MVC                              | MVC, OOP                                                                                                                 |
| PHP5.2                          | Ha                                                    | Ha                                             | Ha             | Ha                               | Ha                                                                                                                       |
| O'rnatishning qiyinlik darajasi | Past                                                  | Past                                           | Past           | Yuqori                           | O'rtacha                                                                                                                 |
| Sozlash darajasi                | Ko'p                                                  | Ko'p emas                                      | Ko'p emas      | Ko'p                             | Ko'p emas                                                                                                                |

### YII2 frameworkka kirish. YII frameworkni o'rnatish.

Yii – bu ko'p tarmoqli web ilovalar qurish uchun yuqori sifatli komponentlar asosidagi PHP – framework strukturasidir. U katta kodlardan qayta va qayta foydalanishda moslashuvchanlikni va web ilovani tezkoriliginini oshirishiga imkon beradi. Qisqa qilib aytganda oddiy, samarali va keng

Keling PHP frameworklarning o`zaro imkoniyatlarini chamalangan holda ko`rib chiqaylik.

| PHP framework                 | PHP_4 | PHP_5 | MVC | Multiple DB's | ORM | DB Objects | Templates | Caching | Validation | Ajax | Auth Module | Modules | EDP |
|-------------------------------|-------|-------|-----|---------------|-----|------------|-----------|---------|------------|------|-------------|---------|-----|
| <a href="#">Akelos</a>        | +     | +     | +   | +             | +   | +          | +         | +       | +          | +    | +           | +       | -   |
| <a href="#">ash.MVC</a>       | -     | +     | +   | -             | -   | +          | +         | -       | +          | -    | +           | +       | -   |
| <a href="#">Cake PHP</a>      | +     | +     | +   | +             | +   | +          | -         | +       | +          | +    | +           | +       | -   |
| <a href="#">Codeigniter</a>   | +     | +     | +   | +             | -   | +          | +         | +       | +          | -    | -           | -       | -   |
| <a href="#">DIY</a>           | -     | +     | +   | -             | +   | +          | +         | +       | -          | +    | -           | -       | -   |
| <a href="#">eZ components</a> | -     | +     | -   | +             | -   | +          | +         | +       | +          | -    | -           | -       | -   |
| <a href="#">Fusebox</a>       | +     | +     | +   | +             | -   | -          | -         | +       | -          | +    | -           | +       | -   |
| <a href="#">PHP on TRAX</a>   | -     | +     | +   | +             | +   | +          | -         | -       | +          | +    | -           | +       | -   |
| <a href="#">PHPDevShell</a>   | -     | +     | +   | -             | +   | +          | +         | +       | +          | +    | +           | +       | -   |
| <a href="#">PHP Openbiz</a>   | -     | +     | +   | +             | +   | +          | +         | -       | +          | +    | +           | -       | -   |
| <a href="#">Prado</a>         | -     | +     | +   | +             | +   | +          | +         | +       | +          | +    | +           | +       | +   |
| <a href="#">QPHP</a>          | +     | +     | +   | +             | -   | +          | +         | -       | +          | +    | +           | +       | +   |
| <a href="#">Seagull</a>       | +     | +     | +   | +             | +   | +          | +         | +       | +          | +    | +           | +       | -   |
| <a href="#">Symfony</a>       | -     | +     | +   | +             | +   | +          | -         | +       | +          | +    | +           | +       | -   |
| <a href="#">WACT</a>          | +     | +     | +   | +             | -   | +          | +         | -       | +          | -    | -           | +       | -   |
| <a href="#">WASP</a>          | -     | +     | +   | -             | -   | +          | +         | -       | +          | +    | +           | +       | -   |
| <a href="#">Yii</a>           | -     | +     | +   | +             | +   | +          | +         | +       | +          | +    | +           | +       | +   |
| <a href="#">Zend</a>          | -     | +     | +   | +             | +   | +          | +         | +       | +          | +    | +           | +       | -   |
| <a href="#">ZooP</a>          | +     | +     | +   | +             | -   | +          | +         | +       | +          | +    | +           | -       | -   |

**MVC:** Model-View-Controller arxitekturasi bo`yicha qurish imkoniyatiga ega.

**Multiple DB's:** Bir ma`lumotlar omboridan boshqa bir qancha ma`lumotlar omboriga o`zgartirishlarsiz amalga oshirilishi xususiyatiga ega.

**ORM:** ActiveRecord da ma`lumotlar obyekti bog`lanishi arxitekturasini amalga oshirish.

**DB Objects:** Ma`lumotlar ombori bilan ishlash xususiyatiga ega. ( Insert, Delete, Update kabilar)

**Templates:** Qurilgan shablonlar mexanizmi frameworkiga ega.

**Caching:** Berilgan obyektlarni keshlash yoki shu kabi boshqa uslubdagi keshlashga ega.

**Validation:** Ma`lumotlar oqimini tekshirish yoki filrlash xususiyatiga ega.

**Ajax:** Frameworkda ajax tehnologiyasi orqali ishlash xususiyatiga ega.

**Auth Module:** Foydalanuvchilarni authentication qilishda o`zining maxsus moduliga ega.

**Modules:** Frameworkning o`zida maxsus komponentlarga ega (misol uchun: PDF moduli, RSS moduli). Ya`ni foydali komponentlar jamlanmasi

**EDP:** Event Driven Programming. Eventlar bilan ishslash xususiyatiga ega. Misol uchun jquery kutubxonasining click, dblclick eventlari kabi.

## YII frameworkni o'rnatish.

Yii framework haqida qisqacha to'xtalib o'tib uni windows os muhitida denwer paketiga qanday qilib o'rnatish kerakligi haqida ma'lumotlar keltiramiz. Stiv Jobsni bir gapi bor ediku "Har doim yangi texnologiyadan foydalan yoki foydalanmasang ham oldingda tursin". Biz ham dasturlashda har doim yangi texnologiyalardan foydalanishga harakat qilamiz.

Hozirda Yii framework ga talab kuchli va u o'z o'rnidida boshqa frameworklar bilan raqobatlasha oladi va ulardan ustunlik ham qila oladi.

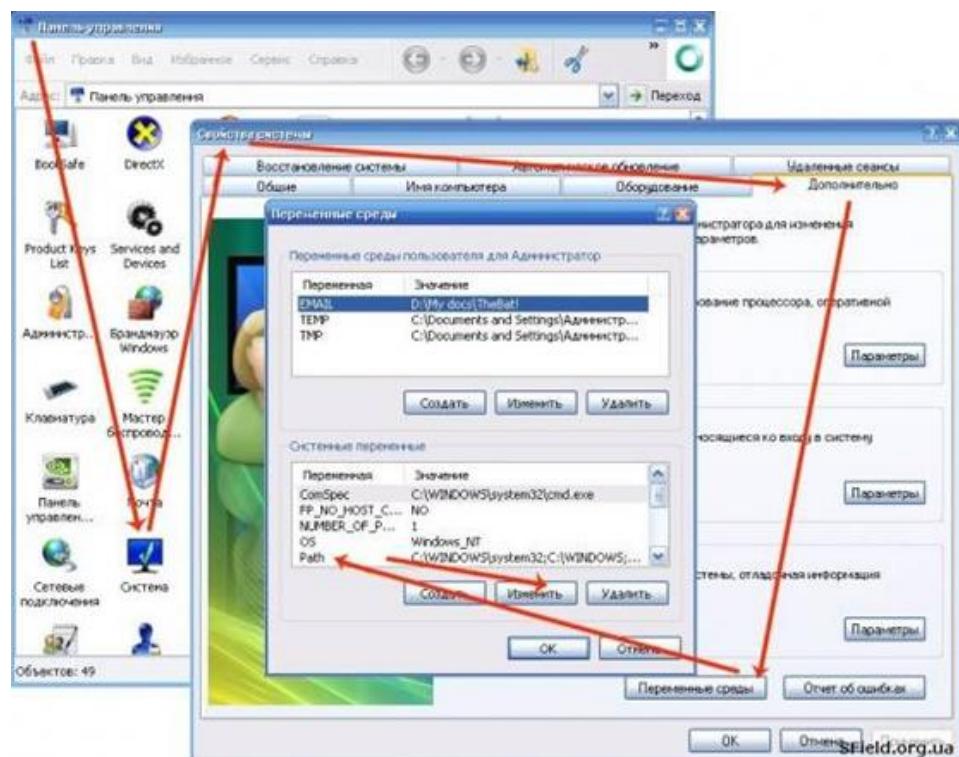
Buning uchun avval Denwer oldindan o'rnatilgan bo'lishi kerak.

Yii frameworkni o'rnatish uchun bir qancha qadamlarni bosib o'tamiz.

1. Windowsning o'zgaruvchilar satriga php o'rnatilgan joyni ko'rsatamiz.

**Пуск->Панель управления->Система->Вкладка “Дополнительно”**

"Переменные среды" tugmasini bosamiz va ochilgan oynadan ikkinchi **Системные переменные** bo'limidan **Path** ni topib **Изменить** tugmasini bosamiz.



Va satrning oxiriga ";Z:\usr\local\php5" ni joylashtiramiz.

1. Endi esa php ni sozlaymiz. Php ning sozlarnmali (Siz o'rnatgan manzilada) c:\WebServers\usr\local\php5\ bu yerda saqlanadi. U yerda php.ini faylini tahrirlash uchun ochamiz (Notepad++, Sublime).

Uchbu satrni topamiz: extension\_dir = "/usr/local/php5/ext" va uni extension\_dir = "Z:\usr\local\php5\ext" ga o'zgartiramiz.

Sozlashda davom etamiz, "session.save\_path = "/tmp"" ushbu satrni topamiz va uni quydagicha session.save\_path = "Z:\tmp" o'zgartiramiz.

Php sozlanmali ham tayyor bo'ldi.

Endi loyiha yaratishga o'tamiz.

1. **bu yerdan** oxirgi relizini yuklab olamiz.
2. **C:\WebServers\home\yii\www** karkasimiz uchun papka hosil qilamiz.
3. Yuklab olgan Yii framework paketini tekshiramiz uning ichida quyudagi 3 ta **demos**, **framework**, **requirements** papkalari mavjud bo'lsa quyidagi manzil yaratib **C:\WebServers\home\yii\YiiRoot** arxivdan chiqarimz.
4. **D:\WebServers\home\yii\www\.htaccess** ushbu faylni yaratamiz va unga **AddDefaultCharset utf-8** ni kiritamiz. Sababi hosil bo'layotgan loyiha kodirovkasi CP1251 da biz esa standart UTF-8 dan foydalanamiz.
5. Windows consolini ochamiz: **Пуск->Выполнить->cmd**
6. Qydagisi yo'l bo'yicha boramiz: **cd \WebServers\home\yii** va bizga quydagisi **C:\WebServers\home\yii> hosil bo'ladi.**
7. Quydagicha buyruq beramiz **C:\WebServers\home\yii>YiiRoot\framework\yiic webapp www** va siz bu buyruqni to'g'ri bajarsangiz quydagisi oyna hosil bo'ladi.

PHP Warning: Directive 'register\_long\_arrays' is deprecated in PHP 5.3 and greater in Unknown on line 0

PHP Warning: Directive 'magic\_quotes\_gpc' is deprecated in PHP 5.3 and greater in Unknown on line 0

PHP Warning: PHP Startup: PDO: Unable to initialize module

Module compiled with module API=20060613

PHP compiled with module API=20090626

These options need to match

in Unknown on line 0

Create a Web application under 'C:\WebServers\home\yii\www'? [Yes|No]

**Yes** deb javob beramiz va yuklash jarayoni boshlanadi.

So'ngida quydagicha xabar hosil bo'ladi:

**Your application has been created successfully under Z:\home\yii\www.**

<http://yii> quydagি address da Yii frameworkimiz tayyor!

### **YII2 framework strukturasi. Kirish skriptlari.**

Biz odatda kattagina ish qilayotganimizda bir hil funksiyalar yozishimizga tog`ri keladi va bu ishni qayta qayta takrorlanishiga duch kelamiz. Misol uchun biror bir joyda 2 ta DateTime ning o`zaro ayirish masalasi bo`lsin. Bu ishni bir nechta joyda ishlataladigan bo`lsa unda xar bir shu ishni qiladigan funksiyani yozib ketaveramiz. Uning o`rniga bir yordamchi funksiya sifatida bir joyda saqlasak o`sha funksiyani kerak bo`lganda o`sha nom ostida chaqirishimiz mumkin bo`ladi. Bunda funksiyani har safar yozib o`tirish shart bo`lmaydi. Bunda har bitta dasturlash tilining bunday imkoniyati mavjud. Shu kabi yii da ham bu imkoniyat mavjud. Keling aslida funksiyani qanday yaratish va undan qanday foydalanish haqida gaplashsak. Yordamchi classning ko`rinishi

```
<?php
// protected/components/MyClass.php

class MyClass {
 ...
}
```

Ushbu holat Classning standart ko`rinishi: Bunda ushbu faylni "protected/components/MyClass.php"

Ko`rinishdagi joyga saqlash kerak bo`ladi. Ya`ni "components" papkasining ichiga.

Yordamchi funksiya ushbu ko`rinishda elon qilinadi

```

<?php
// protected/components/Helpers.php NOT RECOMMENDED!

class Helpers {

 // does the first string start with the second?
 public static function startsWith($haystack, $needle)
 {
 return strpos($haystack, $needle) === 0;
 }

 ...
}

```

Bunda birinchi bo`lib class nomi e`lon qilinmoqdi keyin ehtiyojdagi funksiya yozilmoqda. Shu yordamchi funksiyani chaqirish yo`li quyidagicha

```

...
if (Helpers::startsWith(...))

```

Ya`ni class nomi , funksiya va funksiyaga beriladigan qiymat.

Endi yordamchi fayl e`lon qilish va uni ishlatish usulini bir kichik misol bilan ko`rib chiqamiz.

Yordamchi fayl nima? Yordamchi fayl bu helpers.php nomi ostida yaratiladigan fayl bo`lib unda bir qancha funksiyalar yozishingiz mumkin bo`ladi.

"protected/components/helpers.php"

```
<?php
```

```
function time_difference($t1, $t2)
```

```
{
```

```
list($hh1,$mm1,$ss1) = explode(':', $t1);
```

```
list($hh2,$mm2,$ss2) = explode(':', $t2);
```

```
$time2 = mktime((int)$hh2, (int)$mm2, (int)$ss2);
```

```
$time1 = mktime((int)$hh1, (int)$mm1, (int)$ss1);
```

```

$duration = $time2 >= $time1 ? $time2 - $time1 : $time2 + 86400 - $time1;

$hh3 = floor($duration / 3600);

$mm3 = floor(($duration - $hh3 * 3600) / 60);

$ss3 = $duration - 3600 * $hh3 - 60 * $mm3;

$result = date('H:i:s', mktime($hh3, $mm3, $ss3));

return $result;

}

```

Ko`rib turganingizdek bu yerda class nomi yo`q shunchaki funksiya yaratilib ketmoqda. Bu holatda hozir biz ikkita vaqt orasidagi vaqtini topmoqdamiz

Bu faylni ham odatdagidek "components" papkasiga tashlanadi va nomini "helpers.php" saqlash kerak bo`ladi. Endi bu funksiyani quyidagicha chaqiriladi.

Ko`rib turganingizdek bu yerda class nomi yo`q shunchaki funksiya yaratilib ketmoqda. Bu holatda hozir biz ikkita vaqt orasidagi vaqtini topmoqdamiz

Bu faylni ham odatdagidek "components" papkasiga tashlanadi va nomini "helpers.php" saqlash kerak bo`ladi. Endi bu funksiyani quyidagicha chaqiriladi.

```
<?php echo time_difference($date1,$date2); ?>
```

Bu ishimiz xozirda to`g`ridan to`g`ri ishlamaydi. Chunki bu funksiyani qayerdan olishni bilmaydi

Shuning uchun quyidagicha ish qilamiz:

```

<?php
// protected/config/main.php

require_once(dirname(__FILE__) . '/../components/helpers.php');

...

```

Config papkamizdagi main.php faylimizga yuqoridagi yo`lni ko`rsatib qo`yishimiz kerak bo`ladi.

### **Model, View, Controller.**

Yiining Gii generatoridan foydalanib model va controller hosil qilishni ko`rib chiqamiz.

Buning uchun biz birinchi navbatda "yiitest" degan ma`lumotlar ombori yaratamiz va uning ichida posts degan jadval yaratamiz. Agarda o`zimizning ma`lumotlar omborimiz bo`lsa unda o`shaning ichida jadval hosil qilamiz. Bu jadvalimiz o`z ichiga 3 ta ustunni oladi. Bular id, title va content.

```
CREATE TABLE posts (
 id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
 title VARCHAR(100),
 content TEXT
);
```

Keyin web ilovangizdagi protected/config/main.php fayldagi ushbu kodni yozamiz. Bu kod odatda izoh holatida bo`ladi.

```
'db'=>array(
 'connectionString' => 'mysql:host=localhost;dbname=testdrive',
 'emulatePrepare' => true,
 'username' => 'root',
 'password' => '',
 'charset' => 'utf8',
),
```

dbname dagi "testdrive" o`rniga siz o`zingiz yaratgan ma`lumotlar ombori nomini yozasiz, ya`ni "yiitest". Yuqoridagi kodimiz ma`lumotlar ombori bilan bog`lanish imkonini yaratadi.

Ma`lumotlar omboridagi har bir jadvalga mos yiida CActiveRecord classi ostida model yaratilishi maqsadga muvofiqdir. Jadvalning modeli bo`lmay turib u bilan ishlab ketib bo`lmaydi (ya`ni qiymatlar olish, berish kabi ishlar). Yiida Gii tehnologiyasidan foydalanishda "main" faylimizdagi quyidagicha kodni izohdan chiqarishimiz kerak bo`ladi.

```
'gii'=>array(
 'class'=>'system.gii.GiiModule',
 'password'=>'your password to access gii',
```

```
'ipFilters'=>array('127.0.0.1','::1'),
,
```

'password' - Gii texnologiyasiga kirishda berilgan password orqali bajariladi.

'ipFilters' - ip bo`yicha kirish xuquqi (bu yerda local ip, o`zingizgagina kirish huquqi).

Kodlarimizni sozlaganimizdan keyin ushbu havola: <http://localhost/yiitest/index.php?r=gii>

bo`yicha Gii modulimizga kirishimiz mumkin. Belgilangan password bo`yicha kirasiz.

### **Yiida model hosil qilish.**

**Model** - foydalanuvchi tomonidan kiritilayotgan ma`lumotlarni doimo rolda nazorat qilib boradi. Biz modelni ikki xil ko`rinishda hosil qilishimiz mumkin. Biz ma`lumotlarni olish, qayta ishslash, keyin o`chirish uchun CFormModel dan foydalanamiz; agar ma`lumotni olib uni bazada saqlash bo`ladigan bo`lsa CActiveRecord dan foydalaniladi. Ikkala tur ham bir CModel classi asosida ishlaydi.

" **attribute**" - termini classning ayrim xususiyatlarini boshqa xususiyatlaridan ajratish uchun yaratilgan. Bunday xususiyat (attribut) asosan o`zida ma`lumotlar omboridan olingan yoki foydalanuvchi tomonidan kiritilgan ma`lumotni saqlashda qo`llaniladi.

### **Model classini aniqlaymiz:**

Quyida biz LoginForm modelidan foydalanib, foydalanuvchining ro`yhatdan o`tish pageni yaratamiz.

```
class LoginForm extends CFormModel {

 public $username;

 public $password;

 public $rememberMe=false;

}
```

Biz uchta o`zgaruvchi e`lon qilamizi, bular: \$username, \$password va \$rememberMe. Ular foydalanuvchi nomini, passwordini va shu bilan birgalikda username va passwordni eslab qolish belgisini (\$rememberMe). Bu belgi boshida **false** qiymatga ega bo`lmoqda.

Tekshiruv qoidalarini aniqlash:

Foydalanuvchi formadan ma`lumotlarni yuborganda, model ularni oladi, biz ularni amalga oshirayotganda mal`umotlar to`g`ri ekanligiga ishonch hosil qilishimiz kerak. Bu ishni amalga oshirayotganda qoidalar to`plami bilan ma`lumotlarni tekshiramiz. Tekshirish qoidalarini amalga oshirshda bizga rules() metodi yordam beradi.

```
class LoginForm extends CFormModel {

 public $username;

 public $password;

 public $rememberMe=false;

 private $_identity;

 public function rules()

 {

 return array(

 array('username, password', 'required'),

 array('rememberMe', 'boolean'),

 array('password', 'authenticate'),

);

 }

 public function authenticate($attribute,$params)

 {

 $this->_identity=new UserIdentity($this->username,$this->password); if(!$this->_identity->authenticate())

 $this->addError('password','Неверное имя пользователя или пароль.');//

 }

}
```

Quyida ko`rsatilgan kodimizda username, password - majburiy to`ldirilishi kerak bo`lgan maydonlar, password maydon yana username ga mos kelishi ham tekshirilmoqda. rememberMe true yoki false qiymat qaytaradigan o`zgaruvchi.

**Bu yerda AttributeList da attribute nomlari " , " bilan ajratilib yozilad;**

1.1.11 versiyasidan boshlab alohida qoidalarni inkor qilish mumkin. Agar qandaydir qoida va senariya uchun validatsiyadan o`tkazishni hohlamasangiz " except " parametrli senariya nomiga ko`rsatib qo`yishingiz kerak bo`ladi.

Senariyalar ro`yhatidan ikkita ekvivalent usulni ko`rsatishningiz mumkin.

```
// senariya nomlari massiv ko`rinishida
'on'=>array('update', 'create'),
// nomlar satrlar bilan, " , " ajratilgan holda (probelni qabul qilmaydi, faqat " , ")
'except'=>'ignore, this, scenarios, at-all',
```

Qoidalarni asosida tekshiruvchi validatorni 3 xil usul bilan ko`rsatish mumkin.

Birinchidan, model classi ichida o`zingiz funksiya yaratib uni validatsiyaga bog`lab qo`yishingiz mumkin, ya`ni bunda o`zingiz qoidalarni kiritishingiz mumkin.:

```
/**
 * @param string $attribute maydon nomi, validatsiyadan o`tuvchi qiymatlar
 * @param array $params validatsiya qoidasi uchun qo`shimcha parametrlar
 */
public function ValidatorName($attribute,$params) { ... }
```

Ikkinci usul bu - xaqiqiy class nomida validatorni ko`rsatish. Bu holatda ma`lumotlarni tekshirish uchun tekshiruvchi class nusxasi hosil qilinishi paytda qoida qabul qilinadi. Uning attribut belgilarini tayyorlash uchun qoidadagi qo`shimcha parametrlardan foydalilanadi. Tekshiruv classini [CValidator](#) classi asosida ishlab chiqariladi.

Uchinchi usul - bu yiining maxsus classlaridan bo`lib, ko`p uchraydigan validatorlarni aniqlab qo`ygan. Misol uchun " required " bu - [CRequiredValidator](#) classining laqabi bo`lib, shu laqab orqali ishlatalib ketiladi. Bu class kelayotgan ma`lumotni mavjudmi yoki mavjudmasligini aniqlash uchun ishlataladi. Agar siz " required "ni ishlatsangiz, shunga biriktirilgan hamma attributelarni majburiy to`ldiriladigan maydon qilib chiqadi.

Yii ning qolgan validator classlarini ko`rib chiqaylik.:

- boolean: [CBooleanValidator](#) classning laqabi bo`lib, u teng bo`lishi mumkin true yoki false ga;
- captcha: [CCaptchaValidator](#) classning laqabi bo`lib, kaptchaning belgilari bilan to`g`riligini tekshiradi;
- compare: [CCompareValidator](#) classning laqabi bo`lib, ikkita attributeni solishtiradi;
- email: [CEmailValidator](#) classning laqabi bo`lib, kelayotgan email manzilni haqiqatda email manzilmi yoki yo`qligini aniqlidi;
- date: [CDateValidator](#) classning laqabi bo`lib, joriy attributning dateligini to`g`riligini tekshiradi;
- default: [CDefaultValueValidator](#) classning laqabi bo`lib, joriy tanlangan attributlar bo`yicha belgilarni o`zlashtirish;
- exist: [CExistValidator](#) classning laqabi bo`lib, ko`rsatilgan jadvaldagi ustunning attributida belgi borligini aniqlidi;
- filter: [CFilterValidator](#) classning laqabi bo`lib, o`zgartiriladigan attribut belgisining filtr bilan bajarish;
- in: [CRangeValidator](#) classning laqabi bo`lib, attribut belgisini ko`rsatilgan belgilar to`plamida o`z ichiga olishini tekshiradi;
- length: [CStringValidator](#) classning laqabi bo`lib, attribut belgisining ko`rsatilgan intervaldagi uzunligiga tekshiradi;
- match: [CRegularExpressionValidator](#) classning laqabi bo`lib, attribut belgisini to`g`ri ifodaga muvofiqlikka tekshiradi;
- numerical: [CNumberValidator](#) classning laqabi bo`lib, tekshirilayotgan attribut belgisini sonlikka tekshirish;
- required: [CRequiredValidator](#) classning laqabi bo`lib, attributlar belgisini bo`shlik yoki bo`shmaslikka tekshirish;
- type: [CTypeValidator](#) classning laqabi bo`lib, ko`rsatilgan ma`lumotning ko`rsatilgan tipga to`g`ri kelishini tekshiradi.;
- unique: [CUniqueValidator](#) classning laqabi bo`lib, ko`rsatilgan jadvaldagi ustunda kelayotgan attributning yagona bo`lishini tekshirish;
- url: [CUrlValidator](#) classning laqabi bo`lib, URL to`g`riligini ta`minlaydi.

Quyida biz validatorlarga oid bir qancha misollar ko`rsatamiz:

```
// name maydon — majburiy maydon

array('username', 'required'),

// foydalanuvchi nomining belgilari soni 3 va 12 orasida bo`lishi kerak.

array('username', 'length', 'min'=>3, 'max'=>12),
```

```
// «password» va «password2» maydonlari o`zaro teng bo`lishi kerak.
```

```
array('password', 'compare', 'compareAttribute'=>'password2', 'on'=>'register'),
```

// "password" maydonni ko`rsatilgan foydalanuvchi nomiga muvofiq kelishiga tekshirish, "authenticate" senariyasi bo`yicha.

```
array('password', 'authenticate', 'on'=>'login'),
```

### Attribut belgilarini xavfsiz o`zlashtirish.

Yangi model nusxasi hosil qilgandan keyin, bizdan tezda to`ldirish talab qilinadi. Bu juda oson qilinadi, xuddi massivdaka o`zlashtirish orqali:

```
$model=new LoginForm;
if(isset($_POST['LoginForm']))
 $model->attributes=$_POST['LoginForm'];
```

Misoldagi oxirgi ifoda xuddi massivli o`zlashtirishdir.

Agarda xar bir model attributiga mos \$\_POST['LoginForm'] dan ma`lumot birma bir o`zlashtirmoqchi bo`lsangiz, unda quyidagicha ish qilasiz.

```
foreach($_POST['LoginForm'] as $name=>$value)
{
 if($name xavfsiz attribut hisoblanadi)
 $model->$name=$value;
}
```

Bunday holatda kelayotgan malumotlarni to`g`ridan to`g`ri jadvalga yozishda xatolik bo`lishi mumkin. Agarda biz jadvalimizni bir ustuni tashqi jadvaldagagi ustun bilan bog`lansa unda bu jadvaldagagi ustunni to`ldirish kerak bo`ladi, ya`ni null bo`lib qolmasligi kerak.

Xafvsiz attributlar tarifi.

Agar validatsiya qoidalaridan foydalanib quyidagi senariyni qo`llasak, attribut xavfsiz sanaladi.

```
array('username, password', 'required', 'on'=>'login, register'),
```

```
array('email', 'required', 'on'=>'register'),
```

Yuqoridagi kodda username va password login senariyasi uchun ishlatalayotgan attributlar.

username, password va email register senariyasi uchun ishlatalayotgan attributlar.

Login senariyasi uchun faqat username va password attributlari tekshiruvdan (validatsiyadan) o`tadi. Ya`ni bu kelayotgan attributlar (username va password) majburiy to`ldirilishi kerak bo`lgan maydonlar. Register senariyasi uchun email attributi ham majburiy maydon hisoblanadi.

```
// interface ga kirish senariyasi bo`yicha (login)
```

```
$model=new User('login');
```

```
if(isset($_POST['User']))
```

```
 $model->attributes=$_POST['User'];
```

```
// ro`yxatdan o`tish senariyasi bo`yicha (register)
```

```
$model=new User('register');
```

```
if(isset($_POST['User']))
```

```
 $model->attributes=$_POST['User'];
```

Nima uchun attributlarni xavfsiz yoki yo`qligini aniqlash uchun aynan shu qoidadan foydalanamiz? Agar biz attributga bir qancha validatsiya qoidalarni biriktirsak, nima uchun biz hotirjam bo`lmasligimiz kerak?

To`g`ri user tomonidan kiritilayotgan ma`lumotlarni tekshirish uchun validatsiya qoidalarni tushunish biroz qiyin bo`lishi mumkin. Agar siz jadval yaratib, keyin model hosil qilsangiz model ruleslariga jadvalingizga mos validatsiya qoidalari avtomat biriktiriladi.

Bazan biz attributlarni xavfsiz e`lon qilishni hohlaymiz, buni qilish uchun xaqiqatda shu attributlar uchun xech qanday majburiy qoidalarni biriktirilmagan bo`lishi kerak. Misol uchun text maydoni, foydalanuvchi tomonidan ixtiyoriy belgini olaveradi. Bu uchun quyidagi maxsus safe qoidasidan foydalanamiz.

```
array('content', 'safe')
```

Aniq ko`rsatilgan xavfsiz attributlar uchun unsafe qoidasi ham mavjud.

```
array('permission', 'unsafe')
```

Xavfsiz attributlarni avvaldan aniqlashda biz unsafe dan kamdan kam foydalanamiz. Chunki keyinchalik bizga bu halaqit qilishi ham mimkin.

### Tekshirishni amalga oshirish.

Foydalanuvchi tomonidan kiritilayotgan ma`lumotlar bilan modelni to`ldirishimiz kerak, tekshiruvni amalga oshirayotganimizda [CModel::validate\(\)](#) metodiga murojaat qilishimiz mumkin. Biz bu ishni amalga oshirganimizdan keyin metod false yoki true natija qaytaradi.  [CActiveRecord](#) modeli uchun tekshiruvni amalga oshirayotganingizda to`g`ridan to`g`ri  [CActiveRecord::save\(\)](#) metodiga ham murojaat qilishingiz mumkin.

Tekshiruv uchun qaysidir to`plam qoidasi bajarilishini [scenario](#) xossasi orqali berishimiz mumkin.

Tekshiruv amalga oshayotganda qaysi xossa biriktirilgan bo`lsa o`shanga qaram bo`ladi. Ya`ni o`sha qoidalari to`plami bo`yicha bajariladi.

Qaysi validatsiya qoidalari to`plami amalga oshirish uchun biz modelning [scenario](#) xossasidan foydalanamiz.

Misol uchun: foydalanuvchining faqatgina username va password maydonlari uchun validatsiya ishlatmoqchi bo`lsak unda login senariyasidan foydalanamiz. Agar biz qo`shimcha maydonlarni ham validatsiyadan o`tkazmoqchi bo`lsak (misol uchun: mail, address kabilar ) unda register senariyasidan foydalanamiz. Agar biz register senariyasiga shu maydonlar uchun qoidalari biriktirgan bo`lsak.

Register senariysi uchun qanday ko`rsatilishini quyida ko`rsatib o`tganamiz.

```
// Bu yerda User degan model yaratmoqdamiz va register senariyasini o`rnatmoqdamiz.
```

```
//$model=new User;

//$model->scenario='register';

$model=new User('register');

// modelga ma`lumotlarni to`ldirish

$model->attributes=$_POST['User'];

// tekshiruvni amalga oshirish

if($model->validate()) // agar xaqiqat bo`lsa
```

```
...
else
...
public function rules()
{
 return array(
 array('username, password', 'required'),
 array('password_repeat', 'required', 'on'=>'register'),
 array('password', 'compare', 'on'=>'register'),
);
}
}
```

Birinchidagi qoida ixtiyoriy joyda username va password uchun ishlayveradi. Oxirgi ikkita qoida faqat register senariyasi ishlatganimizda ishlaydi.

### Xatolar haqida ma`lumot berish.

Tekshiruvdan (validate) keyin model obyektida xamma xatolarni topish mumkin. Biz ularni [CModel::getErrors\(\)](#) va [CModel::getError\(\)](#) orqali olishimiz mumkin. Birinchi metod modeldagи ko`rsatilagan attribut uchun xamma xatoliklarni ko`rsatadi, ikkinchisi esa faqat birinchi xatoni ko`rsatadi.

Tekshirish amalga oshayotgan vaqtida qandaydir xato bor yo`qligini [CModel::hasErrors\(\)](#) metodi orqali bilishiz mumkin. Bunda bu metod true yoki false qaytaradi. Agarda xatolik bor bo`lsa, xatolikni olish imkoniyatini [CModel::hasErrors\(\)](#) metodidan foydalanish mumkin.

### Attributlar belgisi.

Formalar bilan ishlaganingizda xar bir maydon uchun uning nomi aks etishini talab qiladi. Shunda bizga u qaysi maydonlarni to`ldirayotganimizni bilishimiz uchun kerak bo`ladi. Ya`ni xar bir maydon uchun shunga mos maydon nomini ko`rsatadi. Buni biz ham berishimiz mumkin lekin avtomatik modelda o`zi nomlab qo`yadi. Nomlarni o`zgartirish uchun [attributeLabels\(\)](#) metodidan foydalanamiz. Bu metod modelda joylashgan boladi.

**Nazorat savollari**

1. YII2 frameworki nima vazifani bajaradi;
2. YII2 frameworkida model nima vazifani bajaradi?
3. Model qanday hosil qilinadi?

## **29-30-ma'ruza. YII2 freymvorkda ma'lumotlar bazasi bilan ishlash**

**Reja:**

1. YII2 freymvorkda ma'lumotlar bazasi bilan ishlash.
2. YII2 da ma'lumotlar bazasi so'rovlari hoslil qilish.
3. ActiveRecord.

Bu bo'lim sizga yangi sahifani yaratishga yordam beradi, u sahifa esa countries jadvalidan kelgan ma'lumotlarni chiqarib beradi. Buning uchun esa siz ma'lumotlar ombori bilan aloqani sozlashingiz kerak, klass yaratish [Active Record, amalni](#) va [ko'rinish](#) yaratishimiz kerak.

Bu bo'limda siz nimalarni o'rganasiz:

Qanday qilib ma'lumotlar omboriga ulanish.

Active Record klassini yasash

Ma'lumotlarni olishni, Active Recordni ishlatishni.

viewda ma'lumotlarni sahifalarga bo'lib chiqarishni.

Etibor bering, shu bo'limni o`zlashtirish uchun, siz ma'lumotlar ombori bilan ishlash bilimingiz bo'lishi kerak. Aniqroq esa, siz ma'lumotlar omborini yasashni va unga SQL-so'rov jo'natishni, ma'lumotlar ombori klientlarini ishlatishni bilishingiz kerak.

### **Ma'lumotlar omborini tayyorlash**

- Birinchi bo'lib kigroup nomli ma'lumotlar ombori yaratamiz, chunki undan siz web ilovadan ma'lumotlarni qabul qilasiz. Ma'lumotlar omborini siz SQLite, MySQL, PostgreSQL, MySQL yoki Oracle da yaratishingiz mumkin, chunki Yiida aytib o'tilgan barcha ma'lumotlar ombori bilan ishlay oladi. Soddalik uchun keyinchalik, ma'lumotlar ombori misol uchun MySQL haqida gap boradi.

Endi esa ma'lumotlar omborida country degan jadval yaratish va unga ozgina misol uchun ma'lumotlar qo'shing. Siz quydagisi SQL-buyruqni bajarishingiz mumkin yuqorida aytilgan amalarni bajarish uchun:

```
CREATE TABLE `country` (`code` CHAR(2) NOT NULL PRIMARY KEY, `name` CHAR(52) NOT NULL, `population` INT(11) NOT NULL DEFAULT '0') ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
INSERT INTO `country` VALUES ('AU','Australia',24016400);
```

```
INSERT INTO `country` VALUES ('BR','Brazil',205722000);
```

```
INSERT INTO `country` VALUES ('CA','Canada',35985751);
```

```
INSERT INTO `country` VALUES ('CN','China',1375210000);
```

```
INSERT INTO `country` VALUES ('DE','Germany',81459000); INSERT INTO `country` VALUES ('FR','France',64513242);
```

```
INSERT INTO `country` VALUES ('GB','United Kingdom',65097000);
```

```
INSERT INTO `country` VALUES ('IN','India',1285400000); INSERT INTO `country` VALUES ('RU','Russia',146519759); INSERT INTO `country` VALUES ('US','United States',322976000);
```

Ma'lumotlar omborida bilan aloqani sozlaymiz

Ishni boshlashdan oldin sizda [PDO](#) kengaytmasi o'rnatilganiga amin bo'ling va PDO-boshqaruvchisi ma'lumotlar ombori bilan boshqara olishingiz uchun(misol uchun, pdo\_mysql MySQL uchun). Bu siz aloqali ma'lumtolar ombori bilan ishlappingiz uchun assosiy talablardir. Endi esa, hammasini o'rnatib bo'lganiningizdan so'ng, config/db.php oching va sozlamalarni o`zingizni ma'lumotlar omboringiz sozlamalariga o`zgartiring. Boshlanish sozlamalar quydagichadir:

```
<?php return ['class' => 'yii\db\Connection', 'dsn' => 'mysql:host=localhost; dbname=yii2basic', 'username' => 'root', 'password' => '', 'charset' => 'utf8',];
```

Fayl config/db.php — oddiy [sozlamalar](#) vositasidir, faylarda assoslangan. Quydagi sozlama fayli [yii\db\Connection](#) klass nusxasini olish uchun kerak bo'lgan assosiy talablarga javob beradigan sozlamalardan iboratdir, undan esa siz SQL-buyruqlarni ma'lumotlar omboriga jonatishingiz mumkin.

Ma'lumotlar omboriga yuqorida sozlamalar sozlab bo'lgandan so'ng ulanish, undan foydalanish uchun Yii::\$app->db orqali amalga oshiriladi.

Active Record klassi (sinf) avlodini yaratamiz [¶](#)

countrydan ma'lumotlarni olib ishlatib ko'rish uchun, [Active Record](#)ni kengaytirgan klass yarating, nomi Country va uni models/Country.php fayliga saqlang.

```
<?php namespace app\models;
use yii\db\ActiveRecord;

class Country extends ActiveRecord { }
```

Country klassi [yii\db\ActiveRecord](#)ni kengaytirmoqda. Siz uning ichida bir satr kod ham yozmasligingiz mumkin! Yuqorida kod assosida Yii o'zi ma'lumotlar omborigagi jadval va klass orasida aloqa yasaydi.

Country klassini ishlatganingizda, siz osonlikcha country jadvalidagi ma'lumotlar bilan boshqarishingiz mumkin, misol uchun pastdag'i qatorlarda shuni ko'rishingiz mumkin:

```
use app\models\Country;

// "country" jadvalidan hamma ma'lumotlarni qabul qilamiz va ularni "name" ustuni bo'yicha saralaymiz.

$countries=Country::find()->orderBy('name')->all();

// "US" jadval ka'liti bo'yicha ma'lumotni olib beradi. $country = Country::findOne('US');
//"United States"

echo $country->name;

// nomi o`zgartiramiz "U.S.A."ga va ma'lumotlar omborida saqlaymiz.

$country->name = 'U.S.A.';

$country->save();
```

## Amal yaratish

Ma'lumotlarni foydalanuvchiga ko'rsatish uchun siz amal yaratishingiz kerak. Oldingi misolarda farqli o'laroq site nazoratchisi ichida amal yasash o'rniha to'g'riroq buning uchun maxsus nazoratchi

yasab unga hamma usularni yig'ish to'g'riroq bo'ladi, Yangi nazoratchini CountryController deb nomlang, va unda index degan amal yarating uning ichida pastda ko'rsatilgandek ma'lumot kirititing:

```
<?php namespace app\controllers;
use yii\web\Controller;
use yii\data\Pagination;
use app\models\Country;
class CountryController extends Controller {
public function actionIndex() {
 $query = Country::find();

 $pagination = new Pagination(['defaultPageSize' => 5, 'totalCount' => $query->count(),]);

 $countries = $query->orderBy('name') ->offset($pagination->offset) ->limit($pagination->limit) ->all();

 return $this->render('index', ['countries' => $countries, 'pagination' => $pagination,]); } }
```

Yuqoridagi kodni quydagicha saqlang controllers/CountryController.php.

index amali Country::find()ni chaqirmoqda. Ushbu Active Record usuli country jadvalidan hamma ma'lumotlarni qaytaradi. kelayotgan ma'lumotlar bo'lish sahfalarga bo'lish uchun [yii\data\Pagination](#) klassi ishlatalmoqda. Pagination nusxasi ikkta maqsadga kerak:

offset va limit SQL-buyrug`i uchun belgilanadi, bu esa birmartada faqat bitta sahifani qaytarishga berilgan (bitta sahifada misl uchun 5 ta satr)

u esa viewda paginatsiya (pagination) chiqarilishi uchun ishlatalmoqda, turi esa sonlardan iborat, bularni hammasini oldinda tushuntirilib kelinadi.

Kodni oxirida esa index amali index ko'rinishini chiqarib beradi, unga hamma chiqarilishi kerak bo'lgan ma'lumotlarni va paginatsiyanni qaytarayapti.

### Ko'rinish yaratish

Birinchi bo'lib country degan papka ochamiz views ichida. Bu papka hamma nazoratchi ko'rinishlarni saqlash uchun ishlamatamiz. Ushbu papkada views/country fayl yaratamiz index.php, so'ng quydagi kodni yozamiz:

```
<?php use yii\helpers\Html;
use yii\widgets\LinkPager; ?> <h1>Countries</h1>
<?php foreach ($countries as $country): ?>
 <?= Html::encode("{$country->code} ({$country->name})") ?>: <?= $country->population
?>
<?php endforeach; ?>
<?= LinkPager::widget(['pagination' => $pagination]) ?>
<?php use yii\helpers\Html;
```

```

use yii\widgets\ActiveForm;

$this->title = 'Login'; ?>

<h1><?= Html::encode($this->title) ?></h1> <p>Please fill out the following fields to login:</p>
<?php $form = ActiveForm::begin(); ?>

<?= $form->field($model, 'username') ?>
<?= $form->field($model, 'password')->passwordInput() ?>
<?= Html::submitButton('Login') ?>
<?php ActiveForm::end(); ?>

```

Ko'rinish ikkta qisimga bo'lingan ma'lumot chiqishiga nisbatan. Ko'rinishni birinchi qismida ma'lumotlar tartibsiz HTML-ro'yxati kabi chiqayapti. Ikkinchisida esa [yii\widgets\LinkPager](#) vidjiti, u ko'rinishga qaytarilayotgan paginatsiya ma'lumoti assosida shaklanadi. Vidjet LinkPager sahifalar tugalarini chiqarib beradi. Ulardan birortasiga bosangiz sahifa yangilanib ma'lumotlar o'zgaradi

- Keling sinab ko'raylik
- Yuqoridagi yozgan kodni ishlashini ko'rish uchun iltimos ushbu yo'l (url) bo'yicha o'ting:

## Countries

- AU (Australia): 24016400
- BR (Brazil): 205722000
- CA (Canada): 35985751
- CN (China): 1375210000
- FR (France): 64513242



Boshida siz ma'lumotlar omboridan beshta ma'lumotni chiqarib bergenini ko'rishingiz mumkin. Ma'lumotlar ostida. Agar siz "2" tugmasiga bosangiz, keyingi ikkinchi sahifa ma'lumotlarini ko'rsatayotganini ko'rishingiz mumkin. Etibor bilan qarasangiz, URLda ham quydagi "2" sahifa chiqtishi mumkin.

<http://hostname/index.php?r=country%2Findex&page=2>

[Pagination](#) Ma'lumotlarni sahifalarga bo'lib qisman chiqarish umumiyligi ishlash tartibi esa quydagicha:

Boshida [Pagination](#) birinchi sahifani ko'rsatadi, uning SQL-buyrug'i qismi LIMIT 5 OFFSET 0 bo'ladi. Natijada, birinchi beshta ma'lumot qaytarilib ko'rsatiladi.

Vidjet [LinkPager](#) URLdagi ma'lumotda assosan tugmachalarni shaklanirib beradi, URLni esa [Pagination](#). Bu URLda page parametri mavjud bo'ladi, tugmachalarda esa xar xil page parametri mavjud.

Agar "2" tugmasiga bosangiz, sahifa yangilanib country/index uchun yangi so'rov amalga oshadi. Shundey qilib SQL-so'rovi LIMIT 5 OFFSET 5 tashkil qiladi va keyingi beshtalikni qaytaradi.

Ushbu bo'limda [Gii](#)- dan ba'zi bir keng tarqalgan veb-sayt xususiyatlarini amalga oshiradigan kodni avtomatik ravishda yaratish uchun qanday foydalanish tasvirlangan . Kodni avtomatik ravishda yaratish uchun Gii-dan foydalanish shunchaki Gii veb-sahifalarida ko'rsatilgan ko'rsatmalarga to'g'ri ma'lumotlarni kiritish bilan bog'liq.

Ushbu o'quv qo'llanma orqali siz quyidagilarni bilib olasiz:

arizangizda Gii-ni yoqish

Active Record sinfini yaratish uchun Gii dan foydalanish,

MB jadvali uchun CRUD operatsiyalarini amalga oshiradigan kodni yaratish uchun Gii dan foydalanish,

Gii tomonidan yaratilgan kodni sozlash.

[Gii](#) Yii da [modul](#) sifatida taqdim etilgan . Siz Gii- ni dasturning [modullari](#) xususiyatida sozlash orqali yoqishingiz mumkin . Arizangizni qanday yaratganingizga qarab, config/web.php konfiguratsiya faylida quyidagi kod mayjudligini topishingiz mumkin :

```
$config = [...]; if (YII_ENV_DEV) { $config['bootstrap'][] = 'yii'; $config['modules']['yii'] = ['class' => 'yii\gii\Module',];
```

Ushbu qator tufayli sizning ilovaingiz ishlab chiqish rejimida va yuqoridagi konfiguratsiya bo'yicha Gii-ni allaqachon yoqib qo'yan bo'ladi. Endi Gii-ga quyidagi URL orqali kirishingiz mumkin:

- <http://hostname/index.php?r=gii>

The screenshot shows the Yii Gii code generator interface. At the top, there's a navigation bar with the Yii logo, 'Home', 'Help', and 'Application'. Below the header, the main title is 'Welcome to Gii a magical tool that can write code for you'. There are six main sections, each with a 'Start »' button:

- Model Generator**: Generates an ActiveRecord class for a specified database table.
- CRUD Generator**: Generates a controller and views for CRUD operations.
- Controller Generator**: Helps generate a new controller class and actions.
- Form Generator**: Generates a view script file for a form.
- Module Generator**: Helps generate skeleton code for a Yii module.
- Extension Generator**: Helps generate files needed by a Yii extension.

At the bottom left, there's a green button labeled 'Get More Generators'.

## Faol yozuvlar sinfini yaratish

Gii-dan Active Record sinfini yaratish uchun foydalanish uchun "Model Generator" -ni tanlang (Gii indeks sahifasidagi havolani bosish orqali). Keyin shaklni quyidagicha to'ldiring:

Jadval nomi: country

Model sinfi: Country

The screenshot shows the Yii Model Generator interface. On the left, there's a sidebar with links: Model Generator (selected), CRUD Generator, Controller Generator, Form Generator, Module Generator, and Extension Generator. The main area has the following fields:

- Table Name:** country
- Model Class:** Country
- Namespace:** app\models (highlighted)
- Base Class:** yii\db\ActiveRecord (highlighted)
- Database Connection ID:** db (highlighted)
- Use Table Prefix:** (unchecked)
- Generate Relations:** All relations (highlighted)
- Generate Labels from DB Comments:** (unchecked)
- Generate ActiveQuery:** (unchecked)
- Enable I18N:** (unchecked)
- Use Schema Name:** (checked)
- Code Template:** default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/model/default) (highlighted)

At the bottom, there are two buttons: **Preview** and **Generate**. Below them is a note: "Click on the above **Generate** button to generate the files selected below:". A table shows the selected file: **Code File: models\Country.php** with **Action: create** and checkboxes for **Create**, **Unchanged**, and **Overwrite**.

- Keyin, "Ko'rib chiqish" tugmasini bosing. models/Country.php Natijada hosil bo'ladijan sinf faylida ko'rsatilganligini ko'rasiz . Uning tarkibini oldindan ko'rish uchun sinf fayli nomini bosishingiz mumkin.
- Gii-dan foydalanganda, agar siz o'sha faylni allaqachon yaratgan bo'lsangiz va uning ustiga yozishni diffistasangiz, yaratiladigan kod va mavjud versiya o'rtaqidagi farqlarni ko'rish uchun fayl nomi yonidagi tugmani bosing.

The screenshot shows the Yii code generator's Model Generator interface. The main area displays the generated code for a `Country` model:

```

9 9 *
10 10 * @property integer $id
11 11 * @property string $name
12 12 * @property string $code
13 13 */
13 14 class Country extends \yii\db\ActiveRecord
14 15 {
15 16 ...
16 17 public function rules()
17 18 {
18 19 return [
19 20 [['id'], 'integer'],
20 21 [['name'], 'string', 'max' => 50],
21 22 [['code'], 'string', 'max' => 15],
22 23];
23 24 }
24 25 ...
25 26 public function attributeLabels()
26 27 {
27 28 return [
28 29 'id' => 'ID',
29 30 'name' => 'Name',
30 31 'code' => 'Code',
31 32];
32 33 }
33 34 }
34 35
35 36
36 37
37 38
38 39
39 40
40 41
41 42
42 43
43 44
44 45
45 46
46 47
47 48

```

Below the code editor are several configuration options:

- Generate Relations:** Options include "All relations" (selected), "Generate Labels from DB Comments", "Generate ActiveQuery", and "Enable I18N".
- Code Template:** Set to "default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/model/default)".
- Buttons:** "Preview" and "Generate".
- Action Bar:** Buttons for "Create" (checked), "Unchanged" (checked), and "Overwrite" (checked).
- Table:** Shows the selected code file "models\Country.php" with an "overwrite" action button.

A Product of Yii Software LLC

Powered by Yii Framework

- Mayjud faylni qayta yozishda "qayta yozish" yonidagi katakchani belgilang va keyin "Yaratish" tugmasini bosing. Agar yangi fayl yaratilsa, shunchaki "Yaratish" tugmacha sinasi bosishingiz mumkin.
- Keyinchalik, kod muvaffaqiyatli ishlab chiqarilganligini tasdiqlovchi sahifani ko'rasiz. Agar sizda mayjud bo'lgan fayl bo'lsa, unda siz yangi yaratilgan kod bilan yozilganligini ko'rsatadigan xabarni ko'rasiz.
- CRUD kodini yaratish

CRUD yaratish, o'qish, yangilash va o'chirish degan ma'noni anglatadi, aksariyat veb-saytlardagi ma'lumotlar bilan olingan to'rtta umumiyl vazifani aks ettiradi. Gii yordamida CRUD funktsiyasini yaratish uchun "CRUD Generator" ni tanlang (Gii indeks sahfasidagi havolani bosish orqali). "Mamlakat" misoli uchun quyidagi shaklni to'ldiring:

Model sinfi: app\models\Country

Model sinfini qidirish: app\models\CountrySearch

Nazoratchi sinfi: app\controllers\CountryController

yii code generator

Home Help Application

Model Generator >

**CRUD Generator** >

Controller Generator >

Form Generator >

Module Generator >

Extension Generator >

## CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

**Model Class**

app\models\Country

**Search Model Class**

app\models\CountrySearch

**Controller Class**

app\controllers\CountryController

**View Path**

@app\views\country

**Base Controller Class**

yii\web\Controller

**Widget Used in Index Page**

GridView

**Enable I18N**

**Enable Pjax**

**Code Template**

default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/crud/default)

**Preview**

Keyin, "Ko'rib chiqish" tugmasini bosing. Quyida ko'rsatilgandek yaratiladigan fayllar ro'yxatini ko'rasisiz.

yii code generator

Home Help Application

**View Path**

@app\views\country

**Base Controller Class**

yii\web\Controller

**Widget Used in Index Page**

GridView

**Enable I18N**

**Enable Pjax**

**Code Template**

default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/crud/default)

**Preview** **Generate**

Click on the above **Generate** button to generate the files selected below:

Code File	Action	
controllers\CountryController.php	create	<input checked="" type="checkbox"/>
models\CountrySearch.php	create	<input checked="" type="checkbox"/>
views\country\_form.php	create	<input checked="" type="checkbox"/>
views\country\_search.php	create	<input checked="" type="checkbox"/>
views\country\create.php	create	<input checked="" type="checkbox"/>
views\country\index.php	create	<input checked="" type="checkbox"/>
views\country\update.php	create	<input checked="" type="checkbox"/>
views\country\view.php	create	<input checked="" type="checkbox"/>

**Create**  **Unchanged**  **Overwrite**

- Agar siz controllers/CountryController.phpva views/country/index.phpfayllarni ilgari yaratgan bo'lsangiz (qo'llanmaning ma'lumotlar bazalari bo'limida), ularni almashtirish uchun "ustiga yozish" katagiga belgi qo'ying. (Oldingi versiyalarda CRUD to'liq qo'llab-quvvatlanmagan.)
- Qanday ishlashini ko'rish uchun quyidagi URL-ga kirish uchun brauzeringizdan foydalaning:
- <http://hostname/index.php?r=country%2Findex>
- Ma'lumotlar bazasi jadvalidan mamlakatlarni ko'rsatadigan ma'lumotlar tarmog'ini ko'rasisz. Siz katakchani saralashingiz yoki ustun sarlavhalariga filtr shartlarini kiritish orqali filtrlashingiz mumkin.
- Tarmoqda ko'rsatilgan har bir mamlakat uchun uning tafsilotlarini ko'rishni, yangilashni yoki o'chirishni tanlashingiz mumkin. Shuningdek, yangi mamlakat yaratish uchun ariza bilan ta'minlash uchun tarmoqning yuqori qismidagi "Mamlakat yaratish" tugmachasini bosishingiz mumkin.

My Company

Home About Contact Login

Home / Countries

## Countries

[Create Country](#)

Showing 1-10 of 10 items.

#	Code	Name	Population	
1	AU	Australia	18886000	
2	BR	Brazil	170115000	
3	CA	Canada	1147000	
4	CN	China	1277558000	
5	DE	Germany	82164700	
6	FR	France	59225700	
7	GB	United Kingdom	59623400	
8	IN	India	1013662000	
9	RU	Russia	146934000	
10	US	United States	278357000	

« 1 »

© My Company 2014

Powered by

## Update Country: United States

**Code**

US

**Name**

United States

**Population**

278357000

**Update**

- Quyida ushbu xususiyatlarning qanday amalga oshirilishini tekshirishni yoki ularni sozlashni istasangiz, Gii tomonidan yaratilgan fayllar ro'yxati keltirilgan:
- Nazoratchi: controllers/CountryController.php
- Modellar: models/Country.phpvamodels/CountrySearch.php
- Ko'rishlar: views/country/\*.php

## **II. LABORATORIYA MASHG'ULOT MATERIALLARI**

Web dasturlashga kirish fanidan laboratoriya ishlarini bajarish bo'yicha uslubiy ko'rsatma «WEB dasturlashga kirish» kursi WWW ning rivojlanish bosqichlari, kliyentserver texnologiyalari, internet protokollar, web ilovaning tarkibiy qismlari, frontend va backend texnologiyalarini loyihalash va yaratish, web-saytlarni ishlab chiqishda ma'lumotlar bazasi bilan aloqani o'rnatish, murakkab so'rovlarni amalga oshirish ko'nikmalarini hosil qilishga qaratilgan. Mazkur kursning maqsadi zamonaviy web dasturlashning nazariy va amaliy bilimlarini o'rganish va ularni yaratish texnologiyalarini, web dasturlash uchun mo'ljallangan HTML5, CSS3-4, JavaScript, JQuery, Bootstrap, AngularJs va PHP dasturlari, zamonaviy freymvork platformalari orqali web texnologiyalarning asosiy tushunchalarini, server tomondan dasturlash texnologiyalarini, MySQL, AJAX texnologiyasi va ular bilan ishlash, MVC freymvork texnologiyalari bilan ishlash, YII2 freymvorkda veb-saytlarni loyihalashning ko'nikma va malakalarini shakllantirishdan iborat. Ushbu dasturga muvofiq kurs ma'ruzalar mavzulari, talabalarning individual mashg'ulotlari mavzulari, talabalarning mustaqil ishlari mavzularidan iborat. Talabalar individual mashg'ulotlarini bajarish orqali yuqorida bilimlarga ega bo'lishga zamin yaratiladi. Talabalarning mustaqil ishlashlar uchun quyida laboratoriya topshiriqlar keltiriladi. Har bir talaba bitta mavzuni tartib nomeri bo'yicha oladi. Bu mavzuda berilgan qandaydir web saytni yaratish jarayonida 10 ta mavzulashtirilgan laboratoriya mashg'ulotini bajaradilar. Laboratoriya ishlarini baholash umumiyligi bahoning 30% tashkil qiladi. Har bir topshiriq uchun maksimal umumiyligi ballning 3% i beriladi.

Mavzulashtirilgan laboratoriya ishlarini topshirishda talaba hisobot shaklida topshiriq natijalarining rasmlari va dastur kodilarini topshiradi.

Web dasturlashga kirish fanidan laboratoriya ishlari topshiriqlari № Laboratoriya ishi mavzusi

1. Boshlang'ich ta'lim online mакtab web sayti
2. Elektron kutubxona (online kitoblar)
3. Ilmiy anjumanlar faoliyatini yoritib borishga mo'ljallangan web ilova yaratish
4. Taomlar, ovqatlar, pishiriqlar retsepti web sayti
5. Sog'liqni saqlash uchun (kasalliklar ma'lumotnomasi +online doktor konsultatsiyasi) web sayti
6. "Dasturchilar forumi" nomli web ilova yaratish
7. online virtual turizm web sayti
8. Ingliz tilini o'rganuvchilar uchun "Conversation dictionary" (so'zlashuvda ishlatiladigan so'z va iboralar) nomli veb ilovani yaratish.
9. "Ofis qurilmalariga xizmat ko'rsatish" mavzusini yoritishga mo'ljallangan web ilova yaratish ( kompyuter, printer, skayner, mobil qurilmalar ... , tuzatish yo'llari va maqsadli foydalanish bo'yicha ko'rsatma va izohlar)
10. Ishga joylashish haqida konsultatsiya web sayti (employing.uz)
11. Bolalar ijodiyotini yoritib borishga yo'naltirilgan web ilova yaratish
12. Business ideas - Yangi biznes g'oyalar web sayti (biznesni qanday va nimadan boshlash bo'yicha konsultatsiya)

- 13 History of the Uzbekistan. O'zbekiston tarixi haqida ma'lumotnomma web sayti
14. Online Oliy Matematika web sayti (Ma'lumotlar, topshiriqlar, yechimlar)
15. AutoOlam.uz AvtomMashinalar, AvtoKompaniyalar, AvtoSalonlar. Spravochnik. Foydali materiallar web sayti
16. Xususiy maktablarda masofaviy ta'limni amalga oshiruvchi axborot tizimni yaratish
17. Futbol chempionatlari web sayti
18. Bola tarbiyasi (chaqaloqlikdan to o'smirlik paytigacha) uchun yordam beruvchi maxsus web sayt (ota-onalar uchun)
19. Online saylov tizimi web sayti
20. Arxitektura sohasi uchun web sayti (Yangiliklar, foydali manbalar, Arxitektura tashkilot firmalari haqida ma'lumotlar)
21. Manufacture.uz. Ishlab chiqarish korxonalari faoliyati, yangiliklari, ko'rsatkichlari web sayti (O'zbekistondagi yirik korxonalar)
22. O'zbekistondagi barcha Oliyohlar (Universitet, institut...) ni umumiyligi ma'lumotlari, yangiliklari, yagona birlashmasi web sayti
- 23 Reklama xizmati web sayti (video, audio, image ko'rinishdagi barcha turdag'i reklama xizmatlari)
24. Dori-darmon ma'lumotnomasi web sayti. (Barcha turdag'i dorilar yo'riqnomasi, xususiyatlari va hk)
25. Qishloq xo'jalik mahsulotlari online savdosi web sayti
26. Chorvachilik bo'yicha maxsus tizim (chorva mollari xususiyatlari, vet. spravochnik, kerakli ko'rsatma, ma'lumotlar)
27. Dehqonchilik hamda bog'dorchilik bo'yicha maxsus tizim (foydali manbalar, ma'lumotlar)
28. Dasturchilarning interaktiv veb saytini yaratish
29. Go'zallik sirlari web sayti (Sog'lom turmush tarzi, go'zallik sirlari bo'yicha foydali ma'lumotlar)
30. Yo'l harakati qoidalarini o'rgatishga yo'naltirilgan web ilova yaratish
31. Dasturchilarning interaktiv veb saytini yaratish.
32. Korporativ sayt yaratish
33. Elektron do'kon yaratish
34. Elektron jurnal yaratish
35. Elektron dorixona yaratish

## 1-laboratoriya ishi

### Mavzu: HTML hujjat tuzilishi. HTML ning asosiy teglari, teg attributlari

**Ishdan maqsad:** HTML teglaridan foydalanishni o'rganish va web sayti tuzish ko'nikmalarini hosil qilishdan iborat.

### Nazariy qism

HTML (Hyper Text Markup Language) – belgili til bo'lib, ya'ni bu tilda yozilgan kod o'z ichiga mahsus ramzлarni mujassamlashtiradi. Bunday ramzlar hujjat ko'rinishini faqatgina boshqarib, o'zi esa ko'rinxaydi. «<>» belgi ichidagi yozuvlar teg deb ataladi. Ko'p teglarning ochiq(<>) va yopiq(</>) hollari ishlataladi.

**<html>...</html>** — ochiluvchi va yopiluvchi «**html**» nomli teg. Bu teg HTML hujjat boshlanishini bildiradi va barcha kodlar aynan shu teg ichida yoziladi.

**<head>...</head>** sahifaning bosh qismi. Bu tegdagи kodlar foydalanuvchilarga ko'rinxaydi. Oddiy sahifalar yaratishda judayam muhim teg hisoblanmaydi. Bu teg ichida quyidagi teglar joylashishi mumkin:

- html meta teglari(meta)

- sahifa nomi(title)

- JavaScript kodlarini ularash(JavaScript)

- stillarni qo'shish(link)

**<title>...</title>** — sahifa nomini berish. Bu yerga yozilgan yozuvlar brauzerning yuqori qismida namoyon bo'ladi. **<body>...</body>** — saqifaning tana qismi. Asosiy qism hisoblanadi va bu qism

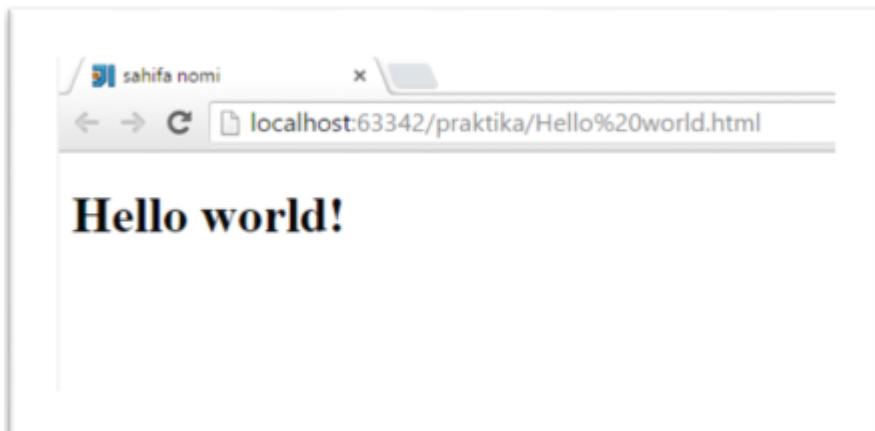
foydalanuvchiga bevosita namoyon bo'ladi. Barcha asosiy kodlar aynan shu teg ichida joylashgan bo'ladi.

«<!— ... —>» — bu belgilar kommentariyani bildiradi, brauzerda bularning ichidagi yozuv ko'rinxaydi. Kodlarni tushuntirishda kommentariyalardan foydalanish foydali bo'ladi.

“Hello world” so'zini browser ekraniga chiqarish

```
<!DOCTYPE html>
<html>
 <head>
 <title>Sahifa nomi</title>
 </head>
 <body>
 <h1>Hello World!</h1>
 </body>
</html>
```

Natija:



**<center>...</center>** — bu teg ichidagi barcha yozuvlar(teqlar ham bo‘lishi mumkin) sahifaning markazidan boshlab yoziladi, shunda umumiy yozuv o‘rtada namoyon bo‘ladi.

**<hr>** — bu tegning yopiluvchi sherigi yo‘q. Teg oddiy chiziq chizib beradi. Uning quyidagi attributlari mavjud:

**align** — chiziqni tekisligini o‘rnatadi, quyidagi qiymatlarni qabul qilishi mumkin:

\*left — chap tomondan tekislash

\*right — o‘ng tomondan tekislash

\*center — markazga tekislash.

**color** — chiziq rangini berish(rang nomi yoki 16lik tizimdagি raqami).

**noshade** — effektlarsiz chizish.

**size** — chiziq qalinligi(size=<20"').

**width** — chiziq uzunligi, piksel yoki fozda beriladi(width=<20%>).

**<br>** — kursorni bitta pastga tushirish, undan keyin keladigan barcha narsalar(jadval, yozuv,...) bitta pastdan yoziladi(qisqasi Enter). Yopiluvchi sherigi yo‘q.

**<h1>...</h1>** — Matnga nom qo‘yish(zagalovka). Bu klass 6 xil ko‘rinishda yozilishi mumkin(h1, h2, h3, h4, h5, h6). Raqamlar kattalashgan sari yozuvlar o‘lchami kichiklashib boradi. Bu teg har xil o‘lchamdagи nomlar(bo‘lim nomi, band nomi, bob nomi,...) berishga qulay. Uning bitta attributi bor:

**align** — yozuvni tekislash, qiymatlari quyidagilar:

\*left — chap tomonga to‘g‘irlash(odatiy holi).

\*center — markazga to‘g‘irlash.

\*right — o‘ng tomonga to‘g‘irlash.

\*justify — butun qator bo‘ylab to‘g‘irlash.

**<p>...</p>** — paragraflar qo‘shish uchun ishlataladi. Har bir ochilgan <p> teg yozuvni, yangi qatordan boshlab ekranga chiqaradi. Bu tegning quyidagi attributi bor:

**align** — yozuvni tekislash, u quyidagi qiymatlarni qabul qiladi:

\*left — chap tomonga tekislash;

\*right — o‘ng tomonga tekislash;

\*center — markazga tekislash;

\*justify — butun qator bo‘ylab tekislash;

**<font>...</font>** — yozuvni formatlangan holda chiqarib beradi. Quyidagi attributlari mavjud:

**color** — yozuv rangini o‘rnatish;

**face** — yozuv shriftini o‘rnatish(times new roman, verdana, arial);

**size** — yozuv o‘lchamini o‘rnatish(1 dan 7 gacha qiymat).

**Misol:**

<html>

```
<head> <title>Hello</title></head>
<body bgcolor=<orange" text=<black">
MarkaZ EmaS
<center>MarkaZ</center>
<hr color=<red" >
Mashhur Manchester United klubi

1878 yilda tashkil etilgan!
<h1 align="center">Giggs</h1>
<h2 align="right">Young</h2>
<h3 align="left">Valencia</h3>
<h4>Phil John</h4>
<h5>Wayne Rooney</h5>
<h6>Ferdinand</h6>
<p>1. AkmX.uz</p>
<p align="center"><font size=<4" color=<green" face=<verdana">Manchester United 2012/2013
yili
g'olib</p>
</body>
</html>
```

## Topshiriqlar

Har bir talaba “Web dasturlashga kirish fanidan laboratoriya ishlari” uslubiy ko’rsatmasida berilgan laboratoriya topshiriqlari bo'yicha HTML teglarining asosiy teglaridan foydalaniib, topshirqlarini shaklantiradi.

## 2-laboratoriya ishi

**Mavzu: HTML5 da yangi teglar. HTML5 da audio, video va grafika bilan ishlash teglari.**

**Ishdan maqsad:** HTML5 teglaridan foydalanishni o'rganish va web sayti tuzish ko'nikmalarini hosil qilishdan iborat.

### Nazariy qism

Sayt strukturasini yaratishda asosan biz <div> tegidan yoki jadvallardan foydalanamiz. Saytning qismini <div> tegi orqali ajratamiz. Html5 da esa sayt strukturasi uchun alohida teglar yaratilgan. Bu teglar quyidagilar:

<HEADER>- sahifa nomi uchun teg.

```
<header>
<h1>This is the Title of the Content</h1>
<div class=<post-meta">
<p>By Author <span class=<category">Filed in Web 2.0</p>
</div>
</header>

<FOOTER> - saytning eng pastki qismi uchun zarur bo'lган teg.
Asosan bu tegning ichiga "copyright" kabi ma'lumotlar yoziladi.
<footer>
<div class=<tags">
<span class=<tags-title">Tags: <a href="#" rel=<tag">Command
Prompt, <a href="#" rel=<tag">Compass, <a href="#" rel=<tag">CSS, <a href="#" rel=<tag">Sass, <a href="#" rel=<tag">Terminal
</div>
<div class=<facebook-like">
<div>10 likes</div> <!-- let's pretend it to be the facebook like -->
</div>
</footer>
```

<NAV> - sayt navigatsiyasi uchun teg.

```
<nav>

Overview
History
Development

</nav>
```

<ASIDE> - asosan o'ng yoki chap menyular yaratish uchun ishlatish mumkin bo'lган teg.

<SECTION> - sahifadagi asosiy ma'lumotlarni chiqarish uchun.

<ARTICLE> - yangiliklar, kommentariylar, maqolalarning kontentlarini chiqarish uchun.

<article>

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus id felis et augue sagittis
euismod quis at sem. Nunc sit amet magna ac velit congue ultricies. Sed eros justo, lacinia in
fringilla sollicitudin, congue id massa. Nunc dignissim bibendum nibh, sed dictum massa pharetra
```

sit amet.</p>
 </article>
 <VIDEO> - saytga video joylashtirish uchun.
 <video width=<>600</> height=<>400</> controls=<>controls</> poster=<>video/poster.jpg</>>
 <source src=<>video/video.mp4</> type='video/mp4; codecs=<>avc1.42E01E, mp4a.40.2</>'>
 Teg video ne podderjivaetsya vashim brauzerom.
 <a href="video/video.mp4">Skachayte video</a>.
 </video>
 <AUDIO> - saytga audio joylashtirish uchun.
 <audio src=<>video/1.mp3</> controls=<>controls</>></audio>
 <CANVAS> - bazadan qiymat olib diagramma shaklida chiqarish uchun.
 <canvas id=<>smile</> width=<>200</> height=<>200</>>
 <p>Vash brauzer ne podderjivaet risovanie.</p>
 </canvas>
 Bu teglarni kerakli joylarda to'g'ri ishlatish sayt strukturasini tushunarli (tartibli) bo'lishiga olib keladi.

```

<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>video</title>
 </head>
 <body>
 <video width="600" height="400" controls="controls" poster="video/poster.jpg">
 <source src="video/video.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
 Скачайте видео.
 </video>

 Audio

 <audio src="video/1.mp3" controls="controls"></audio>
 </body>
</html>

```

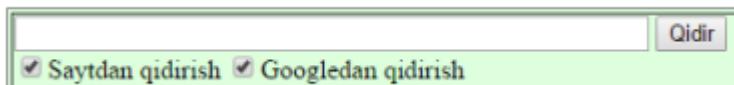


## HTML5 da formalar bilan ishlash

Web saytlarda formalar juda ko‘p joyda ishlatiladi. Misol uchun, **registratsiyada, so‘rovlarda, avtorizatsiyada, kommentariyalarda** va hokazo.

```
<html>
<head> <title>Form of Form</title></head>
<body>
<form name="forma" action="index.php" method="get">
<table border=1 bgcolor=<#ddffdd> align="center">
<tr>
<td align="left">
<input type="text" name="search" size=<50> value="Qidirish tizimi...">
<input type="submit" value="Qidir">

<input type="checkbox" name="site">Saytdan qidirish
<input type="checkbox" name="google">Googledan qidirish
</td>
</tr>
</table>
</form>
</body>
</html>
```



Bu kodlarda quyidagi teglar ishlatilgan:

1. **<form>...</form>** — veb sahifaga forma joylashtirish, forma foydalanuvchi va veb server bilan aloqani ta‘minlab beradi. Agar forma elementlari(**button**, **checkbox**, **radiobutton**,...) kiritilmasa, bu teg brauzerga hech nima chiqarmaydi, uning attributlari:

**accept-charset** — serverga jo‘natilayotgan axborot kodirovkasini o‘rnatish.

**<form accept-charset=<windows-1251>...</form>**

**action** — forma orqali jo‘natilayotgan ma’lumotlar qaysi sahifa orqali qayta ishlanishi kerakligi beriladi, biror adres ko‘rsatiladi.

**<form action=<index.php>...</form>**

formadagi barcha axborotlar **«index.php»** sahifasiga olib o‘tilib, u yerda tahrirlanadi.

**autocomplete** — formani avtomat to‘ldirishni boshqarish, brauzer keshida qandaydir ma’lumotlar bo‘lsa, shular orqali forma to‘latilishi mumkin, uning qiymati:**on**(yoqig‘), **off**(o‘chiq).

**method** — axborotlarni jo‘natish turi.

**name** — formaga nom berish, bu nom skriptlar yozishda ishlatiladi.

**novalidate** — formaga kiritilgan ma’lumotlarni brouzer tomonidan tekshirish. Brouzer pochta adreslarni, url adreslarni tekshirishi mumkin.

**target** — jo‘natilayotgan axborotni qayta ishlovchi veb sahifani ochilish turi, uning qiymatlari: **\_blank**, **\_self**, **\_parent**, **\_top**. **«a»** tegidagi **«target»** attributi bilan bir hil.

2. **<input>...</input>** — bu teg orqali forma elementlarini yaratish mumkin, ya’ni formani asosi hisoblanadi. Bu elementlarni yuqoridaagi kodda keltirilgan

Bu tegning attributlari:

**type** — forma elementi turi, bu attribut formada har xil ko‘rinishdagi elementlar hosil qilib beradi, ularni ba’zilarini misollarda ko‘rish mumkin, ularning qiymatlari quyidagilar:

\*button — tugma(knopka) hosil qilish.

\*checkbox — bir necha tanlovni amalga oshirib beruvchi bayroqcha hosil qilish.

\*file — serverga jo‘natiladigan fayl adresini kiritish, fayl elementi.

\*hidden — ko‘rinmas element, sahifada ko‘rinmaydi, lekin ko‘p hollarda ishlatiladi.

\*image — rasmlı qator, rasmga bosilganda, axborotlar serverga jo‘natiladi.

\*password — oddiy matnli element, lekin yozuvlari ko‘rinmaydi, parol yozish uchun.

\*radio — bir necha tanlovalardan faqat bittasini tanlash imkonini beruvchi element.

\*reset — forma qiymatlarini dastlabki holatga qaytarish.

\*submit — formadagidagi axborotlarni serverga jo‘natish.

\*text — oddiy matn kiritish elementi.

**align** — tip rasm(**image**) bo‘lganda, rasmni tekislash, qiymatlari: bottom, left, middle, right, top

**alt** — rasmlı tugmada, rasm ochilmasa, uning o‘rnida paydo bo‘ladigan yozuv.

**autocomplete** — elementni avtomat to‘ldirishni o‘rnatish.

**border** — rasm atrofidagi ramka qalinligini o‘rnatish.

**checked** — **checkbox** yoki **radio** uchun dastlabki holatni «**belgilash**» ko‘rinishida namoyon qilish.

<input type=«checkbox»>Messi

<input type=«checkbox» checked=«checked»>Rooney

<input type=«checkbox»>Neymar

**disabled** — elementni blokirovka qilish, ya’ni dastlabki holatni o‘zgartirib bo‘lmaydi.

<input type=«text» disabled value=«Yozolmaysan»>

**form** — boshqa forma bilan identifikator orqali bog‘lanish.

**formaction** — axborotlarni qayta ishlaydigan sahifa adresini berish(form tegi action attributi bilan bir hil).

**formmethod** — axborotlari qaysi metod orqali serverga jo‘natishni belgilash(Get, Post. Form tegi method attributi bilan bir hil).

**maxlength** — foydalanuvchi tomonidan maksimal kiritilishi mumkin bo‘lgan axborot o‘lchami(kiritilgan raqamgacha bo‘lgan simvollar).

**multiple** — element tipi fayl bo‘lganda, bir paytda bir necha fayllar kiritilishi mumkinligini bildiradi.

**name** — forma elementiga takrorlanmas nom berish, bu nom axborot serverga jo‘natilayotganda ishlatiladi.

**readonly** — yozuvli elementda ishlatilsa, faqat yozuvni ko‘rish imkonini yaratadi, o‘zgartirib bo‘lmaydi.

**size** — yozuvli elementlarda(text, password) yozuv yozish mumkin bo‘lgan joyning uzunligini o‘rnatish.

**src** — tip rasm(typeqimage) bo‘lgan elementni adresini ko‘rsatish.

**tabindex** — klaviaturadan nechi marotaba «tab» tugmasi bosilsa, shu elementga o‘tish kerakligini o‘rnatish. Qiymat raqamlardan iborat bo‘ladi.

**value** — forma elementiga dastlabki qiymatni berish.

```

1 <html>
2 <head> <title>Form of Form</title></head>
3 <body>
4 <p align="center">Registratsiya
5 <form name="forma" action="index.php" method="post">
6 <table border=1 bgcolor="#ffffdd" align="center">
7 <tr>
8 <td align="left">
9 Login kiriting:
10 <input type="text" name="search" size="25" value="user name">

11 Parol kiriting:
12 <input type="password" name="password" value="password" size="15">
13

14 <hr color="green">
15 <fieldset><legend>shaxsiy ma'lumotlar</legend>
16 Jinsingiz?

17 <input type="radio" name="club" value="male" checked="checked"> Erkak

18 <input type="radio" name="club" value="female"> Ayol
19 <hr color="green">
20 Yoshingiz?

21 <select name="age">
22 <option value="yigirma">20
23 <option value="yigirma bir">21
24 <option value="yigirma ikki" selected="selected">22
25 </select>
26 <hr color="green">
27 Kasbingiz?

28 <select name="notebook" multiple size="1">
29 <option value="oshpacz">Oshpacz
30 <option value="prog">Dasurchi
31 <option value="admin">Administrator
32 <option value="ras">Rassom
33 <option value="sartosh">Sartanosh
34 <option value="taxi">Taksist
35 </select>
36

37 <hr color="green">
38 <textarea rows="3" cols="45">
39 Katta hajmdagi matnlar textarea tegi ichida yoziladi
40 </textarea>
41 <center><input type="button" value="OK"></center>
42 </tr>
43 </table>
44 </form>
45 </body>
46 </html>

```

**1. <fieldset>...</fieldset>** — formadagi bir necha elementlarni guruhlash uchun ishlataladi. guruhlangan elementlar chiziq bilan chegaralanadi. uning attributlari:

**disabled** — guruhning barcha elementlarini bloklash.

**form** — boshqa formalar bilan bog'lanish(identifikator orqali).

**title** — formadagi guruhga yordam sifatida yozuv chiqarish.

<fieldset>

<input type=<text>> Name<br><input type=<Password> value=<akmx>> Parol<br>

<input type=<checkbox> checked=<checked>> Belgilash

</fieldset>

**2. <legend>...</legend>** — <fieldset> orqali hosil qilingan guruhga nom berish, bu nom chizilgan chegara chizig'ining tepa qismida paydo bo'ladi(misolda **shaxsiy ma'lumotlar** degan nom).

Attributlari quyidagilar:

**accesskey** — klaviatura tugmalari orqali tez o'tishni amalga oshiradi.

**align** — guruh nomini tekislash: left, right, center.

**title** — forma nomiga sichqoncha oborilsa, yordam sifatida yozuv chiqarish.

<fieldset title=<guruh>><legend align=<center>>Umumiyl guruh</legend>

<input type=<text>> Name<br>

<input type=<Password> value=<akmx>> Parol<br>

</fieldset>

Umumiy guruh

Name	
Parol	

3.<select>...</select> — foydalanuvchi tanlashi uchun bir necha variantlarga ega bo‘lgan ro‘yxat hosil qilish. Bu teg «option» tegi bilan birga ishlataladi. Bir yoki bir nechta variantni tanlash imkonini beradi(**size** attributi orqali o‘rnataladi). Uning attributlari:

**accesskey** — klaviatura tugmalari orqali tez o‘tishni amalga oshiradi.

**disabled** — o‘zgartirib bo‘lmaydigan holga olib keladi, ya’ni blokirovka.

**form** — boshqa forma bilan identifikator orqali bog‘lanishni amalga oshirib beradi.

**multiple** — bir vaqtning o‘zida bir necha qiymatni tanlash imkonini beradi.

**name** — serverga jo‘natish uchun element nomini o‘rnatish.

**size** — qiymatlarni nechi qator bo‘lib ekranga chiqarishini o‘rnatish, qolgan qiymatlar ko‘rinmas holda bo‘lib turadi va menuy ochilsa ko‘rinadi.

4. <option>...</option> — «select» orqali hosil qilingan ro‘yxatdagi elementlarni yaratish.

Nechta «option» bo‘lsa, ro‘yxatda shuncha element bo‘ladi. Uning attributlari:

**disabled** — ma’lum bir ro‘yxat elementlarini bloklash.

**label** — ro‘yxat elementiga metka qo‘yish.

**selected** — brauzer yuklanganda shu attribut o‘rnatalgan element tanlangan bo‘lib chiqadi.

**value** — serverga shu nom orqali element qiymati ketadi.

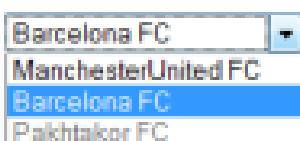
<select name=”club” size=”1”>

<option value=”mufc”>ManchesterUnited FC

<option value=”fcb” selected>Barcelona FC

<option value=”pfc” disabled>Pakhtakor FC

</select>



5. <textarea>...</textarea> — forma elementi bo‘lib, katta hajmdagi matnlarni kiritish uchun ishlataladi. Uning attributlari:

**accesskey** — klaviatura tugmalari orqali tez o‘tishni amalga oshiradi.

**cols** — matn yozish uchun ajratilgan joy uzunligi(raqamda beriladi, nechta simvol kiritilishi).

**disabled** — elementni blokirovka qilish.

**form** — boshqa forma bilan identifikator orqali bog‘lanishni amalga oshirib beradi.

**maxlength** — foydalanuvchi tomonidan kiritilishi mumkin bo‘lgan maksimal simvollar soni.

**name** — elementning nomi.

**readonly** — element ichidagi yozuvni faqat o‘qish mumkin bo‘lgan holga keltirish(o‘zgartirib bo‘lmaydi).

**rows** — matn yozish uchun ajratilgan joy balandligi(qatorlar soni), shu sondan so‘ng pastga tushish uchun aylantirgich hosil bo‘ladi(prokrutka).

**tabindex** — klaviaturadan nechinchi marotaba «tab» tugmasi bosilsa, shu elementga o‘tishni.

**wrap** — yozilgan matnni keyingi qatorga o‘tkazish qoidasini o‘rnatish, uning qiymatlari

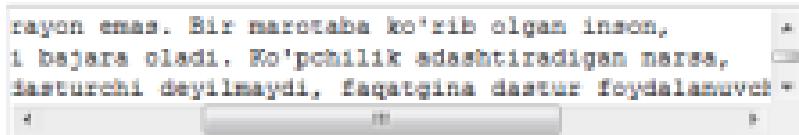
\*soft — matn uzunligi bitta qatorga sig‘masa, avtomat keyingi qatorga o‘tadi.

\*hard — «cols» da berilagan qiymatdan so‘ng uzun matnlar keyingi qatorga o‘tadi.

\*off — matn uzunligi tugamaguncha birinchi qator ham tugamaydi(surilgich chiqadi).

```
<textarea rows=<<3" cols=<<50" readonly wrap=<<off>>
```

```
</textarea>
```



Yuqoridagi kodning natijasi quyidagi holda bo'ladi.

[Registratsiya](#)

A screenshot of a registration form dialog box. It contains fields for login, password, gender (male selected), age (22), and marital status (Oshpaz). A note at the bottom states: "Kette hejmidaqagi matnlar textarea tegi ichida yoziladi". An OK button is at the bottom right.

## Topshiriqlar

Har bir talaba “Web dasturlashga kirish fanidan laboratoriya ishlari” uslubiy ko’rsatmasida berilgan laboratoriya topshiriqlari bo'yicha HTML5 teglarining yangi teglaridan foydalanish, sahifasida forma elementlarini hosil qilish(misol uchun, **registratsiyada, so'rovlarda, avtorizatsiyada, kommentariyalarda** va hokazo.) zarur.

### **3-laboratoriya ishi**

**Mavzu:** CSS da selektorlar. CSS xususiyatlari. CSS3 da effektlar, transformatsiya, animatsiya

**Ishdan maqsad:** Web sahifa shakllantirishda CSS selektorlari, CSS xususiyatlari, CSS3 dagi effektlar, transformatsiya, animatsiyalardan foydalanishni o'rganish va web sayti tuzish ko'nikmalarini hosil qilishdan iborat.

### **Nazariy qism**

CSS – bu stillar bilan ishlay oladigan kaskadli stillar majmuasidir. Web sahifalarga har xil stillar berish uchun ishlataladi. HTML xujjat ichida teglar bilan birga foydalaniladi. CSS kodlarini kompilyastiya qilish uchun qandaydir kompillyatorlar, qandaydir dasturlar, va bu kod tushuna oladigan qandaydir redaktorlar kerak bo'lmaydi. Bu kodlarni HTML singari web brauzerning o'zi kompilyastiya qiladi va natijani chiqarib beradi.

CSS ni HTML xujjati ichida ishlatalishini birin ketin ko'rib o'tamiz: Web sahifada ishlataladigan CSS kodlari orqali hosil qilinadigan stillarni hammasini bitta faylga yozib, uni serverda saqlaymiz. So'ng, sahifadan, serverdagi saqlab qo'yilgan faylga murojaat qilib, kerakli stilni olamiz. Bu usul katta hajmdagi web sahifalar yaratishda foydali hisoblanadi. Faylga murojaat, <head>tegi ichida amalga oshiriladi va quyidagi ko'rinishga ega bo'ladi.

```
<LINK REL=STYLESHEET TYPE=>text/css</LINK>
```

```
<link> tegi sahifa biror faylga murojaatni amalga oshirmoqda degani, REL – fayl qandaydir stillardan iborat degani, TYPE — stillar css kodlaridan iborat va fayl joylashgan manzil yoziladi.
```

CSS kodlar veb sahifa ichidagi <head> tegi orasida beriladi. Bunda yaratilgan stillar shu sahifaga tegishli bo'ladi, bu degani faqat shu sahifada ishlatalish mumkindir.

```
<head><style type=>text/css</style>
<!--
A {text-decoration:none; }
-->
</style></head>
```

Navbatdagi turi, bu har bir element uchun alohida teg ichida stil berib chiqishdan iborat. Misol uchun, “p” tegi ichida biror matn yozilsa, shu matn uchun teg ichida alohida stil beriladi va bu stil shu teg uchun xususiy bo'ladi. Bu usuldan foydalanishni maslahat bermiman, chunki bu usul kodlarni chalkash va tushunarsiz bo'lishiga olib keladi. Shoshilinch vaziyatlarda yoki stil qay tartibda namoyon bo'lishni ko'rib olish uchun bu usuldan foydalanishni maslahat beraman.

Undan tashqari <style> tegi orqali css faylni import qilish mumkin bo'ladi.  
@import: url(mystyles.css);

CSS stillarini ishlatalishda, quyidagi guruhash usullaridan foydalanish, tartibli web sahifa tuzishga olib keladi:

— Kodlarni ixchamlash uchun, xar hil elementlar uchun bir hil tipdagi stillarni guruhlab ishlatalish maqsadga muvofiqdir.

H1 {font-family: Verdana}

H2 {font-family: Verdana}

Yuqoridagi stilni guruhlab quyidagicha yozamiz.

H1, H2 {font-family: Verdana}

— Guruhlashda bir hil element uchun bir necha stillarni birlashtirish mumkin.

H2 {font-weight: bold}

H2 {font-size: 14pt}

H2 {font-family: Verdana}

Barcha stillarni bittaga yig'amiz.

H2 {font-weight: bold; font-size: 14pt; font-family: Verdana;}

— Ba'zi stillar bitta stil ichida berilishi ham mumkin.

H2 {font: bold 14pt Verdana}

CSS stillari bilan ishlaganda, element ichidagi elementlar ham o'zidan yuqori turgan element stilini qabul qiladi, ya'ni o'zidan oldingi element "ota" vazifasini bajaradi va o'z "farzandlari"ga ham o'zida borini beradi. Misol uchun, <p> tegi ichidagi matn ko'k rangda yozilishi lozim bo'lsa(P {color: blue}), <p> tegi ichidagi <em> tegiga tegishli bo'lgan matn ham ko'k rangda yoziladi.

Ba'zi stillar faqat yakka tartibda ishlaydi, ya'ni ichki teglarga stillari o'tmaydi(misol uchun, background), shuning uchun barcha stillarni birma bir ishlatib, o'rganib olish lozim.

<body> tegiga stil berib, dastlabki o'rnatilish(po umolchaniyu) jarayonini hosil qilish mumkin.

Shunda biror elementga stil berish esdan chiqib qolsa, <body> tegida ta'lqli stil esdan chiqgan elementga o'rnatiladi.

BODY {color: green;  
font-family: «Verdana»;  
background: url(joke.gif) white; }

CSS da asosiy fon va rang uchun javob beradigan stil xususiyatlari quyidagilardan iborat:

— color;  
— background-color;  
— background-image;  
— background-repeat;  
— background-attachment;  
— background-position;  
— background.

Bu stil xususiyatlarini birma-bir ko'rib chiqamiz.

Color — u yoki bu elementga rang berish uchun ishlatiladi, misol uchun, hamma paragraf teqlari qizil rangda, barcha h1 teglar qora rangda bo'lish kerak bo'lsa, quyidagicha stil beriladi.

```
<html>
<head>
<title>Rang_1</title>
<style type="text/css">
p{color:red}
h1{color:black}
</style>
</head>
<body>
<h1>Qora</h1>
<p>Qizil</p>
```

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

Qayerda bu teglar ishlatilsa, shu tegga mos holda stillar ham ishlaydi, natijani ko‘ramiz:

# Qora

Qizil

Background-color — biror element yoki butun veb sahifa foniga rang berish. HTML dagi «bgcolor»dan farqli ravishda, bu stil xususiyati istalgan element foniga rang berish mumkin: havolalarga, paragraflarga, jadvallarga,...

Misol ko‘ramiz, butun sahifa foniga va barcha H3 teglari foniga rang beramiz.

```
<html>
<head>
<title>Rang_2</title>
<style type="text/css">
body{background-color:gray}
h3{color:orange}
</style>
</head>
<body>
<h3>Fongini rangini qara, daxshatu a!</h3>
</body>
</html>
```

Rang berishda odatagidek rang nomi(ingliz tilida) yoki 16 lik sanoq tizimda berish lozim(16lik tizimdagি rangi aniqlash), endi natijani ko‘ramiz:

Fongini rangini qara, daxshatu a!

Background-image — sahifa foni sifatida biror rasmni qo‘yish, nafaqat sahifa boshqa elementlarga ham qo‘yish mumkin, faqat yaxshi chiqmasligi ehtimoli katta. Misol ko‘ramiz,

```
<html>
<head>
<title>Rang_3</title>
<style type="text/css">
body{
background-image: url(bum.jpg);
} <
</style>
</head>
<body>
```

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



E'tibor bergen bo'lsangiz, rasmni ko'rsatish HTMLga nisbatan boshqacharoq. «url» kalit so'zi yozilib, so'ng rasm adresi berilmoqda, bu holda rasm sahifa fayli saqlangan papka bilan bir joyda joylashgan, agar boshqa joyda bo'lsa, (image/bum.jpg) ko'rinishida(rasmning to'liq adresi) berish kerak.

Background-repeat — qaytarilishni amalga oshirish. Agar yuqoridagi misolga e'tibor bergen bo'lsangiz, kichik rasmlar yuqori chap burchak bo'lab qaytarilib takrorlanmoqda va shu tariqa butun veb sahifa foni rasmga to'lmoqda.

Background-repeat shu qaytarilishni boshqara oladi. Bu stil xususiyatining quyidagi qiymatlari mavjud:

Background-repeat: repeat-x ; — gorizontal bo'yicha qaytarilish.

Background-repeat: repeat-y ; — vertikal bo'yicha qaytarilish.

Background-repeat: repeat ; — vertikal va gorizontal bo'yicha qaytarilish.

Background-repeat: no-repeat ; — qaytarilishni amalga oshmaydi.

Bunday xususiyatlar HTML gippermattin tilida mavjud emas. Keling misol ko'ramiz,

```
BODY {
background-image : url(bum.jpg) ;
background-repeat: repeat-x;
}
```

Fon sifatida tanlangan rasm, faqat gorizontal bo'yicha takrorlanadi.



Background-attachment — agar fon sifatida rasm bo'lsa, bu stil xususiyati o'rnatiladi. Bunda, agar sahifada o'tkazgich(prokrutka)bo'lsa, fon rasmi ham o'tish kerak yoki fon rasmi sahifaga qotirib qo'yilishi kerakligi ko'rsatiladi. Qiymatlari:

scroll — fon sifatida o'rnatilgan rasm, veb sahifa oynasi bilan birga o'tadi (prokrutka).  
fixed— fon sifatida o'rnatilgan rasm, veb sahifa oynasiga qotirib qo'yilgan bo'ladi va o'zgarmaydi.

BODY {

background-image : url(bum.png) ;

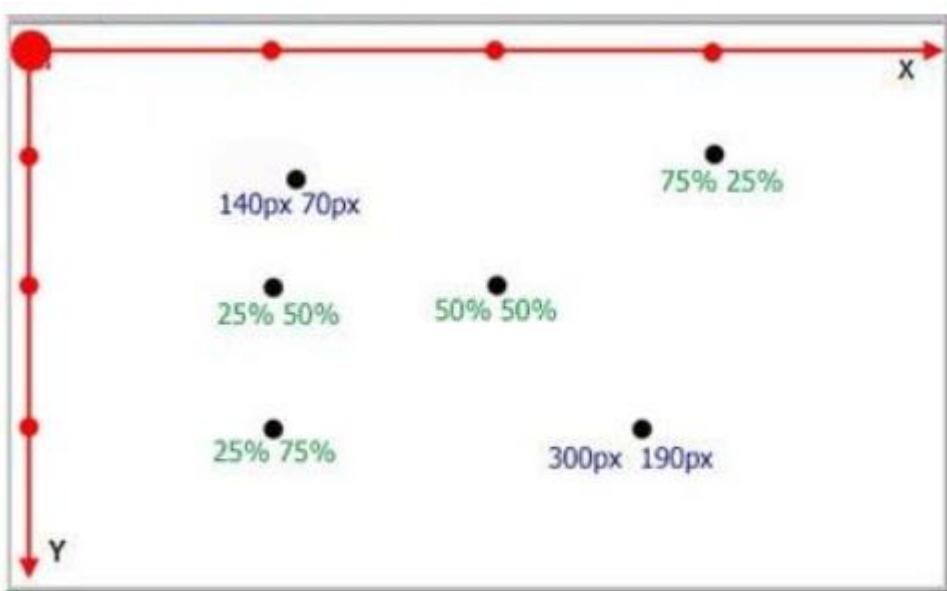
background-repeat: no-repeat;

background-attachment: scroll ;

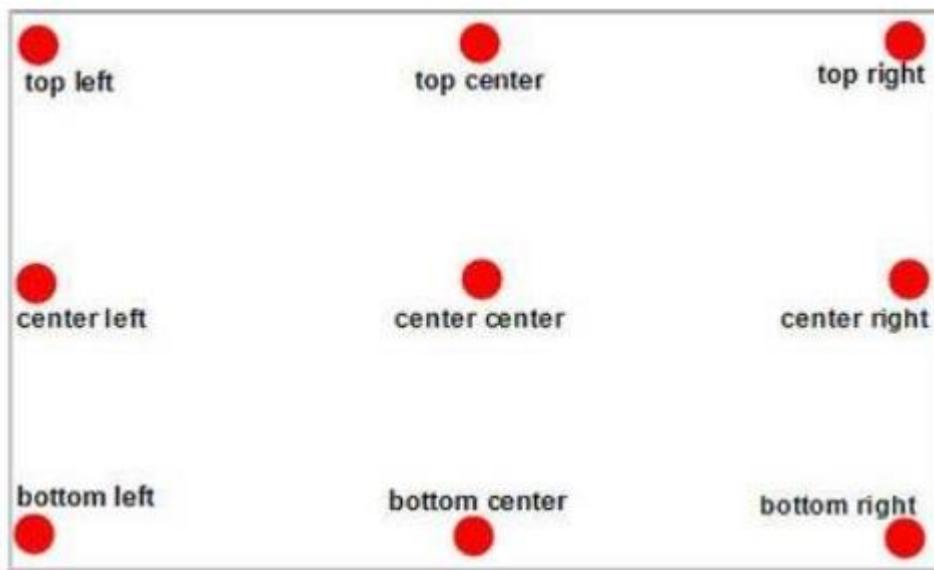
}

Background-position — fon sifatida tanlangan rasmni pozitsiyasini o'rnatadi, ya'ni fon rasmi aynan qayerda joylashishini aniqlashtirish, qiymat foiz yoki uzunlik o'lchovida beriladi.

Boshlanish, chap yuqori tomondan boshlanadi. Quyidagi rasmda pozitsiya haqida yaqqol ko'rsatilgan.



Dastlab, gorizontal bo'yicha qiymat beriladi, keyin vertikal bo'yicha. Qiymat brauzer o'lchamiga qarab piksel yoki foizda beriladi. Undan tashqari kalit so'zlar orqali ham pozitsiyani berish mumkin.



Keling endi misol ko‘ramiz,

```
<html>
<head>
<title>Rang_4</title>
<style type="text/css">
BODY {
background-image : url(bum.jpg);
background-repeat: no-repeat;
background-position: center center;
} <
</style>
</head>
<body>
</body>
</html>
```



Yana bir misol,

```
BODY {
background-image : url(bum.jpg) ;
background-repeat: no-repeat;
background-position: 300px 500px ;
}
```

Background — bu umumiy, hamma xarakteristikalarini ishlatalish uchun foydalilaniladi, ya’ni background-color, background-image, background-repeat, background-attachment, background-position stil xususiyatlarini bitta shu stil orqali ishlatalish mumkin, ya’ni

```
BODY {
background-color:#ffee8c ;
background-image : url(bum.png) ;
background-repeat: no-repeat;
background-attachment: fixed;
background-position: top right;
}
```

Shu uzun kodni, bitta «background» orqali yozish mumkin:

```
BODY {
background: #ffee8c url(bum.png) no-repeat fixed top right ;
}
```

Agar biror xususiyatini kiritmasak, doimiy qiymatni qabul qilib, ishlataadi.

## **Topshiriqlar**

Har bir talaba “Web dasturlashga kirish fanidan laboratoriya ishlari” uslubiy ko’rsatmasida berilgan laboratoriya topshiriqlari bo'yicha HTML teglari asosida tayyorlagan web sahifalarini dizayniga ishlov berish uchun CSS texnologiyalaridan foydalanadi.

## **4-laboratoriya ishi**

**Mavzu** JavaScript brauzer va veb-hujjat obyektlari modeli bilan ishlash (BOM, DOM). HTML5 da JavaScript API lardan foydalanish.

**Ishdan maqsad:** Javascript tili asosiy konstruksiyalarini o'rganish.

### **Nazariy qism**

Javascript – bu Webni gipermatnli sahifalarini klient tomonida ko'rish senariyalarini boshqarish tili. Yanada aniqroq aytadigan bo'lsa, Javascript – bu nafaqat klient tomonidagi dasturlash tili. Liveware Javascript tilining avlodи bo'lib, Netscape serveri tomonida ishlovchi vosita bo'ladi. Ammo Javascript tilini mashhur qilgan narsa bu klient tomonida dasturlashdir.

Javascriptning asosiy vazifasi – HTML-konteynerlar atributlarining qiymatlarini va ko'rsatuvchi muhitining hossalirini HTML-sarlavxalarni ko'rish jarayonida foydalanuvchi tomonidan o'zgartirish imkoniyatlarida, boshqacha aytganda ularni dinamik sarlavxalar qilish (DHTML). Yana shuni aytish joizki, sarlavxalar qayta yuklanmaydi Amalda buni, masalan, quydagicha ifodalash mumkin, sarlavxaning fonini rangini yoki xujjatdagi rasmni o'zgartirish, yangi oyna ochish yoki ogoxlantirish oynasini chiqarish. "JavaScript" nomi Netscape kompaniyasining hususiy maxsuloti. Microsoft tomonidan amalga oshirilgan til rasman Jscript deb nomlanadi. Jscript versiyalari Javascriptning mos versiyalari bilan mos keladi (aniqroq qilib aytganda oxirigacha emas).

Javascript – ECMA (European Computer Manufacturers Association – Evropa Kompyuter Ishlab Chiqaruvchilar Assotsiyatsiyasi) tomonidan standartlashtirilgan. Mos standartlar quydagicha nomlanadi ECMA-262 va ISO-16262. Ushbu standartlar bilan Javascript 1.1ga taqriban ekvivalent ECMAScript tili aniqlanadi. Eslatish joizki, bugungi kunda Javascript ning hamma versiyalari ham ECMA standartlariga mos kelavermaydi. Mazkur kurs yoki qo'llanmada barcha hollarda biz Javascript nomidan foydalanamiz. Javascriptning asosiy hususiyatlari. Javascript – bu Internet uchun katta bo'limgan klient va server ilovalarni yaratishga mo'ljallangan nisbatan oddiy ob'ektga yo'naltirilgan til. Javascripttilida tuzilgan dasturlar HTML-xujjatning ichiga joylashtirilib ular bilan birga uzatiladi. Kurish dasturlari (brauzerlar – browser ingliz suzidan) Netscape Navigator va Microsoft Internet Explorer xujjat matniga joylashtirilgan dasturlarni (Scriptkod) uzatishadi va bojarishadi. Shunday qilib, Javascript – interpritatorli dasturlash tili xisoblanadi. Javascriptda tuzilgan dasturlarga foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni tekshirayotgan yoki xujjatni ochganda yoki yopganda biror bir amallarni bagaruvchi dasturlar misol bo'lishi mumkin. JavaScript da yaratilgan dasturlarga misol sifatida foydalanuvchi tomonidan kiritilgan ma'lumotlarni tekshiruvchi, dokumentni ochish yoki yopish vaqtida qandaydir amallarni bajaruvchi dasturlarni keltirish mumkin. Bunday dasturlar foydalanuvchi tomonidan berilgan kursatmalarga – sichqoncha tugmachaсинi bosilishiga, ma'lumotlarni ekran orqali kiritishiga yoki sichqonchani sahifa buylab siljitimishiga kura ish bajaradi. Bundan tashqari JavaScript dagi dasturlar brauzerning uzini va dokumentning atributlarini ham boshqarishi mumkin. JavaScript dasturlash tili sintaktik jihatdan Java dasturlash tiliga, ob'ektli modellashni istisno qilgan holda, o'hshab ketsada, lekin ma'lumotlarni statik tiplari va qat'iy tiplashtirish kabi hususiyatlarga ega bulmaydi. JavaScript da Java dasturlash tilidan farq qilib, sinf (klass) tushunchasi bu tilning asosiy sintaktik qurilmasi hisoblanmaydi. Bunday asos sifatida foydalanilayotgan tizim tomonidan qullab-quvvatlanayotgan, oldindan aniqlangan ma'lumot tiplari: sonli, mantiqiy va satrli; mustaqil ham bulishi, ob'ektning metodi (JavaScriptda metod tushunchasi funksiya/qism-dastur ning uzi) sifatida ham ishlatilishi mumkin bulgan funktsiyalar; katta sondagi uz hossalariga va metodlariga ega bulgan oldindan aniqlangan

ob'ektlardan iborat ob'ektl model va yana dastur ichida foydalanuvchi tomonidan yangi ob'ektlarni berish qoidalari hisoblanadi.

JavaScript da dasturlar yaratish uchun hech qanday qo'shimcha vositalar kerak bulmaydi – faqatgina tegishli versiyadagi JavaScript qullanishi mumkin bulgan brauzer va DHTML-dokumentlarni yaratishga imkon beruvchi matn muharriri kerak bo'ladi. JavaScript dagi dastur bevosita HTML –dokumentlarni ichiga joylashtirilganligi uchun dastur natijasini dokumentni brauzer yordamida kurish orqali tekshirish mumkin va kerakli holda uzgartirishlar kiritilishi mumkin. JavaScript dasturlash tilining imkoniyatlari. Uning yordamida HTML –dokumentlarning ko'rinishi va tuzilishini dinamik ravishda boshqarish mumkin. Ekranda tasvirlanayotgan dokumentga brauzer tomonidan yuklangan dokumentning sintaktik tahlil qilish jarayonida istalgan HTML-kodlarni joylashtirish mumkin. "Dokument" ob'ekti yordamida foydalanuvchining oldingi bajargan amallari yoki boshqa bir faktorlarga kura yangi dokumentlarni avtomatik hosil qilish mumkin. JavaScript yordamida brauzer ishini boshqarish mumkin. Masalan, Window ob'ekti suzib yuruvchi oynalarini ekranga chiqarish, brauzerning yangi oynalarini yaratish, ochish va yopish, oynalarning yugurdagi va o'lchamlarining rejimlarini o'rnatish va hokazolarni imkoniyatini beruvchi metodlarga ega. JavaScript dokumentdagi ma'lumotlar bilan bog'lanish imkoniyatini beradi. Document ob'ekti va undagi mavjud ob'ektlar dasturlarga HTML-dokumentlarning qismlarini o'qish va bazida ular bilan bog'lanish imkoniyatini beradi. Matnning o'zini o'qish mumkin emas, lekin masalan berilgan dokumentdagi gipermatnli o'tishlar ro'yhatini olish mumkin. Hozirgi vaqtida Form ob'ekti va undagi mavjud bo'lishi mumkin bo'lgan ob'ektlar: Button, Checkbox, Hidden, Password, Radio, Reset, Select, Submit, Text va Textarealar dokumentdagi ma'lumotlar bilan bo'g'lanish uchun keng imkoniyatlar beradi. JavaScript foydalanuvchi bilan aloqa qilishga imkon beradi. Bu tilning eng muhim hususiyati unda amalga oshirilgan hodisalarni qayta ishlashni aniqlish imkoniyati – ma'lum bir hodisaning (odatda foydalanuvchi tomonidan bajarilgan amal) ro'y berish vaqtida bajariladigan dastur kodining ihtiyyoriy qismi hisoblanadi. JavaScript hodisalarni qayta ishlovchi sifatida ihtiyyoriy yangi oldindan berilgan funksiyalardan foydalanish imkoniyatini beradi. Masalan, foydalanuvchi sichqoncha ko'rsatkichini gipermatnli o'tishlar ustiga keltirsa, holatlar satrida mahsus habarni chiqaruvchi yoki ma'lum bir amalni bajarishni tasdiqlashni so'rovchi dialogli oynani ekranga chiqaruvchi yoki foydalanuvchi tomonidan kiritilgan qiymatlarni tekshiruvchi va hatolik yuz bergen holda kerakli kursatmalarni berib, tug'ri qiymatni kiritishni so'rovchi dasturlarni yaratish mumkin. JavaScript ihtiyyoriy matematik hisoblashlarni bajarish imkoniyatini beradi. Bundan tashqari bu tilda vaqt va sanalarning qiymatlari bilan ishlovchi yuqori darajada rivojlangan vositalar mavjud. JavaScript CGI-dasturlarga va Perl dasturlash tiliga va to'diruvchi sifatida ayrim hollarda Java tiliga muqobil til sifatida yaratilgan. Har bir boshlovchi dasturchining asosiy savoli: "Dasturlar qanday tuziladi va bajariladi? ". Bu savolga iloji boricha soddarroq, lekin JavaScript-kodlarini qo'llanilishining barcha usullarini unutmagan holda javob berishga harakat qilamiz. Birinchidan, JavaScript-kodlari brauzer tomonidan bajariladi. Unda mahsus JavaScript interpretatori mavjud. Unga kura programmaning bajarilishi interpretator tomonidan boshqaruvni qachon va qay tarzda olishiga bog'liq bo'ladi. Bu esa, o'z navbatida kodning funksiyaviy qo'llanilishiga bo'g'liq bo'ladi. Umuman olganda JavaScript ning funktsional qo'llanishining 4 hil usulini ajratib ko'rsatish mumkin:

1. gipermatnli o'tish (URL sxemasi);
2. hodisalarni qayta ishlash (handler);
3. o'miga qo'yish(entity)
4. qo'yish (SCRIPT konteyneri).

JavaScript bo'yicha o'quv qo'llanmalarida JavaScript ni qo'llashning bayoni odatda SCRIPT konteyneridan boshlanadi. Lekin dasturlash nuqtai nazaridan bu unchalik ham to'g'ri emas, chunki bunday tartib asosiy savol: "JavaScript-kodi boshqaruvni qanday oladi?" ga javob bermaydi. Ya'ni JavaScript da yozilgan va HTML-dokumentning ichiga joylashtirilgan dastur qanday tarzda chaqiriladi va bajariladi. HTML-sahifa muallifining kasbi va uning dasturlash asoslardan habardarligining darajasiga qarab JavaScript ni o'zlashtirishga kirishishni bir necha hil variantlari mavjud. Agar siz klassik tillar (C, Fortran, Pascal va h.) bo'yicha dasturlovchi bo'lsangiz, u holda dokument matni ichida dasturlashdan boshlagan ma'qul, agar siz Windows sistemasida dasturlashga o'rgangan bo'lsangiz, u holda hodisalarни qayta ishlashni dasturlashdan boshlaganingiz ma'qul, agar siz faqat HTML bo'yichagina tajribaga ega bo'lsangiz yoki anchadan beri dasturlash bilan shug'ullanmayotgan bo'lsangiz, u holda gipermatnli o'tishlarni dasturlashdan boshlaganingiz ma'qul.

URL-sxemali JavaScript. URL sxemasi (Uniform Resource Locator)- bu Web- texnologiyalarning asosiy elementlaridan biri. Web dagi har bir informatsion resurs uzining o'ziga hos URLiga ega bo'ladi. URL A konteynerining HREF atributida, IMG konteynerining SRC atributida, FORM konteynerining ACTION atributida va h.larda ko'rsatiladi. Barcha URL lar resursga ruhsatning protokoliga bog'liq bo'lgan ruhsat sxemalariga bo'linadi, masalan, FTP-arkiviga kirish uchun ftp sxemasi, Gopher-arkiviga kirish uchun gopher sxemasi, elektron maktublarni jo'natish uchun smtp sxemasi qo'llaniladi. Sxemaning tipi URLning birinchi komponentasiga ko'ra aniqlanadi: http://directory/page.html

Bu holatda URL http bilan boshlanayapti- mana shu kirish sxemasini aniqlashdir (sxema http). Gipermatnli sistemalar uchun dasturlash tillarining asosiy vazifasi gipermatnli o'tishlarni dasturlashdir. Bu shuni bildiradiki, u yoki bu gipermatnli o'tishlarni tanlashda gipermatnli o'tishni amalga oshiruvchi dastur chaqiriladi. Web-texnologiyalarida standart dastur sifatida sahifani yuklash dasturi hisoblanadi. HTTP protokoli bo'yicha standart o'tishni JavaScript da dasturlangan o'tishdan farq qilish uchun yaratuvchilar yangi URL sxemasi – JavaScript ni kiritishdi:

```
<A HREF=<JavaScript:JavaScript_kod">...
<IMG SRC=<JavaScript:JavaScript_kod">
```

Bu holatda "JavaScript\_kod" matni birinchi holatda gipermatnli o'tishni tanlanganda chaqiriladigan va ikkinchi holatda rasmni yuklashda chaqiriladigan JavaScript da yaratilgan dasturiy-qayta ishlovchilarni bildiradi.

Masalan, Внимание!!! gipermatnli o'tishga keltirilgan holda ogohlantiruvchi oynani chiqarish mumkin:

```
<A HREF=<JavaScript: alert(' Внимание!!!');"> Внимание!!!
```



Formadagi submit tipidagi tugmachani bosish orqali shu formadagi matnli maydonni to'ldirish mumkin:

```

<FORM NAME= METHOD=post
ACTION=<>JavaScript: window.document.f.i.VALUEq' Click tugmani bosing';void(0);"
<TABLE BORDER=0>
<TR>
<TD><INPUT NAME=i></TD>
<TD><INPUT TYPE=submit VALUE=Click></TD>
<TD><INPUT TYPE=reset VALUE=Reset></TD>
</TABLE>
</FORM>

```

URLda murakkab dasturlar va funktsiya chaqirilishlarini joylashtirish mumkin. Faqatgina shuni yodda tutish kerakki, JavaScript sxemasi hamma brauzerlarda ham ishlamaydi, faqatgina Netscape Navigator va Internet Explorer larning to'rtinchi versiyalaridan boshlab ishlaydi.

Shunday qilib gipermatnli o'tishlarni dasturlashda interpretator boshqaruvni foydalanuvchi sichqoncha tugmasini gipermatnli o'tishga "bosgandan" keyingina oladi.

O`qish: 752

**Fikringiz uchun rahmat:**

**Ma'lumot qoldirish:**

Ismingiz:

Fikrigizni qoldiring:

Rasmdagi raqamni kriting

$$3+2=$$

**Tasdiqlash**

### Topshiriqlar

Har bir talaba "Web dasturlashga kirish fanidan laboratoriya ishlari" uslubiy ko'rsatmasida berilgan laboratoriya topshiriqlari bo'yicha HTML teglari asosida tayyorlagan web sahifalarida JavaScript brauzer va veb-hujjat obyektlari modeli bilan ishlashni amalga oshiradi.

## 5-laboratoriya ishi

### Mavzu: JQuery kutubxonasi. Selektorlar. Hodisalar.

**Ishdan maqsad:** JQuery kutubxonasi haqida tushunchaga ega bo'lish. Selektor va hodisalarni o'rGANISH.

#### Nazariy qism

JQuery – bu tez, kichik va boy JavaScript kutubxona. U orqali HTML hujjat ob'yekt modeli manipulyatsiyalari, hodisalarni qayta ishslash, animatsiyalar bajarish, AJAX so'rovlari bilan ishslashni osonlashtirish chun mo'ljallangan.

Misollar:

#### 1. Yashirish va ko'rsatish.

```
<html>
<head>
<link rel=<>stylesheet" href="bootstrap.min.css"/>
<script src=<>jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#hide").click(function(){
 $("p").hide();
 });
 $("#show").click(function(){
 $("p").show();
 });
});
</script>
</head>
<body>
<div class=<>container well">
<p class=<>text-center">If you click on the "Hide" button,I will disappear.</p>
<button id=<>hide" class=<>btn btn-success btn-sm">Hide</button>
<button id=<>show" class=<>btn btn-warning btn-sm">Show</button>
</div>
</body>
</html>
```



#### 2. Matnni yashirish.

```
<html>
<head>
<link rel=<>stylesheet" href="bootstrap.min.css"/>
<script src=<>jquery.min.js"></script>
<script>
$(document).ready(function(){
```

```

$("p").click(function(){
$(this).hide();
})
})
</script>
</head>
<body>
<div class=<>container">
<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p></div>
</body>
</html>

```



### 3. Focus va Blur

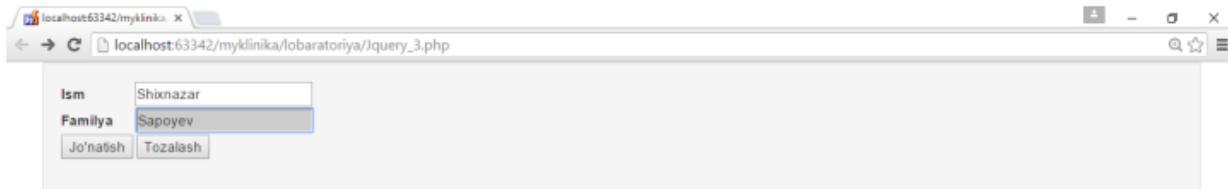
```

<html>
<head>
<link rel=<>stylesheet" href="bootstrap.min.css"/>
<script src=<>jquery.min.js"></script>
<script>
$(document).ready(function(){
$("input").focus(function(){
$(this).css("background", "#cccccc");
})
$("input").blur(function(){
$(this).css("background", "#ffffff")
})
})
</script>
</head>
<body>
<div class=<>container well">
<form action="JQuery_1.php">
<label for=<>ism">Ism </label><input id=<>ism" style="margin-left: 50px" type="text"/>

<label for=<>fam">Familya </label><input id=<>fam" style="margin-left: 22px" type="text"/>

<input type="submit" value="Jo'natish"/>
<input type="reset" value="Tozalash"/>
</form>
</body>
</html>

```

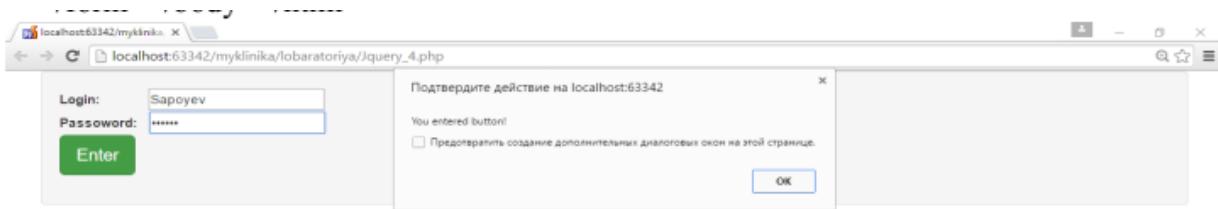


#### 4. Hover hodisasi.

```
<html>
<head>
<link rel=<stylesheet" href="bootstrap.min.css"/>
<script src=<jquery.min.js"></script>
<script>
$(document).ready(
function(){
 $("#btn").hover(function(){
 alert("You entered button!");
 },
 function(){
 alert("By! You now leave button!");
 })
}
</script>
</head>
<body>
<div class=<container well">
<form action="JQuery_1.php">
<label for=<ism">Login: </label><input id=<ism" style="margin-left: 50px"
type="text"/>

<label for=<fam">Passoword: </label><input id=<fam" style="margin-left: 12px"
type="password"/>

<button class=<btn btn-success btn-lg" id=<btn">Enter</button>
</form></body></html>
```



#### 5. Mouseleave hodisasi.

```
<html>
<head>
<link rel=<stylesheet" href="bootstrap.min.css"/>
<script src=<jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#in").mouseleave(function(){
 alert("Emailingizni kiritdingizmi!")
 })
})
```

```

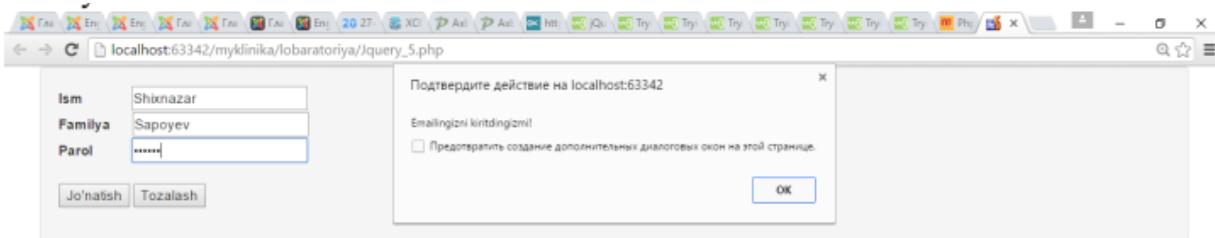
</script>
</head>
<body>
<div class=<>container well">
<form action="JQuery_1.php">
<label for=<>ism">Ism </label><input id=<>ism" style="margin-left: 50px"
type="text"/>

<label for=<>fam">Familya </label><input id=<>fam" style="margin-left: 22px"
type="text"/>

<label>Parol </label><input id=<>in" style="margin-left: 40px" type="password"/>

<input type="submit" value="Jo'natish"/>
<input type="reset" value="Tozalash"/>
</form></div>
</body></html>

```



## Topshiriqlar

Har bir talaba “Web dasturlashga kirish fanidan laboratoriya ishlari” uslubiy ko’rsatmasida berilgan laboratoriya topshiriqlari bo'yicha HTML teglari asosida tayyorlagan web sahifalarida JQuery kutubxonasidagi selektor va hodisalar bilan ishlashni amalga oshiradi.

## **6-Laboratoriya ishi**

### **Mavzuy: Bootstrap frameworkini ularash, bazaviy shablonlar**

**Ishdan maqsad:** Twitter bootstrap CSS frameworki haqida bilimlarga ega bo'lish.

#### **Nazariy qism**

Bootstrapga twitterning asoschisi asos solgan va shuning uchun ham nomi twitter bootstrap deb nomlangan. Bootstrap bu soda va oson quriladigan HTML, CSS va Javascript framework hisoblanadi. Zamonaviy, standartlashgan, qulay va tez ishlaydigan saytlarni qurishimiz uchun yordam beradi.

Bootstrapning qulayliklari:

1. Qisqa vaqt ichida sayni va shablonlarni noldan, osongina qurish.
2. Adaptiv saytlarni va shablonlarni qurishda yengillik.
3. Aloqalarni tashkil qilishda qo'shimcha kutubxonalarining ko'pligi.
4. Tayyor CSS fayllar

Bu bootstrapning bizga kerak bo'ladigan, asosiy qulayliklari edi.

Bootstrapning asosiy qurilmalari:

Setkalar- bu Grid system deyiladi

Shablonlar- feksirovanniy yoki dokumentning rezina shabloni

Tipografika- shriftlar haqida ma'lumot

Media- rasmlar va videolarni boshqarish uchun

Tablitsalar- tablitsalarni oformleniya qilish, ularni saralash uchun

Formalar- faqatgina formalarni oformit qilish uchun va ular ustida ish olib boorish uchun

Navigatsiya- Navigation bar yoki navbar deyiladi. Saytlarning yuqori qismidagi instrumentlar paneli xisoblanadi. U orqali web saytning boshqa sahifalarini ko'rishimiz mumkin.

Alertlar- ogohlantiruvchi vosita

Bootsrapning fayllar strukturasi quyidagicha:

Bootstrap/

css/

  bootstrap.css

  bootstrap.min.css

js/

  bootstrap.js

bootstrap.min.js

img/

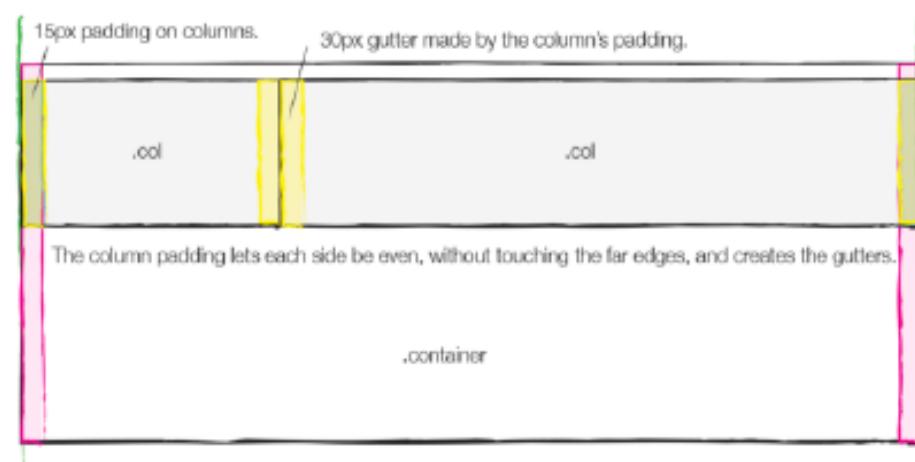
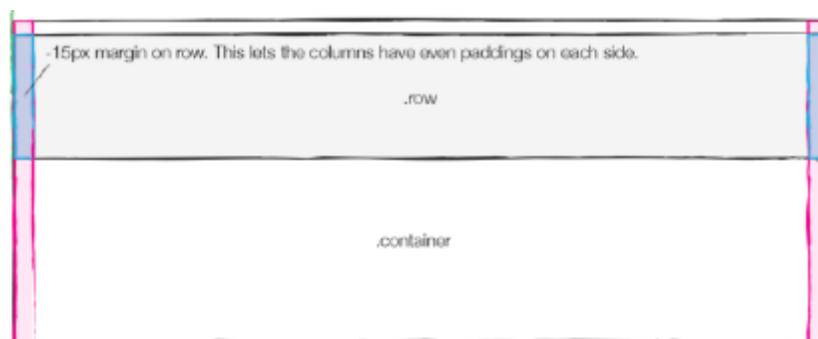
glyphicon-halflings.png

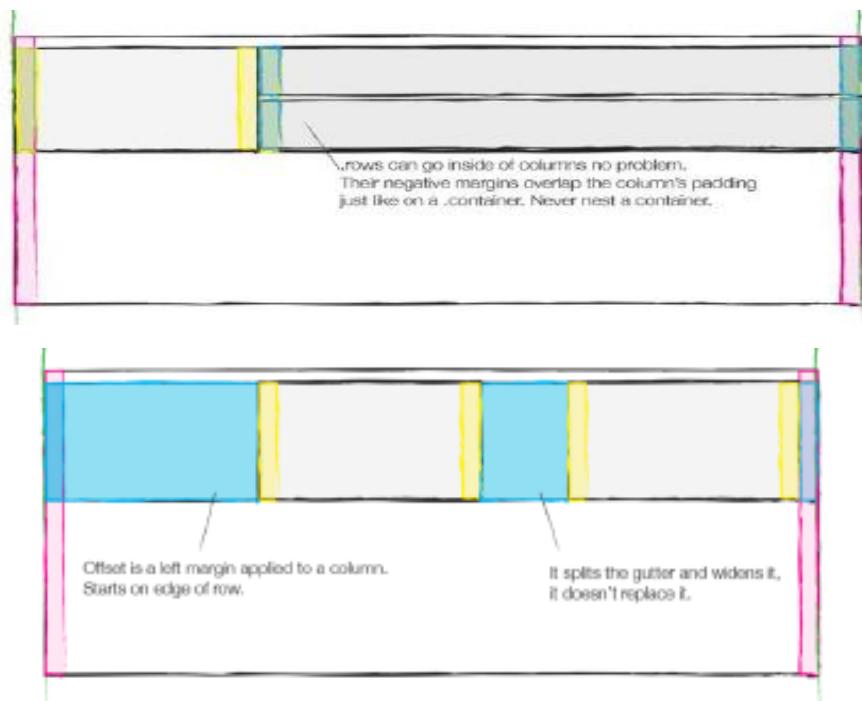
glyphicon-halflings.whi

README.md

Grid systemda container, row, col-lg-2, col-md-6, col-sm-3, col-xs-4, col-offset-lg-1 va shu kabi

boshqa asosiy tushunchalar bor. Ular saytlarning standart holda bo'lishini taminlab beradi.





```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title></title>
5 <link rel="stylesheet" href="resources/bootstrap/css/bootstrap.min.css"/>
6 </head>
7 <body style="background-color: #deefd7">
8 <div class="container">
9 <h1 class="text-center text-info">What we do</h1>

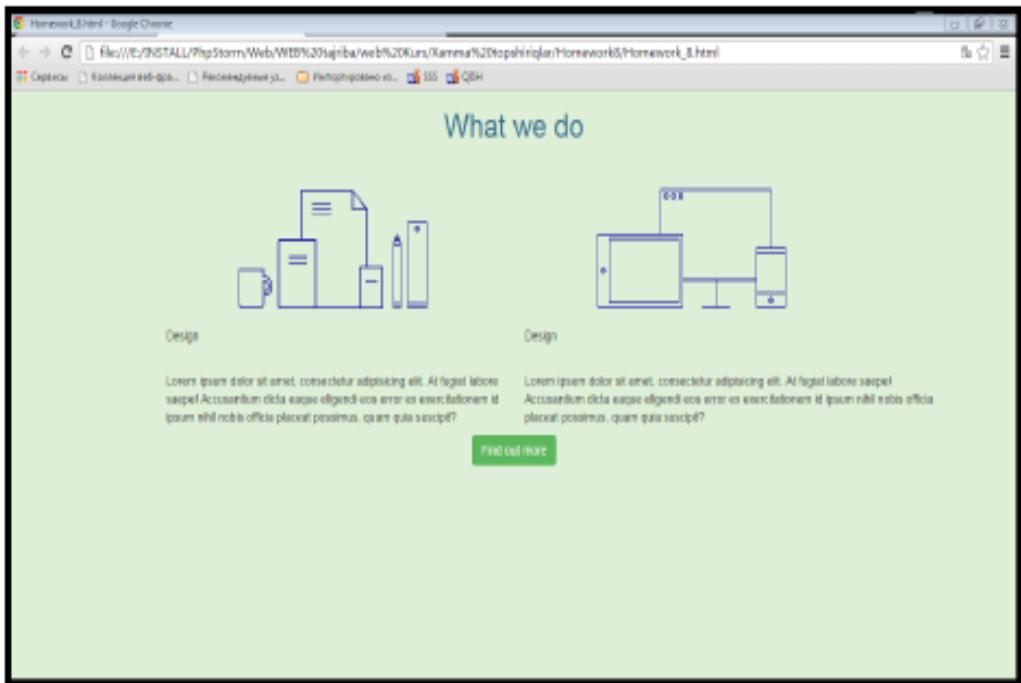
10
11 <div class="row">
12 <div class="col-md-4 col-md-offset-2">
13

14 </div>
15
16 <div class="col-md-4 col-md-offset-1">
17

18
19 </div>
20 </div>
21 <div class="row">
22 <div class="col-md-5 col-md-offset-1">
23 <p>Design</p>

24 <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. At fugiat
labore saepe! Accusantium dicta eaque eligendi eos error ex exercitationem id
ipsum nihil nobis officia placeat possimus, quam quia suscipit?</p>
25 </div>
26 <div class="col-md-6">
27 <p>Design</p>

28 <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. At fugiat
labore saepe! Accusantium dicta eaque eligendi eos error ex exercitationem id
ipsum nihil nobis officia placeat possimus, quam quia suscipit?</p>
29 </div>
30 </div>
31 <button class="btn btn-success center-block">Find out more</button>
32
33 </div>
34 </body>
35 </html>
```



## Topshiriqlar

Har bir talaba “Web dasturlashga kirish fanidan laboratoriya ishlari” uslubiy ko’rsatmasida berilgan laboratoriya topshiriqlari bo'yicha HTML teglari asosida tayyorlagan web sahifalarini Bootstrap4 shablonlari asosida tayyorlashni amalga oshiradi.

## 7-Laboratoriya ishi

### Mavzu: PHP da satrlar va massivlar bilan ishlash

**Ishning maqsadi:** PHP da satrlar va massivlar bilan ishlashni o‘rganish.

#### Nazariy qism

String (satr) tipi.

Satr – bu belgilar to‘plamidir. PHP dasturlash tilida belgi bu bir bayt va 256 ta turli belgilar mavjud. PHP dasturlash tili Unicode tipidagi belgilarni qabul qilmaydi. PHP dasturlash tilida amalda satrlarga chegirma mavjud emas, shuning uchun satrlarni ishlatganda uning aniq uzunligi haqida o‘ylash shart emas.

PHP dasturlash tilida satrlar uchta turli xil usullarda aniqlanadi:

bittalik qo‘shtirnoqlar yordamida (‘);

qo‘shtirnoqlar yordamida (“”);

heredoc-sintaksisi yordamida.

Bittali tirnoqlar

Satrlarning aniqlashning oddiy usuli – u «'» bittali qo‘shtirnoqlar ichida yoziladi. Agarda satr ichida ham bittali tirnoq ishlatishga to‘g‘ri kelib qolsa, bittali tirnoqdan oldin «\» belgini qo‘yish, ya’ni uni ekranlash shart. Agarda «\» belgi bittali tirnoqdan oldin yoki satrning oxirida bo‘lsa, u holda belgini ikkilantirish kerak, ya’ni «\\'».

Agarda bittali tirnoq ichidagi satr ichida ixtiyoriy belgidan oldin («\» va «'» lardan farqli ravishda) teskari slesh «\» belgisi uchrasa, u holda uni oddiy belgi deb qarab barcha belgilarni o‘z holicha ekranga chiqaradi. Shuning uchun teskari slesh «\» belgisini satr oxirida yopiladigan qo‘shtirnoqdan avval turganini ekranlash shart.

PHP dasturlash tilida teskari slesh «\» belgisi bilan ifodalanadigan bir qator belgilar majmui mavjud. Ularni ketma-ketlikni boshqaruvchilar deb ataladi hamda ular maxsus vazifalarni bajaradi. Ular haqida keyinroq to‘xtalib o‘tamiz. O‘zgaruvchilar va ketma-ketlikni boshqaruvchilar bittalik qo‘shtirnoqlar satri ichida uchrashsa, ular o‘rtasidagi farq ketma-ketlikni boshqaruvchilarni qayta ishlanmaydi.

```
<?php
```

```
echo'Satrlar majmui';
```

```
// Ekranga chiqaradi:’ belgini chiqarish uchun
```

```
// undan oldin \ belgi qo‘yiladi.
```

```
echo' Belgini \' chiqarish uchun undan oldin'
```

```
‘ \\ belgini qo‘yish kerak';
```

```
// Ekranga chiqaradi: Siz shuni o‘chirmoqchimisiz C:*.*?
```

```
echo' Siz shuni o‘chirmoqchimisiz C:*.*?';
```

```

// Ekranga chiqaradi: Buni qo‘ymang: \n
// yangi qatorga
echo' Buni qo‘ymang: \n yangi qatorga';
// Ekranga chiqaradi: o‘zgaruvchi $expand ham
// $either qo‘yilmaydi
echo'o‘zgaruvchi $expand ham $either' .
‘qo‘yilmaydi’;
?>

```

Array (massiv) tipi.

PHP dasturlash tilida massiv tipi tartiblangan kartalarga o‘xshaydi va qiymatini kalitga o‘zlashtiradigan tipdir. Bu tip bir necha yo‘nalishlarda optimallashtiriladi, shuning uchun siz uni xususiy massiv, ro‘yxat (vektor), xesh-jadvali (kartani amalga oshirish uchun ishlataladi), stek, navbat va boshqalar sifatida foydalanishingiz mumkin. Modomiki, PHP dasturlash tilida bir massivni qiymatini boshqasiga o‘zlashtirish uchun daraxtlardan foydalanasiz.

Massivlarni array() konstruksiyasi yordamida aniqlanadi yoki elementlariga qiymat berish bilan aniqlanadi.

array() konstruksiyasi yordamida aniqlash.

array ([key] => value,

[key1] => value1, ... )

PHP dasturlash tilining array() konstruksiyasi vergul bilan ajratilgan juft parametrlar kalit => qiymat bilan ajratilgan. => belgi mos ravishda qiymat va uning kaliti o‘rtasida aloqa o‘rnatadi. Kalit butun son bo‘lishi mumkin, uning qiymati esa PHP dasturlash tilidagi ixtiyoriy tipni qabul qilishi mumkin. Kalit raqamini biz ko‘pincha indeks deb ataymiz. PHP dasturlash tilida indekslash noldan boshlanadi. Massiv elementining qiymatini olish uchun massiv nomi va kvadrat qavs ichida uning kaliti ko‘rsatilishi kerak. Agar massiv kaliti standart butun son bo‘lsa, u holda uning qiymatini butun son deb qarasa bo‘ladi, aks holda u satr deb qaraladi. Shuning uchun \$a["1"] yozuv \$a[1] yozuvga teng kuchli, \$a["-1"] yozuv esa \$a[-1] yozuvga teng kuchli.

Misol. PHP dasturlash tilida massivlar

```

<?php
$books = array ("php" =>
 "PHP users guide",
 12 => true);
echo $books["php"];
//ekranga chiqaradi: "PHP users guide"
echo $books[12]; //ekranga chiqaradi: 1
?>

```

Agarda element uchun kalit berilmagan bo'lsa, u holda kalit sifatida kalitning maksimal qiymatiga bir qo'shib hisoblanadi. Agarda qiymati mavjud kalit ko'rsatilgan bo'lsa, u holda shu kalit qiymatini ekranga chiqaradi. PHP 4.3.0 dasturlash tili versiyasidan boshlab kalitning maksimal qiymati manfiy son deb qaralsa, u holda massivning keyingi kaliti nol (0) bo'ladi.

Misol. PHP dasturlash tilida massivlar.

```
<?php
// $arr hamda $arr1 massivlar ekvivalentdir.

$arr = array(5 => 43, 32, 56, "b" => 12);

$arr1 = array(5 => 43, 6 => 32,
 7 => 56, "b" => 12);

?>
```

Agarda TRUE yoki FALSE kalit sifatida qo'llanilsa, u holda uning qiymati mos ravishda integer tipining bir va noliga o'zlashtiriladi. Agar NULL dan foydalanilsa, u holda kalit o'miga bo'sh satr hosil bo'ladi. Bu bo'sh satrni kalit sifatida foydalansa bo'ladi, ammo uni qo'shtirnoqqa olish kerak bo'ladi. Bu usul bo'sh kvadrat qavsni ishlatalish kabi emas. Massivlar yoki obyektlar kaliti sifatida foydalanish mumkin ham emas.

#### Kvadrat qavs sintaksisi yordamida aniqlash.

Massivga qiymat berish orqali massiv yaratish mumkin. Biz yuqorida aytib o'tganimizdek, massiv elementi qiymatiga ega bo'lish uchun kvadrat qavs ichiga uning kaliti ko'rsatilishi kerak, masalan, \$book["php"]. Agarda yangi kalit va yangi qiymat ko'rsatsangiz quyidagicha bo'ladi: \$book["new\_key"]="new\_value" hamda massivga yangi element qo'shiladi. Agarda kalitni ko'rsatmay faqat qiymatni o'zlashtirsak, ya'ni \$book[]="new\_value", u holda massivga yangi element qo'shiladi va uni kaliti mavjud maksimal qiymatga bir qo'shiladi. Agarda biz qiymat bergen massiv yaratilmagan bo'lsa, u holda biz qiymat bergandan keyin u yaratiladi.

```
<?

$books["key"] = value; // key kaliti bilan value qiymat

 // $books massiviga
 // qo'shiladi

$books[] = value1; /* 13-kalit bilan value1 qiymati

 massivga qo'shiladi, chunki
 bizda kalitning maksimal
 qiymati 12 edi. */

?>
```

Massivning aniq bir elementini o'zgartirish uchun uning shu kaliti bilan yangi qiymatga o'zlashtirish kerak. Massiv elementi kalitini o'zgartirish mumkin emas, faqat o'chirish (kalit va elementi juftligini) va yangi qo'shish mumkin xolos. Massiv elementini o'chirish uchun unset() funksiyasidan foydalanish kerak.

```

<?php
$books = array ("php" =>
 "PHP users guide",
 12 => true);

$books[] =
"Book about Perl"; // 13-kalit(indeks) bilan
 // yangi element qo'shildi,
 // bu quyidagiga ekvivalent
 // $books[13] =
 // "Book about Perl";

$books["lisp"] =
123456; /* Bu massivga yangi "lisp" kalitli
 123456 qiymali yangi
 element qo'shish*/
unset($books[12]); // Bu 12-kalitli elementni
 // massivdan o'chirish
unset ($books); // massivni butunlay o'chirish
?>

```

Bo'sh kvadrat qavsdan foydalanganda kalitning maksimal qiymati massivda mavjud oxirgi qayta indekslangan kalitlar orasidan qidiriladi. Massivni array\_values() funksiyasi yordamida qayta indekslash mumkin.

Misol. PHP dasturlash tilida massivlar.

```

<?php
$arr =
array ("a","b","c"); /* "a", "b" va "c"
 qiymatli massivni
 yaratamiz.
 bu yerda kalit ko'rsatilmagan
 biroq mos ravishda
 ular 0,1,2 bo'ladi. */
print_r($arr); // massivni ekranga chiqaramiz (kaliti va

```

```

// qiymatini)

unset($arr[0]);
unset($arr[1]);
unset($arr[2]);

// massivdan hamma elementini o'chiramiz

print_r($arr); // massivni ekranga chiqaramiz (kaliti va

// qiymatini)

$arr[] = "aa"; // massivga yangi element

// qo'shamiz.

// uni indeksi(kaliti)

// 3 bo'ladi, 0 emas.

print_r($arr);

$arr =
array_values($arr); // massivni

// qayta indekslaymiz.

$arr[] = "bb"; // bu elementni

// kaliti 1 bo'ladi.

print_r($arr);

?>

Bu skriptning natijasi quyidagicha bo'ladi:

Array ([0] => a [1] => b [2] => c)

Array ()

Array ([3] => aa)

Array ([0] => aa [1] => bb)

Object (obyektlar) tipi.

```

Obyektlar – obyektga yo'naltirilgan dasturlashdan kirib kelgan ma'lumot tipidir. Obyektga yo'naltirilgan dasturlash tamoyiliga ko'ra, sinf – aniq xossalarga ega va ular bilan ishlaydigan metodli obyektlar to'plami. Obyekt esa mos ravishda sinf nusxasidir. Masalan, dasturchilar – bu dasturni tuzuvchi, kompyuter adabiyotlarini o'rganadigan odamlar sinfi va bundan tashqari hamma odamlar qatori ism va familiyasi mavjud. Endi agarda biror aniq dasturchi – Azamat Boboyevni olsak, u holda uni shu xossaga ega bo'lgan dasturchi sinfini obyekti sifatida qarash mumkin va u ham dastur tuzadi, hamda ismi mavjud va boshqalar.

PHP dasturlash tilida obyekt metodiga murojaat -> amalidan foydalaniladi. Obyektni initsializatsiya qilishda obyektni o‘zgaruvchan nusxasini yaratadigan new ifodasidan foydalaniladi.

Misol. PHP dasturlash tilida obyektlar.

```
<?php
// odam sinfini yaratamiz.
class Person
{
 // PHP dasturlash tilini o‘rganadigan odam metodi
 function know_php()
 {
 echo "Endi men PHP dasturlash tilini bilaman!";
 }
}

$bob = new Person; // odam sinfini
 // obyektini yaratamiz.
$bob -> know_php(); // uni PHP tiliga o‘rgatamiz.
?>
```

Resource (resurslar) tipi.

Resurs – bu tashqi resursga (masalan, ma’lumotlar bazasi bilan bog‘lanish) ssilka orqali bog‘langan maxsus o‘zgaruvchidir. Resurslar maxsus funksiyalar (masalan, mysql\_connect(), pdf\_new() va shunga o‘xshashlar) yordamida yaratiladi va foydalaniladi.

Null tipi.

Maxsus NULL qiymati o‘zgaruvchini qiymatga ega emasligi haqida ogohlantiradi.

O‘zgaruvchi NULL qiymat qabul qiladi, agarda:

unga o‘zgarmas NULL (\$var = NULL) o‘zlashtirilgan bo‘lsa;

unga hech qanday qiymat berilmagan bo‘lsa;

u unset()funksiya yordamida tozalangan bo‘lsa.

NULL tipli faqat bitta qiymati mavjud – registrga sezgirligi kichik NULL kalit so‘zidir.

Masalaning yechilishi.

Endi bo‘limning boshida qo‘yilgan masalaga qaytsak. U turli sabablar bo‘yicha har xil odamlarga tuzilgan maktubni jo‘natishdan iborat edi. Bu masalani hal etish uchun o‘rganilgan vositalardan – o‘zgaruvchilar, amallar, o‘zgarmaslar, satrlar va massivlardan foydalanishga harakat qilamiz. Ko‘rsatilgan maktub qabul qiluvchiga bog‘liq ravishda murojaat va holati o‘zgaradi, shuning uchun tabiiy ravishda bu kattalikni o‘zgaruvchi deb belgilaymiz. Bundan tashqari hodisalar va odamlar ko‘p, shuning uchun massiv o‘zgaruvchi tipidan foydalanish qulay. Maktub matni har doim

o‘zgarmas, shuning uchun uni o‘zgarmas deb berish maqsadga muvofiqdir. Juda uzun va qo‘pol satrlarni yozmaslik uchun satrlar konkatenatsiya(qo‘shish) amalidan foydalanamiz. Shunday qilib, quyidagiga ega bo‘lamiz:

```
<?
// bizning yozuvimiz
// o‘zgarmas bo‘lsin.
define("SIGN","Hurmat bilan, Azamat");
// odamlar va hodisalar massivini beramiz
$names = array("Ivan Ivanovich",
 "Petr Petrovich",
 "Semen Semenovich");
$events = array(
 "f" => "ochiq eshiklar kuni",
 "o" => "ko‘rgazmaning ochilishi",
 "p" => "bitiruvchilar bali");
// taklifnomasi matnini tuzamiz.
$str = "Hurmatli, $names[0]";
$str .= "
 Sizni taklif etamiz ".
 $events["f"];
$str .= "
" . SIGN;
echo $str; // matnni ekranga chiqaramiz.
?>
```

Shunday qilib, bu bo‘limda biz PHP dasturlash tilining asosiy sintaksisi bilan tanishib chiqdik, turli tipdagi o‘zgaruvchilar, o‘garmaslar va amallar bilan ishlashni, PHP dasturlash tilidagi mavjud tiplarini o‘rgandik. Massivlar va satrlar ma’lumot tiplari haqida gap ketganda ularni chuqur va qismlarga ajratib o‘rgandik. Bu konstruksiyalar foydalanishga qulay va soddadir. Bular haqida keng ma’lumotlar keyingi bo‘limlarda keltirilgan. Masalaning yechilishi bor bilimlarga asoslangan holda sodda yechilgan, shuning uchun yechim amaliyotda qo‘llashga juda yaqin kelmaydi. Keyingi bo‘limlarda bu kamchiliklarni to‘g‘rilaymiz va elektron maktubni umumiylab shablonini yaratamiz.

### Masalaning qo‘yilishi

1. PHP da o‘zgaruvchilardan foydalanishni o‘rganish.
2. PHP da o‘zgaruvchilardan foydalanib, misol yechish.

### Variantlar

1. Bir o‘lchamli sonli massivni elementlari massivni eng katta elementini kvadratiga bo‘lib chiqilsin

2. Bir o`lchamli sonli massiv [a,b] kesmadagi elementlarini o`rtacha qiymatidan xisoblansin
3. Bir o`lchamli sonli massiv tok o'rindagi elementlarini yigindisi xisoblansin
4. Bir o`lchamli sonli massiv ni tok qiymatli elementlarini o`rtacha qiymatidan xisoblansin
5. Bir o`lchamli sonli massiv [a,b] kesmada yotmaydigan elementlari soni aniqlansin
6. Bir o`lchamli sonli massivni M dan keyingi elementlari yigindisini toping
7. Bir o`lchamli sonli massivni elementlarini kvadratlari yigindisi va o`rtacha qiymatidan xisoblansin
8. Bir o`lchamli sonli massivni tok kiymatli elementlarini juft o`rinli elementlari yigindisiga bo`lib chiqilsin
9. Bir o`lchamli sonli massiv max elementi bilan k chi elementi urnini almashtiring
10. Bir o`lchamli sonli massiv k - elementidan 1 - elementigacha bo`lgan elementlarining kublari yigindisi xisoblansin
11. Bir o`lchamli sonli massivni manfiy elementlari massivni o`rtacha qiymatidanni logarifmi bilan almashtirilsin
12. Bir o`lchamli sonli massiv K va M ga teng elementlari yigindisixisoblansin
13. Bir o`lchamli sonli massivni manfiy elementlari massivni eng kichik elementini kvadratiga almashtirilsin
14. Bir o`lchamli sonli massivni juft kiymatli elementlarini o`rtacha qiymatidan xisoblansin
15. Bir o`lchamli sonli massivni 2, 3 va 5 ga bo`linadigan elementlari yigindisi xisoblansin
16. A massivning eng katta (eng kichik) elementini ekranga chiqarish dasturini tuzing.
17. A massiv elementlari kiymatlarini kamayish tartibida saralash dasturini tuzing.
18. A massivda elementlar berilgan. Mazkur massiv elementlaridan shunday V massiv shakllantiruvchi shunday dastur tuzingki, V massiv elementlari kamayish tartibida saralangan bo'lsin.
19. Elementlari o'sish tartibida joylashgan A sonli massiv va a soni berilgan. a ni A massivga shunday qo'shingki, tartiblanganlik buzilmasin.
20. Elementlari o'sish tartibida joylashgan A massivni, elementlari kamayish tartibida joylashgan massiv ko'rinishida tez quruvchi dastur tuzing.
21. Manfiy va musbat sonlardan tashkil topgan A massiv berilgan. Barcha manfiy sonlarni chiqarib, musbatlarini o'sish tartibda joylashtiruvchi dastur tuzing.
22. Berilgan A massivdan ketma-ket sonlar olib, ulardan o'sish tartibida shakllantirilgan V massiv hosil qiluvchi dastur tuzing.
23.  $n$  o`lchamli massiv va  $k$  butun son berilgan ( $1 \leq k \leq n$ ). Massivning  $k$ -indeksli elementining oldiga yangi 0 kiymatli element joylashtirilsin.
24.  $n$  o`lchamli massiv va  $k$  butun son berilgan ( $1 \leq k \leq n$ ). Massivning  $k$ -indeksli elementidan keyin 0 kiymatli element joylashtirilsin.
25.  $n$  o`lchamli massiv berilgan. Massivning eng kichik elementidan oldin va eng katta elementidan keyin 0 kiymati joylashtirilsin(1-uchragan).
26.  $n$  o`lchamli massiv hamda 2 ta butun  $k$  va  $m$  ( $1 \leq k \leq n$ ,  $1 \leq m < k$ ) sonlari berilgan. Massivning  $k$ -indeksli elementidan oldingi  $m$  ta elementiga 0 kiymati o'rnatisin.
27.  $n$  o`lchamli massiv hamda 2 ta butun  $k$  va  $m$  ( $1 \leq k \leq n$ ,  $1 \leq m < n-k$ ) butun sonlari berilgan. Massivning  $k$ -indeksli elementidan keyingi  $m$  ta elementiga 0 kiymati o'rnatisin.
28.  $n$  o`lchamli massiv berilgan. Shatrli operatordan foydalanmasdan uning juft indeksdagi (2, 4, 6,...) elementlari ikkilantirilsin.
29.  $n$  o`lchamli massiv berilgan. Shatrli operatordan foydalanmasdan uning toq indeksdagi (3, 6, 9,...) elementlari uchlantirilsin.
30.  $n$  o`lchamli massiv berilgan. Massivning har bir musbat elementidan oldingi elementiga 0 kiymatlansin.

**8-Laboratoriya ishi**  
**Mavzu: PHP da ma'lumotlar bazasi tashkil etish**  
**PHPda ma'lumotlar bazasi bilan ishlash (CRUD) amallari.**

**Ishning maqsadi:** PHP da ma'lumotlar bazasi tashkil etish va u bilan ishlash.

**Nazariy qism**

Ushbu bo'lim PHP va MySql MBBS (SUBD) o'rtasidagi hamkorlik usullari bilan tanishishga mo'ljallangan. Asosiy e'tibor ma'lumotlar bazasi bilan bog'lanishni o'rnatish, so'rovlar jo'natish funksiyalari va javoblarni (mysql\_connect, mysql\_query, mysql\_result, mysql\_num\_rows, mysql\_close) qayta ishlashga qaratiladi. Misol sifatida virtual tarix muzeyi ma'lumotlar bazasining ma'muriyati uchun web-interfeys tuzish masalasini ko'raylik. PHP distributiviga MySQL ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan funksiyalarni oluvchi kengaytma kiradi. Bu bo'limda MySQL bilan ishlash uchun ba'zi bir ma'lumotlar bazasini tasvirlash va to'ldirish maqsadida web-interfeyslarni tuzish topshirig'ini yechish uchun kerak bo'ladigan asosiy funksiyalar bilan tanishamiz. Savol tug'iladi: bunday interfeyslarni tuzish nega kerak? SQL so'rovlar tili bilan notanish odamlar ma'lumotni ma'lumotlar bazasiga kiritish va uning tarkibini ko'rib turish imkoniyati bo'lishi uchun shunday qilinadi. Ma'lumotlar bazasiga ma'lumotlarni qo'shish uchun web-interfeys bilan ishlashda bu ma'lumotlarni shunchaki html-formaga kiritish va ularni serverga jo'natish kerak bo'ladi, bizning skript esa qolgan barcha amallarni bajaradi. Jadval tarkibini ko'rib turish uchun havola ustiga bir marta bosish va kerakli sahifaga kirish kifoya.

Ko'rinishi turishi uchun bu interfeyslarni virtual muzey eksponatlari haqidagi ma'lumotlar joylanadigan Artifacts jadvallari uchun tuzamiz. Avvalgi bo'limda bu kolleksiya strukturani hamda uning shaxs (Persons) va tasvirlar (Images) tavsiflari kolleksiyalari bilan aloqasini kiritgan edik. Artifacts kolleksiyasidagi har bir eksponat quyidagi xarakteristika yordamida tasvirlanishini eslatib o'tamiz:

nom (title);  
muallif (author);  
ifoda (description);  
o'rindosh nom (alternative);  
tasvir (photo).

Nomlanish va o'rindosh nomlanish uzunasiga 255 belgidan kam satr (ya'ni VARCHAR(255)), tasvirlash – matnli maydon (TEXT turiga mansub) hisoblanadi, "muallif" va "tasvir" maydonlarida esa Persons kolleksiyasidan muallifning identifikatorlari va Images kolleksiyasidan eksponat tasvirlariga muvofiq mavjud bo'ladi.

**Ma'lumotni qo'shish uchun interfeys tuzish**

Demak, bizda ma'lumotlar bazasida biron-bir jadval bor. Bu jadvalga ma'lumotni qo'shishga mo'ljallangan interfeys tuzish uchun uning strukturasini (ya'ni uning maydonlari jamlanmasini) html-formada tasvirlash kerak bo'ladi.

Bu topshiriqni quyidagi tarkibiy topshiriqlarga bo'lib chiqamiz:

MB bilan ulanishni o'rnatish;

MB ishini tanlash;

Jadval maydonlari ro‘yxatini olish;

html-formada maydonlarni tasvirlash.

Bundan keyin formaga kiritilgan ma’lumotlarni ma’lumotlar bazasiga kiritish kerak bo‘ladi.

Bu topshiriqlarning barchasini tartib bilan ko‘rib chiqamiz.

Aloqa o‘rnatish

Demak, birinchi galdegisi vazifa – bu ma’lumotlar bazasi bilan aloqa o‘rnatish. mysql\_connect funksiyasidan foydalanamiz. mysql\_connect sintaksi

mysql\_connect resursi ([server qatori[], username qatori[], password qatori[], mantiqiy new\_link[], butun client\_flags]]])

Bu funksiya MySQL serveri bilan aloqa o‘rnatadi va bu aloqaga ko‘rsatkich qaytaradi yoki muvaffaqiyatsiz chiqqanda FALSE ko‘rsatadi. Yetishmayotgan parametrlar uchun quyidagi ma’nolar muvaqqat qo‘yiladi:

server = ‘localhost:3306’

username = server jarayoni egasidan foydalanuvchi ismi

password = bo‘sh parol

Agar funksiya aynan bir xil parametrlar bilan ikki marta chaqiriladigan bo‘lsa, yangi ulanish o‘rnatilmaydi, aksincha havola eski aloqaga qaytariladi. Bundan qochish uchun esa, har qanday holatda yangi bir aloqa ochishga majbur qiluvchi new\_link parametridan foydalaniladi.

client\_flags parametri – bu quyidagi konstantalar kombinatsiyasi:

MYSQL\_CLIENT\_COMPRESS (siqish protokolidan foydalaniladi),

MYSQL\_CLIENT\_IGNORE\_SPACE

(funksiya nomidan so‘ng probel qo‘yishga ruxsat beradi),

MYSQL\_CLIENT\_INTERACTIVE (interactive\_timeout bir soniya kutish - wait\_timeout – ulanish tugatilishigacha). new\_link parametri PHP 4.2.0 da mavjud, client\_flags esa – PHP 4.3.0 da.

Server bilan ulanish, agar u bungacha mysql\_close() yordamida yopilmagan bo‘lsa, skriptni amalga oshirish tugallanishida berkiladi.

Shunday qilib, “123” parolli nina foydalanuvchisi uchun lokal serverda ma’lumotlar bazasi bilan ulanishni o‘rnatamiz.

Ma’lumotlar bazalarini tanlash

Ulanishni amalga oshirgandan so‘ng birga ishlashimiz kerak bo‘lgan ma’lumotlar bazasini tanlashimiz kerak bo‘ladi. Bizning ma’lumotlarimiz book ma’lumotlar bazasida saqlanadi.

MySQL da ma’lumotlar bazasini tanlash use buyrug‘i yordamida amalga oshiriladi:

mysql>use book;

PHP da buning uchun mysql\_select\_db funksiyasi mavjud

`mysql_select_db`: sintaksisi

```
mantiqiy mysql_select_db (
 database_name qatori
 [,link_identifier resursi])
```

Bu funksiya TRUE qiymatni ma'lumotlar bazasini muvaffaqiyatli tanlanganda qaytaradi va FALSE ni esa – aksincha bo'lganda.

Jadval maydonlari ro'yxatini olish

Endi topshiriqni yechish bilan alohida shug'ullansa bo'ladi. Jadval maydonlari ro'yxatini qanday olish mumkin? Juda oddiy. PHP da bu holda ham o'ziga tegishli buyruq mavjud – `mysql_list_fields`.

`mysql_list_fields` sintaksisi

```
mysql_list_fields (resursi
 database_name qatori,
 table_name qatori
 [, resurs link_identifier])
```

Bu funksiya maydonlar ro'yxatini `database_name` ma'lumotlar bazasidagi `table_name` jadvaliga qaytaradi. Kelib chiqadiki, ma'lumotlar bazasini tanlash bizga majburiy emas, lekin keyinroq asqotadi. Bu funksiya ishining natijasini – resurs tipi o'zgaruvchisini qanday baholash mumkin. YA'ni bu biz olishni xohlagan narsa emas. Bu ularning nomlari, tiplari va bayroqlari kiradigan jadval maydonlari haqidagi ma'lumotlarni olish uchun foydalanish mumkin bo'lgan havola. `mysql_field_name` funksiyasi so'rov amalga oshirilishi natijasida olingan maydon nomini qaytaradi. `mysql_field_len` funksiyasi maydon uzunligini qaytaradi. `mysql_field_type` funksiyasi maydon tipini qaytaradi, `mysql_field_flags` funksiyasi esa probel bilan yozilgan maydon bayroqlari ro'yxatini qaytaradi. Maydon tiplari int, real, string, blob va b. bo'lishi mumkin. Bayroqlar not\_null, primary\_key, unique\_key, blob, auto\_increment va b. bo'lishi mumkin.

Bu barcha buyruqlar sintaksisi bir xil:

```
mysql_field_name (result qatori, butun field_offset) resursi;
mysql_field_type (result qatori, butun field_offset) resursi;
mysql_field_flags (result qatori, butun field_offset) resursi;
mysql_field_len (result qatori, butun field_offset)
```

Bu yerda `result` – bu so'rov natijasi identifikatori (masalan, `mysql_list_fields` yoki `mysql_query` funksiyalar bilan jo'natilgan so'rov(bu haqda keyinroq gapirib o'tiladi)), `field_offset` esa – natijadagi maydonning tartib raqami. Umuman olganda, `mysql_list_fields` yoki `mysql_query` turidagi funksiyalarni qaytaruvchi jadvalni, aniqrog'i uning unga bo'lgan ko'rsatkichini aks ettiradi. Bu jadvallardan aniq ma'nolar olish uchun maxsus bu jadvalni ostki qator sifatida o'quvchi funksiyalarni ishga tushirish kerak. Bu funksiyalarga `mysql_field_name` va shu kabilar ham kiradi. So'rovni amalga oshirish natijasi jadvalidagi barcha qatorlarni saralash uchun bu jadvaldagi qatorlar miqdorini bilish kerak.

`mysql_num_rows(result resursi)` buyrug‘i result ning ko‘pgina natijalari qatorlari miqdorini qaytaradi.

html-formada maydonlar ro‘yxatining aks etishi

Endi biroz yuqoridagi misolni tekshirib ko‘ramiz. Maydon haqidagi ma’lumotni shunchaki chiqarmaymiz, uni munosib html-forma elementida aks ettiramiz. BLOB turidagi elementlarni textarea ga o‘tkazamiz (TEXT turida biz tuzgan description maydoni BLOB turiga ega ekanligini e’tibordan qochirmaymiz), raqamlar va qatorlarni `<input type=text>` kirish qismi matnli qatorlarida aks ettiramiz, avtoinkrement belgisiga ega elementni esa umuman aks ettirmaymiz, chunki uning ma’nosni o‘z-o‘zidan o‘rnataladi.

Bularning barchasi anchayin oddiy hal qilinadi, bayroqlar ro‘yxatidaig `auto_increment` bayrog‘i bundan mustasno. Buning uchun explode funksiyasidan foydalaniladi:

`explode`: sintaksisi

`explode` massivi (separator qatori,

string qatori [, int limit])

Bu funksiya string qatorini separator taqsimlagichi yordamida qismlarga bo‘ladi va olingan qatorlar massivini qaytaradi.

Bizning holatda taqsimlagich sifatida probel “ ” ni olish kera, bo‘lish uchun boshlang‘ich qator sifatida esa – maydon bayroqlari qatorini.

Ma’lumotlar bazasiga ma’lumotlar yozish

Shunday qilib, forma tuzildi. Endi eng asosiysini bajarish qoldi – bu formadagi ma’lumotlarni bizning ma’lumotlar bazasiga jo‘natish. Ma’lumki, ma’lumotlarni jadvalga yozish uchun SQL tilidagi `INSERT` buyrug‘i ishlatiladi. Masalan:

`mysql> INSERT INTO Artifacts`

`SET title='Kamolov';`

PHP skriptda bunday buyruqdan (yoki SQL dagi istalgan buyruqdan) qanday foydalaniladi, degan savol tug‘iladi. Buning uchun `mysql_query()` funksiysi mavjud.

`mysql_query` sintaksisi

`mysql_query` resursi (query qatori

[, resurs link\_identifier])

`mysql_query()` SQL-so‘rovni MySQL ma’lumotlar bazasining `link_identifier` ko‘rsatkichi yordamida aniqlanadigan aktiv ma’lumotlar bazasiga jo‘natadi (bu MySQL serveri bilan biron-bir aloqaga havola). Agar `link_identifier` parametri o‘tkazib yuborilgan bo‘lsa, so‘nggi ochiq aloqa ishlatiladi. Agar ochiq aloqa bo‘lmasa, funksiya parametrsiz `mysql_connect()` funksiyasiga o‘xshash holda MBBT (SUBD) bilan bog‘lanishga urinadi.

So‘rov natijasi buferlanadi.

Misol yechish namunalari

Mavjud Artifacts nomli jadval bilan ishlang.

Artifacts jadvali maydonlari ro'yxatini olishni ko'ramiz.

<?

```
$conn = mysql_connect("localhost","nina","123") or die("Aloqa o'rnatib bo'lmaydi: ".
mysql_error());
echo "Aloqa o'rnatildi";
mysql_select_db("book");
$list_f = mysql_list_fields ("book","Artifacts",$conn);
$n = mysql_num_fields($list_f);
for($i=0;$i<$n; $i++){
 $type = mysql_field_type($list_f, $i);
 $name_f = mysql_field_name($list_f,$i);
 $len = mysql_field_len($list_f, $i);
 $flags_str = mysql_field_flags ($list_f, $i);
 echo "
Maydon nomi: ". $name_f;
 echo "
Maydon turi: ". $type;
 echo "
Maydon uzunligi: ". $len;
 echo "
Maydon bayroqlari qatori: ".
 $flags_str . "<hr>";
}
?>
```

Ma'lumotlarni Artifacts jadvaliga kiritish uchun forma tuzamiz:

<?

```
$conn=mysql_connect("localhost","nina","123");
// aloqa o'rnatamiz
$database = "book";
$table_name = "Artifacts";
mysql_select_db($database); // ish uchun ma'lumotlar bazasi tanlaymiz
$list_f = mysql_list_fields($database,$table_name);
// bazadan maydonlar ro'yxatini olamiz
$n = mysql_num_fields($list_f); /*avvalgi so'rovdag'i natijadagi qatorlar soni (Artifacts
jadvalida ja'mi qancha maydonlar bor) */
echo "<form method=post action=insert.php>";
```

```

// ma'lumotlarni kiritish uchun forma tuzamiz

echo " <TABLE BORDER=0 CELLSPACING=0 width=50% ><tr>

<TD BGCOLOR=#005533' align=center>
 Add new row in $table_name</td></tr><tr><td> </td></td></tr></TABLE>";

echo "<table border=0 CELLSPACING=1 cellpadding=0 width=50% >";

// har bir maydon uchun uning nomi, turi, uzunligi va bayrog'iini olamiz

for($i=0;$i<$n; $i++)

{

$type = mysql_field_type($list_f, $i);

$name_f = mysql_field_name ($list_f,$i);

$len = mysql_field_len($list_f, $i);

$flags_str = mysql_field_flags ($list_f, $i);

// bayroqlar qatoridan massiv qilamiz,

// unda massivning har bir elementi – maydon bayrog'i

$flags = explode(" ", $flags_str);

foreach ($flags as $f){

if ($f =='auto_increment') $key = $name_f;

// avtoinkrement nomini eslatib o'tamiz

}

/* avtoinkrement hisoblangan har bir maydon uchun, uning turidan kelib chiqib mos keluvchi
forma elementini chiqaramiz*/

if ($key <> $name_f){

echo "<tr><td align=right bgcolor=#C2E3B6'>

 ". $name_f ."</td>";

switch ($type){

case "string":

$w = $len/5;

echo "<td><input type=text name=\"$name_f\""

size = $w ></td>";

break;

case "int":

$w = $len/4;

```

```
echo "<td><input type=text name=\"$name_f\"
size = $w ></td>";

break;

case "blob":

echo "<td><textarea rows=6 cols=60 name=\"$name_f\"></textarea> </td>";

break;

}

}

echo "</tr>";

}

echo "</table>";

echo "<input type=submit name='add' value='Add'>";

echo "</form>";

?>
```

## Topshiriqlar

Talabalar o‘z laboratoriya topshiriqlari bo‘yicha ma’lumotlari bazasini tuzadilar.

## **9-Laboratoriya ishi**

**Mavzu: PHPda AJAX texnologiyani qo'llash. PHPda avtorizasiya va autentifikasiyani amalga oshirish usullari.**

**Ishdan maqsad: PHP da sinflar yaratishni va Ob'ekt tushunchasini o'rghanish.**

### **Nazariy qism**

#### **Sinflar va ob'ektlar.**

Ob'ektga yo'naltirilgan dasturlashning asosiy tushunchalari – sinflar hamada ob'ektlardir. Bu tushunchalarni quyidagicha tushunish mumkin: ob'ekt – bu dasturda qo'llaniladigan tushuncha yoki biror fizik predmet haqida ma'lumot beradigan strukturalangan o'zgaruvchidir, sinflar esa bu ob'ektlarning tavsifi va ular ustida bajariladigan harakatlardir.

PHP dasturlash tilida sinflar quyidagi sintaksis yordamida aniqlanadi:

```
class Sinf_nomi{
 var $xususiyat_nomi;
 /*xususiyatlar ro'yxati*/
 function metod_nomi(){
 /* usullarning tanasi */
 }
 /*usullar ro'yxati*/
}
```

Sinf ob'ektlari xususiyatlari nomi var kalit so'zi yordamida e'lon qilinadi, berilgan sind ob'ektlariga qo'llanilgan usullar funksiya sifatida ishlatiladi. Sinf tanasi ichida this kalit so'zi yordamida taqdim qilinayotgan joriy sinfga murojaatni amalga oshirish mumkin.

Masalan, biz maqola kategoriyasini tasvirlovchi sind tuzishimiz kerak. Har bir maqolaning nomi, muallifi va qisqa mazmuni kabi xususiyatlari bor. Biz maqola bilan qanday amal bajarmoqchimiz? Biz sanab o'tilgan xususiyatlarga ma'no berishimiz, maqolani brauzerda ko'rsatishimiz kerak bo'ladi. Shunda bu sindning ifodalanishi quyidagicha holatda bo'ladi:

```
<?
class Articles { // Maqola sinfini tuzamiz
 var $title;
 var $author;
 var $description;
 // maqola atributi
 // ma'nosini o'zlashtiruvchi usul
 function make_article($t, $a, $d){
```

```

$this->title = $t;
$this->author = $a;
$this->description = $d;
}

//sinf nusxasini ifodalash uchun usul

function show_article(){
$art = $this->title . "
" .
$this->description .

"
Muallif: " . $this->author;
echo $art;
}

}

?>
- 68 -

```

Shunday qilib “maqola” turidagi fizik ob’ektlarni tasvirlash uchun biz uch o’zgartuvchidan tashkil topgan, maqola xarakteristikasini o’zida jamlagan Articles nomli sinf va muayyan maqola tuzish va uni tasvirlash uchun ikkita funktsiya tuzdik.

Ma’lumki, PHP bilan ishlash davriy holatda HTML rejimida yuklanishi mumkin. Bu holda dastur bir necha kodning bo’laklari(bloklar)dan tashkil topadi. Sinfni ifodalash php-kodning har xil bloklari bo’yicha va qolaversa har xil fayllar bo’yicha tarqatilmasligi kerak. Ya’ni quyidagicha yozsak:

```

<?php
class Articles { // Sinfni tasvirlashning boshi
 var $title;
?>
<?php
// sinfni tasvirlashning davomi
function show_article(){
 // usulning tarkibi
}
} // sinfni tasvirlashning yakuni
?>
bunda dastur tartibli ishlaydi.

```

Sinf nomi masalasida ayrim narsalarni e'tiborda tutish kerak. Sinfning nomi PHP tilidagi ob'ektlar nomlanishi qoidalariga javob berishi lozim, lekin bir qator nomlar borki, texnik mutaxassislar tomonidan o'z maqsadi uchun zaxira qilinadi. Birinchi navbatda bu nomlar “\_” quyilchiziqdan boshlanuvchilardir. Sinflar va funktsiyalar tuzish uchun bu kabi nomlarni ishlatmaslik kerak. Bundan tashqari stdClass nomi zaxira qilingan, zero u PHP surilgichi ichida ishlatiladi.

## O'zgaruvchilarni initsiallashtirish

Ba'zan ayrim sinf atributlariga ma'noni sinf ishtirokchisini tuzish bilanoq o'zlashtirish kerak bo'ladi. Biz maqola sinfini tuzganimizda, sinf atributlari (xususiyatlari) ma'nolarini o'zlashtirish uchun maxsus funktsiya make\_article() dan foydalandik. Umuman olganda, biz to'g'ri yo'l tutmadik, chunki “velosiped ixtirosi” bilan shug'ullanmadik. Sinf atributlarining boshlang'ich ma'nolarini berish uchun maxsus ikkita standart usul mavjud. PHP4da ma'noni var operatori yoki konstruktor funktsiyasi yordamida initsiallashtirish mumkin. var yordamida faqat konstant ma'nolarni initsiallashtirish mumkin. Konstant bo'lмаган ma'nolarni berish uchun ob'ekt sinfdan ajrab chiqqanda o'z-o'zidan ishga tushuvchi konstruktor funtsiyasidan foydalanganiladi. Konstruktor-funktsiya u ifodalangan butun sinfga mos keluvchi nomga ega bo'lishi kerak.

Misol keltiramiz. Aytaylik, “maqola” ob'ektini tuzishda biz uning xususiyatlarini quyidagicha belgilaylik: mualliflar – “Kamolov” qatoriga teng, nomlanish va qisqa mazmun - \$\_POST global massivi elementlariga mos, maqola nashri – mazkur sanada. Unda sinfning quyidagi ifodasi PHP4da batartib bo'lmaydi:

```
<?
class Articles { // maqola sinfini tuzamiz
 var $title;
 var $author = "Kamolov";
 var $description;
 var $published;
 // sinf atributi
 // ma'nosini o'zlashtiruvchi usul
 - 69 -
}
```

Sinfning quyidagi ifodasi esa PHP4da keraklicha yo'sinda ishlaydi:

```
<?
class Articles { // maqola sinfini tuzish
 var $title;
```

```

var $author = "Kamolov";
var $description;
var $published;
// sinf atributi
// ma'nosini o'zlashtiruvchi usul
function Articles(){
 $this->title = $_POST["title"];
 $this->description = $_POST["description"];
 $this ->published = date("Y-m-d");
}
?>

```

PHP3 va PHP4 da konstruktordar har xil ishlashini hisobga olish kerak. Funktsiya PHP3 da, agar u sinfniki kabi nomga ega bo'lsa, konsrtuktorga aylanadi, PHP4 da esa – agar u ifodalangan sinfniki kabi nomga ega bo'lsa shunday bo'ladi. Bir sinf boshqasini kengaytirganda va xususiyatlarning va baza sinflar usullarining ergashishida usullar orasidagi farq ko'rinish turibdi. Lekin biz bu haqda biroz keyinroq gapiramiz. PHP5da sinf konstruktori \_construct deb nomlanadi. Bundan tashqari, PHPda destruktorlar – ob'ektini yo'q qilishda o'z-o'zidan ishga tushuvchi funksiyalar paydo bo'ldi. PHP5 da funktsiya-destruktor destruct deb nomlanishi kerak bo'ladi.

Birinchi ma'ruzalardan birida biz PHP da ob'ekt deb ataluvchi tip mavjudligi haqida aytib o'tgandik. Sinf – bu ob'ekt tipidagi ma'lumotlarning bir turidagi ifodalanishidir. Sinflar real o'zgaruvchilar uchun shablon vazifasini o'taydi. Kerakli tipdagi o'zgartuvchi new operatori yordamida sinfdan tuziladi. Ob'ektini tuzib, biz barcha usullarni qo'llashimiz va barcha sinf ifodasida ko'rsatib o'tilgan xususiyatlarni olishimiz mumkin bo'ladi. Buning uchun quyidagicha sintaksidan foydalilanadi: \$ob'ekt\_nomi->xususiyat yoki \_nomi \$ob'ekt\_nomi->usulning\_nomlanishi(argumentlar ro'yxati). Xususiyatlar yoki usular nomlari oldidan \$ belgisi qo'yilmaydi.

```

<?php
$art = new Articles;
// ob'ekt tuzamiz $art
echo ($art ->title);
// ob'ekta nomlanish beramiz $art
$another_art = new Articles;
// ob'ekt tuzamiz $another_art
$another_art->show_article();
// ob'ektning brauzerdagi ifodasi uchun

```

```
// usulni chaqiramiz
```

```
?>
```

Misol: Ob'ekt usullari va xususiyatlariga erkin kirish (dostup)

Sinfning har bir ob'ekti aynan bir xil xususiyatlar va usullarga ega bo'ladi. Demak, \$art ob'ektda va \$another\_art ob'ektda title, description, author xususiyatlari va Articles(),  
- 70 -

show\_article() usullari mavjud. Lekin bular ikki xil ob'ektlar. Ob'ektni fayllar sistemasidagi direktoriya deb hisoblaymiz, uning xarakteristikasi esa – bu direktoriyadagi fayllar singari bo'lsin. Aniqki, har bir direktoriyada bir xil fayllar yotishi mumkin, lekin shunday bo'lsa-da, ular har xil direktoriyalarda saqlanayotgani uchun har xil hisoblanishi mumkin. Xuddi shuningdek, xususiyatlar va usullar ham, agar ular turli ob'ektlarga qo'llaniladigan bo'lsa, har xil hisoblanadi. Yuqori bosqichdagi direktoriyadan kerakli faylni olish uchun bu faylga yo'lni batafsil yozib chiqamiz. Sinflar bilan ishlash mobaynida biz chaqirishni istagan funktsiyaning nomini to'liq yozishimiz kerak bo'ladi. PHP dagi yuqori bosqich direktoriyalariga global o'zgaruvchilarning bo'sh o'rni bo'ladi, yo'l esa -> taqsimlovchisi yordamida ko'rsatiladi. Shu tarzda \$art->title va \$another\_art->title nomlari ikki xil turli o'zgaruvchilarni anglatadi. PHP da o'zgaruvchi nom oldidan faqat bitta dollar belgisiga ega bo'ladi, shuning uchun \$art->\$title ko'rinishida yozish mumkin emas. Bu konstruktsiya \$art ob'ekting title xususiyatiga murojaat sifatida ko'rib chiqilmaydi, \$title o'zgartuvchi ko'rinishida berilgan nomli xususiyat sifatida ko'riladi (masalan, \$art->"").

```
<?php
```

```
$art->title = " Internet ga kirish";
```

```
 // ob'ekt xususiyati ma'nosini
```

```
 // shunday o'rnatish mumkin
```

```
$art->$title = "Internet ga kirish";
```

```
 // ob'ekt xususiyati ma'nosini
```

```
 // bunday o'rnatib bo'lmaydi
```

```
$property = "title";
```

```
$art->$property = "Internet ga kirish";
```

```
 // ob'ekt xususiyati ma'nosini
```

```
 // shunday o'rnatish mumkin
```

```
?>
```

Misol: Xususiyatlar ma'nosini o'rnatish

Sinfni tuzib, bu sinfning ob'ekti qanday nomga ega bo'lishini bila olmaymiz, qolaversa ob'ektlar juda ko'p bo'lishi va ularning barchasi har xil nomga ega bo'lishi mumkin. Sinfni yuzaga chiqarish ichida ob'ektga qanday munosabatda bo'lishni bilmaymiz. Sinf yuzaga chiqishi ichida

funktsiyalar va o'zgaruvchilarga erkin kirish uchun, \$this o'rindosh o'zgaruvchisidan foydalanish kerak. Masalan, \$this->title shunday sinf ob'ektining titleini qaytaradi. Ba'zan bu o'zgaruvchini "mening xususiy mulkim" (xususiyatga munosabat tariqasida) deb o'qishni taklif qilinadi.

Topshiriqlar:

Helper.php

```
<?php
```

```
class Helper {
```

```
 static function add($pdo,$ism="", $fam="", $ota=<">, $tel=<">, $manzil=<">, $guruh=<">){
```

```
 $pdo->prepare("INSERT INTO talabalar(ism,familiya,sharif,tel,manzil,guruh_id)
```

```
 VALUES(?,?,?,?,?,?)")
```

```
 ->execute(array($ism,$fam,$ota,$tel,$manzil,$guruh));
```

```
}
```

```
 static function update($pdo,$id,$ism="", $fam="", $ota="", $tel="", $manzil="", $guruh=""){
```

```
 $pdo->prepare("UPDATE talabalar SET
```

```
 ism=?, familya=?, sharif=?, tel=?, manzil=?, guruh_id=?
```

```
 WHERE idq?")
```

```
 ->execute(array($ism,$fam,$ota,$tel,$manzil,$guruh,$id));
```

- 71 -

```
}
```

```
 static function delete($pdo,$id){
```

```
 $pdo->exec("DELETE FROM talabalar WHERE idq$id");
```

```
}
```

```
 static function getTalabalar($pdo){
```

```
 return $pdo->query("SELECT s.*,g.nomi AS guruh_nomi
```

```
 FROM talabalar s
```

```
 JOIN guruh g ON s.guruh_id=q.id")->fetchAll(PDO::FETCH_ASSOC);
```

```
}
```

```
static function getTalaba($pdo,$id){
 return $pdo->query("SELECT * FROM talabalar WHERE idq$id")->fetch();
}
```

```
static function getGuruhs($pdo){
 return $pdo->query("SELECT * FROM guruhs")
 ->fetchAll(); }
```

Index.php

```
<?php
include "Helper.php";
$dsn = "mysql:host=localhost;dbname=tatu";
$conn = new PDO($dsn,'root','');
$students = Helper::getTalabalar($conn);
?>
<!doctype html>
<html>
<head>

</head>
<body>
Qo'shish
<table border="1">
<tr>
<th>ID</th>
<th>Familiya</th>
<th>Ism</th>
<th>Otasining ismi</th>
<th>Manzili</th>
<th>Tel nomer</th>
<th>Guruhs</th>
```

```

<th>Amallar</th>
</tr>
<?php foreach($students as $student):?>
<tr>
<td><?= $student['id'];?></td>
<td><?= $student['familya'];?></td>
<td><?= $student['ism'];?></td>
<td><?= $student['sharif'];?></td>
<td><?= $student['manzil'];?></td>
<td><?= $student['tel'];?></td>
<td><?= $student['guruh_nomi'];?></td>
<td>
<a href="update.php?idq<?= $student['id']?>">
<button>Tahrirlash</button>

<a href="delete.php?idq<?= $student['id']?>">
<button>O'chirish</button>

</td>
</tr>
<?php endforeach;?>
</table>
</body>
</html>
Update.php
<?php
include "Helper.php";
$dsn = "mysql:host=localhost;dbname=tatu";
$conn = new PDO($dsn,'root','');
$groups=Helper::getGuruhs($conn);
$id = intval($_GET['id']);
$student = Helper::getTalaba($conn,$id);

```

```

if($_POST){

 $ism = $_POST['ism'];
 $fam = $_POST['fam'];
 $ota = $_POST['ota_ism'];
 $manzil =$_POST['manzil'];
 $tel = $_POST['tel'];
 $guruh = $_POST['guruh'];

 Helper::update($conn,$id,$ism,$fam,$ota,$tel,$manzil,$guruh);
 header("Location: index.php");
}

?>

<html>
<head>
</head>
<body>
<form method="post">

<table>
 <tr>
 <td>Ism</td>
 <td><input name="ism" type="text" value=<?=$student['ism']?>"></td>
 </tr>
 <tr>
 <td>Familiya</td>
 <td><input name="fam" type="text" value=<?=$student['familya']?>"></td>
 </tr>
 <tr> <td>Otasing ismi</td>
 <td><input name="ota_ism" type="text" value=<?=$student['sharif']?>"></td>
 </tr>
 <tr>
 <td>Manzil</td>
 <td><textarea name="manzil"> <?=$student['manzil']?> </textarea></td>
 </tr>

```

```

<tr>
 <td>Tel nomer</td>
 <td><input name="tel" type="tel" value=<?=$student['tel']?>></td>
</tr>

<tr>
 <td>Guruh</td>
 <td> <select name="guruh">
 <?php foreach($groups as $g):?>
 <option value=<?=$g['id'];?>> <?php if($g['id']==$student['guruh_id']) echo
"selected";?>><?=$g['nomi'];?></option>
 <?php endforeach;?>
 </select>
 </td>
</tr>

<tr>
 <td colspan="2">
 <input type="submit" value="Saqlash" name="sbmt">
 </td>
</tr>
</table>
</form>
</body>
</html>

```

### **Topshiriqlar**

Talabalar o‘z laboratoriya topshiriqlari bo‘yicha ma’lumotlari bazasidagi ma’lumotlardan obyektga yo’naltirilgan dasturlashni qo’llab, sahifa ma’lumotlarini hosil qiladi.

## 10-Laboratoriya ishi

**Mavzu: YII2 freymvorkda veb ilovalarni ishlab chiqish. “Model, View, Controller” ni boshqarish**

**Ishning maqsadi: YII2 freymvorkda veb ilovalarni ishlab chiqishni o’rganish. “Model, View, Controller” ni boshqarishni o’rganish**

*Nazariy qism*

Ushbu bo’limda Gii- dan ba’zi bir keng tarqalgan veb-sayt xususiyatlarini amalga oshiradigan kodni avtomatik ravishda yaratish uchun qanday foydalanish tasvirlangan . Kodni avtomatik ravishda yaratish uchun Gii-dan foydalanish shunchaki Gii veb-sahifalarida ko’rsatilgan ko’rsatmalarga to’g’ri ma'lumotlarni kiritish bilan bog’liq.

Ushbu o’quv qo’llanma orqali siz quyidagilarni bilib olasiz:

arizangizda Gii-ni yoqish

Active Record sinfini yaratish uchun Gii dan foydalanish,

MB jadvali uchun CRUD operatsiyalarini amalga oshiradigan kodni yaratish uchun Gii dan foydalanish,

Gii tomonidan yaratilgan kodni sozlash.

Gii Yii da modul sifatida taqdim etilgan . Siz Gii- ni dasturning modullari xususiyatida sozlash orqali yoqishingiz mumkin . Arizangizni qanday yaratganingizga qarab, config/web.php konfiguratsiya faylida quyidagi kod mavjudligini topishingiz mumkin :

```
$config = [...]; if (YII_ENV_DEV) {
 $config['bootstrap'][] = 'gii';
 $config['modules']['gii'] = ['class' => 'yii\gii\Module',];
}
```

Ushbu qator tufayli sizning ilovaingiz ishlab chiqish rejimida va yuqoridagi konfiguratsiya bo'yicha Gii-ni allaqachon yoqib qo'yan bo'ladi. Endi Gii-ga quyidagi URL orqali kirishingiz mumkin:

<http://hostname/index.php?r=gii>

# Welcome to Gii

 a magical tool that can write code for you

Start the fun with the following code generators:

## Model Generator

This generator generates an ActiveRecord class for the specified database table.

[Start »](#)

## CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

[Start »](#)

## Controller Generator

This generator helps you to quickly generate a new controller class, one or several controller actions and their corresponding views.

[Start »](#)

## Form Generator

This generator generates a view script file that displays a form to collect input for the specified model class.

[Start »](#)

## Module Generator

This generator helps you to generate the skeleton code needed by a Yii module.

[Start »](#)

## Extension Generator

This generator helps you to generate the files needed by a Yii extension.

[Start »](#)[Get More Generators](#)

A Product of [Yii Software LLC](#)

Powered by [Yii Framework](#) 

Faol yozuvlar sinfini yaratish

Gii-dan Active Record sinfini yaratish uchun foydalanish uchun "Model Generator" -ni tanlang (Gii indeks sahifasidagi havolani bosish orqali). Keyin shaklni quyidagicha to'ldiring:

Jadval nomi: country

Model sinfi: Country

**Model Generator**

This generator generates an ActiveRecord class for the specified database table.

**Table Name**: country

**Model Class**: Country

**Namespace**: app\models

**Base Class**: yii\db\ActiveRecord

**Database Connection ID**: db

**Use Table Prefix**

**Generate Relations**: All relations

**Generate Labels from DB Comments**

**Generate ActiveQuery**

**Enable I18N**

**Use Schema Name**

**Code Template**: default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/model/default)

**Preview** **Generate**

Click on the above **Generate** button to generate the files selected below:

Code File	Action
models\Country.php	create

**Create**  **Unchanged**  **Overwrite**

Keyin, "Ko'rib chiqish" tugmasini bosing. models/Country.php Natijada hosil bo'ladigan sinf faylida ko'rsatilganligini ko'rasiz . Uning tarkibini oldindan ko'rish uchun sinf fayli nomini bosishingiz mumkin.

Gii-dan foydalanganda, agar siz o'sha faylni allaqachon yaratgan bo'lsangiz va uning ustiga yozishni diffistasangiz, yaratiladigan kod va mavjud versiya o'rtasidagi farqlarni ko'rish uchun fayl nomi yonidagi tugmani bosing.

The screenshot shows the Yii Gii Model Generator interface. On the left, a sidebar lists generators: Model Generator, CRUD Generator, Controller Generator, Form Generator, Module Generator, and Extension Generator. The main area displays the generated code for `Country.php`:

```

9 9 *
10 10 * @property integer $id
11 11 * @property string $name
12 12 * @property string $code
13 13 */
13 14 class Country extends \yii\db\ActiveRecord
14 15 {
15 16 ...
16 17 return [
17 18 [['id'], 'integer'],
18 19 [['name'], 'string', 'max' => 50],
19 20 [['code'], 'string', 'max' => 15],
20 21];
21 22 }
22 23 ...
23 24 ...
24 25 return [
25 26 'id' => 'ID',
26 27 'name' => 'Name',
27 28 'code' => 'Code',
28 29];
29 30 }
30 31 }
31 32 }
32 33 }
33 34 }
34 35 }
35 36 }
36 37 }
37 38 }
38 39 }
39 40 }
40 41 }
41 42 }
42 43 }
43 44 }
44 45 }
45 46 }
46 47 }
47 48 }
48

```

Below the code editor, there are several configuration sections:

- Generate Relations:** A dropdown menu showing "All relations". Other options include "Generate Labels from DB Comments", "Generate ActiveQuery", "Enable I18N", and "Use Schema Name".
- Code Template:** Set to "default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/model/default)".
- Buttons:** "Preview" and "Generate".
- Action Bar:** Buttons for "Create" (checked), "Unchanged" (checked), and "Overwrite".
- Table:** A table showing the selected file "models\Country.php" with a "diff" button. The "Action" column has a checkbox and a "overwrite" link.

At the bottom, it says "A Product of Yii Software LLC" and "Powered by Yii Framework".

Mavjud faylni qayta yozishda "qayta yozish" yonidagi katakchani belgilang va keyin "Yaratish" tugmasini bosing. Agar yangi fayl yaratilsa, shunchaki "Yaratish" tugmachaсини bosishingiz mumkin.

Keyinchalik, kod muvaffaqiyatli ishlab chiqarilganligini tasdiqlovchi sahifani ko'rasiz. Agar sizda mavjud bo'lgan fayl bo'lsa, unda siz yangi yaratilgan kod bilan yozilganligini ko'rsatadigan xabarni ko'rasiz.

CRUD kodini yaratish

CRUD yaratish, o'qish, yangilash va o'chirish degan ma'noni anglatadi, aksariyat veb-saytlardagi ma'lumotlar bilan olingan to'rtta umumiylazifani aks ettiradi. Gii yordamida CRUD funktsiyasini yaratish uchun "CRUD Generator" ni tanlang (Gii indeks sahfasidagi havolani bosish orqali). "Mamlakat" misoli uchun quyidagi shaklni to'ldiring:

Model sinfi: app\models\Country

Model sinfini qidirish: app\models\CountrySearch

## Nazoratchi sinfi: app\controllers\CountryController

yii code generator

Home Help Application

Model Generator >

**CRUD Generator** > **Controller Generator** > **Form Generator** > **Module Generator** > **Extension Generator** >

## CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

**Model Class**  
app\models\Country

**Search Model Class**  
app\models\CountrySearch

**Controller Class**  
app\controllers\CountryController

**View Path**  
@app\views\country

**Base Controller Class**  
yiiweb\Controller

**Widget Used in Index Page**  
GridView

**Enable I18N**

**Enable Pjax**

**Code Template**  
default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/crud/default)

**Preview**

Keyin, "Ko'rib chiqish" tugmasini bosing. Quyida ko'rsatilgandek yaratiladigan fayllar ro'yxatini ko'rasisiz.

yii code generator

Home Help Application

**View Path**  
@app\views\country

**Base Controller Class**  
yiiweb\Controller

**Widget Used in Index Page**  
GridView

**Enable I18N**

**Enable Pjax**

**Code Template**  
default (/projects/yii2-app/vendor/yiisoft/yii2-gii/generators/crud/default)

**Preview** **Generate**

Click on the above **Generate** button to generate the files selected below:

Code File	Action
controllers\CountryController.php	create <input checked="" type="checkbox"/>
models\CountrySearch.php	create <input checked="" type="checkbox"/>
views\country\_form.php	create <input checked="" type="checkbox"/>
views\country\_search.php	create <input checked="" type="checkbox"/>
views\country\create.php	create <input checked="" type="checkbox"/>
views\country\index.php	create <input checked="" type="checkbox"/>
views\country\update.php	create <input checked="" type="checkbox"/>
views\country\view.php	create <input checked="" type="checkbox"/>

**Create**  **Unchanged**  **Overwrite**

Agar siz controllers/CountryController.phpva views/country/index.phpfayllarni ilgari yaratgan bo'lsangiz (qo'llanmaning ma'lumotlar bazalari bo'limida), ularni almashtirish uchun "ustiga yozish" katagiga belgi qo'ying. (Oldingi versiyalarda CRUD to'liq qo'llab-quvvatlanmagan.)

Qanday ishlashini ko'rish uchun quyidagi URL-ga kirish uchun brauzeringizdan foydalaning:

<http://hostname/index.php?r=country%2Findex>

Ma'lumotlar bazasi jadvalidan mamlakatlarni ko'rsatadigan ma'lumotlar tarmog'ini ko'rasiz. Siz katakchani saralashingiz yoki ustun sarlavhalariga filtr shartlarini kiritish orqali filtrlashingiz mumkin.

Tarmoqda ko'rsatilgan har bir mamlakat uchun uning tafsilotlarini ko'rishni, yangilashni yoki o'chirishni tanlashingiz mumkin. Shuningdek, yangi mamlakat yaratish uchun ariza bilan ta'minlash uchun tarmoqning yuqori qismidagi "Mamlakat yaratish" tugmachasini bosishingiz mumkin.

My Company

Home About Contact Login

Home / Countries

## Countries

Create Country

Showing 1-10 of 10 items.

#	Code	Name	Population	
1	AU	Australia	18886000	
2	BR	Brazil	170115000	
3	CA	Canada	1147000	
4	CN	China	1277558000	
5	DE	Germany	82164700	
6	FR	France	59225700	
7	GB	United Kingdom	59623400	
8	IN	India	1013662000	
9	RU	Russia	146934000	
10	US	United States	278357000	

« 1 »

© My Company 2014

Powered by

My Company

Home About Contact Login

Home / Countries / United States / Update

## Update Country: United States

Code

Name

Population

[Update](#)

Quyida ushbu xususiyatlarning qanday amalga oshirilishini tekshirishni yoki ularni sozlashni istasangiz, Gii tomonidan yaratilgan fayllar ro'yxati keltirilgan:

Nazoratchi: controllers/CountryController.php

Modellar: models/Country.phpvamodels/CountrySearch.php

Ko'rishlar: views/country/\*.php

### **Topshiriqlar**

Talabalar o‘z laboratoriya topshiriqlari bo‘yicha ma’lumotlari bazasidagi ma’lumotlardan foydalanib, Yii2 frameworkda CRUD texnologiyasini qo’llab, sahifa ma’lumotlarini hosil qiladi.

### **III. MUSTAQIL TA'LIM MAVZULARI**

<b>Nº</b>	<b>Mavzu nomi</b>	<b>Soat</b>
1.	HTMLda gipermurojaatlar. HTML formalar bilan ishlash. HTMLda freymlarning roli;	10
2.	Java Script o'zgaruvchilari va arifmetik amallar. Java Script yangi ob'yekt yaratish usullari, Java Script operatorlari bilan ishlash;	8
3.	JavaScript tilida sahifa dizaynini yaratish. Cascading Style Sheets (CSS) fayllar yaratish;	8
4.	Foydalanuvchi interfeysi mukamallashtirish uchun ishlatiladigan freymvorklar tahlili;	10
5.	PHP Web dasturlash tilida klasslar yaratish. PHP Web dasturlash tilida grafika bilan ishlash;	8
6.	PHP Web dasturlash tili operatorlari ustida amallar PHP Web dasturlash tilida massivlar bilan ishlash, Massiv elementlarini tartiblash va ularni boshqarish;	8
7.	Saxifani MySQL ma'lumotlar bazasi bilan bog'lash va jadvallar ro'yxatini ko'rish, MySQL ma'lumotlar bazasi operatorlari bilan ishlash va ularni boshqarish;	8
8.	MySQL ma'lumotlar bazasiga ma'lumotlarni jadvallar yaratish va ularni to'ldirish, MySQL ma'lumotlar bazasida jadvallar yaratishda va ularni boshqarishda qo'llaniladigan operatorlar;	10
9.	Ma'lumotlar bazasi bilan bog'langan xolda tester yaratish. Bazada berilgan ma'lumotlarni eksport qilish.	10
10.	Web saytda qidiruv tizimini optimallashtish arxitekturasini shakllantirish.	10
	<b>Jami</b>	<b>90</b>

## IV. GLOSSARY

<b>Termin</b>	<b>O‘zbek tilidagi sharhi</b>	<b>Ingliz tilidagi sharhi</b>
<b>!=</b>	Teng emas operatori; mantiqiy inkor amali qiymati bilan birhil.	The inequality operator; compares values for inequality returning a bool.
<b>+=</b>	add-and-assign operatori; masalan $a+=b$ vazifazi jihatdan $a=a+b$ bilan bir xil	add-and-assign operator; $a+=b$ is roughly equivalent to $a=a+b$ .
<b>address</b>	Hotira manzili	a memory location
<b>aggregate</b>	Konstruktorsiz massiv yoki struktura	an array or a struct without a constructor
<b>algorithm</b>	Hisoblashning aniq ketma-ketligi	a precise definition of a computation.
<b>and</b>	<b>&amp;&amp;</b> mantiqiy “va” (ko’paytirish) operatori sinonimi	synonym for <b>&amp;&amp;</b> , the logical and operator.
<b>ANSI</b>	Amerika milliy standart agentligi.	The American national standards organization.
<b>application</b>	Umumiy maqsadga ega dasturlar to’plami	a collection of programs seen as serving a common purpose (usually providing a common interface to their users)
<b>bit</b>	0 yoki 1 qiymatga ega birlik hotira	a unit of memory that can hold 0 or 1.
<b>bool</b>	Mantiqiy tip. Bu tip faqat rost yoki yolg’on qiymat qabul qiladi	the built-in Boolean type. A bool can have the values true and false.
<b>bug</b>	Xatolik termini	colloquial term for error.
<b>byte</b>	Xotiradagi bir nechta belgilar yig’indisi	a unit of memory that can hold a character of the C++ representation character set.
<b>char</b>	Belgili tip. Har bir belgi 8 bit, ya’ni baytga teng.	character type; typically an 8-bit byte.
<b>char*</b>	Char massiviga ko’rsatkich	pointer to a char or an array of char. Typically assumed to point to a C-style string.
<b>cin</b>	Standart kiritish oqimi	standard istream.
<b>class</b>	Foydalanuvchi belgilaydigan tur, sinf. Sinf foydalanuvchi funksiyasi, foydalanuvchi ma’lumotlari va kontentlari bo’lishi mumkin.	a user-defined type. A class can have member functions, member data, member constants, and member types.
<b>compiler</b>	C++ da yozilgan buyruqlarni mashina tiliga o’girib beruvchi vosita.	the part of a C++ implementation that produces object code from a translation unit.
<b>const</b>	Faqatgina bir marta qiymat berish	attribute of a declaration that makes the

	mumkin bo'lgan o'zgaruvchilarni e'lon qilish.	entity to which it refers readonly.
<b>copy()</b>	Nusxa olish operatori	Copy operator
<b>UML</b>	Unifikatsiya qilingan modellashtirish tili	Unified Modeling Language
<b>OMG</b>	Obyektlarni boshqarish guruhi	Object Management Group
<b>4GL</b>	To'rtinchchi avlod tili	Fourth-Generation Language
<b>ANSI</b>	Amerika milliy standartlash instituti	American National Standards Institute
<b>AMPS</b>		Advanced Mobile Phone Service
<b>ERP</b>	Korxona resurslarini rejlashtirish	Enterprise Resource Planning
<b>CRM</b>	Mijozlar bilan o'zaro munosabatlarni boshqarish	Customer Relations Management
<b>SQL</b>	Tuzilmalashgan so'rovlar tili	Structured Query Language
<b>OLAP</b>	Xaqiqiy vaqtida ma'lumotlarga analitik ishlov berish	On-Line Analytical Processing
<b>OLTP</b>	Xaqiqiy vaqtida tranzaksiyalarga ishlov berish	On-Line Transaction Processing
<b>TCO</b>	Egalik qilishning yalpi qiymati	Total Cost of Ownership
<b>JIT</b>	Ayni vaqtida	Just-In-Time
<b>LAN</b>	Lokal hisoblash tarmog'i	Local Area Network
<b>MAN</b>	Maxalliy hisoblash tarmog'i	Metropolitan Area Network
<b>WAN</b>	Xududiy hisoblash tarmog'i	Wide Area Network
<b>ISO</b>	Halqaro standartlashtirish tashkiloti	International Organization for Standardization
<b>API</b>	amaliy dasturlashtirish maxsus interfeysi	Application Programming Interface
<b>WWW</b>	Umumjahon o'rgamchak to'ri	World Wide Web
<b>ASCII</b>	Axborot almashishning Amerika standarti	American Standard Code for Information Interchange
<b>LIFO</b>	«Oxirida keldi, birinchi ketdi» prinsipi	Last In, First Out
<b>FIFO</b>	«Birinchi keldi, birinchi ketdi» prinsipi	First In, FirstOut
<b>PDA</b>	Personal raqamli kotib	Personal Digital Assistant

## V. ILOVALAR

### 1. ISHCHI O'QUV DASTURI

**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA  
KOMUNNIKASIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**MUHAMMAD AL – XORAZMIY NOMIDAGI  
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI  
QARSHI FILIALI**

“T A S D I Q L A Y M A N”

O‘quv va tarbiyaviy ishlar

bo‘yicha direktor o‘rinbosari  
\_\_\_\_\_ A.K.Muhammadiev  
“\_\_\_” \_\_\_\_\_ 2021 y.

### «WEB DASTURLASHGA KIRISH»

fanidan

### ISHCHI O'QUV DASTURI (SILLABUS)

**Bilim sohasi:** 300000 – Ishlab chiqarish texnik soha

**Ta’lim sohasi:** 330000 – Kompyuter texnologiyalari va informatika

**Ta’lim yo‘nalishi:** 5330500 – Kompyuter injiniring (AT-servisi)  
5350100 – Telekommunikatsiya texnologiyalari

### Umumiy o‘quv soati – 180 soat

Shu jumladan:

Ma’ruza-60 soat (3 semestr-60 soat)

Laborotoriya mashg‘ulotlari-30 soat (3 semestr-30 soat)

Mustaqil ta’lim soati-90 (3 semestr - 90 soat)

QARSHI – 2021

Ishchi o'quv dastur O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lif vazirligi tomonidan 2020 yil 14 – avgustdagi 418-sonli buyrug'i bilan tasdiqlangan «**Web dasturlashga kirish**» fan dasturi asosida ishlab chiqildi.

Fanning ishchi o'quv dasturi Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Qarshi filiali Kengashining 2021 yil “\_\_\_” “\_\_\_\_\_” dagi “\_\_\_” - sonli bayoni bilan muhokamadan o'tgan va tasdiqlangan.

**Tuzuvchi(lar):**

Sh.T.Ulasheva – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi katta o'qituvchisi

O.U.Begulov – TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti

K.Ablaqulov - TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti

A.Baxriddinov - TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti

M.A.Ro'ziqulova - TATU Qarshi filiali “Dasturiy injiniring” kafedrasi assistenti

Taqrizchilar:

Z.Uzaqov.– TATU Qarshi filiali, “Dasturiy injiniring” “kafedrasi dosenti;

**“Kompyuter injiniringi”**

**fakulteti dekani:**

**2021 yil “\_\_\_” \_\_\_\_\_ Sh.X.Rustamov**

**“Dasturiy injiniring”**

**kafedrasi mudiri:**

**2021 yil “\_\_\_” \_\_\_\_\_ B.N.Nosirov**

**O'quv – uslubiy bo'lim boshlig'i:**

**2021 yil “\_\_\_” \_\_\_\_\_ S.J.Yaxyayev**

## I. Fanning mazmuni

1-jadval

Fan/modul kodi	O‘quv yili	Semestr	ECTS – Kreditlar
WEB1316	2021-2022	3	6
Fan/modul turi	Ta’lim tili		Haftadagi dars soatlari
Majburiy	O‘zbek/rus		6
<b>1.</b>	<b>Fanning nomi</b>	<b>Auditoriya mashg‘ulotlari (soat)</b>	<b>Mustaqil ta’lim (soat)</b>
	<b>Web dasturlashga kirish</b>	90	90
			180

**“Web dasturlashga kirish” fanidan mashg‘ulotlarning mavzular va soatlar bo’yicha taqsimlanishi:**

Mashg‘ulot turi	3 semestr	Jami
Ma’ruza	60	60
Laboratoriya mashg‘ulotlari	30	30
Mustaqil ish	90	90
<b>Jami</b>	<b>180</b>	<b>180</b>

Fanni o‘qitishdan maqsad – talabalarga zamonaviy web dasturlashning nazariy va amaliy bilimlarini va ularni yaratish texnologiyalarini, web dasturlash uchun mo‘ljallangan HTML5, CSS3-4, JavaScript, Bootstrap, AngularJs va PHP dasturlari, zamonaviy freymvork platformalari orqali web texnologiyalarning asosiy tushunchalarini, server tomonidan dasturlash texnologiyalarini, MySQL texnologiyasi va ular bilan ishlashning ko‘nikma va malakalarini shakllantirishdan iborat.

Fanning vazifasi – web dasturlashning nazariy asoslari, ularni tuzilishi va frontend (HTML5, CSS, JavaScript, Bootstrap) va backend (PHP, MySQL, YII2 freymvork) texnologiyalarini yaratilishi, qayta ishlanishi, sayt arxitekturasini shakllantish, foydalanuvchi interfeysi va muhitini yaratish hamda uni boshqarishni o‘rgatish va ularni qo‘llashdan iborat.

## II. Fanni o‘qitishda Ta’lim texnologiyalari va metodlari:

- Ma’ruzalar;
- Interfaol keys-stadilar;

- Seminarlar (mantiqiy fikirlash, tezkor savol-javoblar);
- Laboratoriya mashg‘ulotlari;
- Gruhlarda ishlash;
- Taqdimotlar qilish;
- Individual loyihalar;
- Jamoa bo‘lib ishlash va himoya qilish uchun loyihalar.

### **III. Ma’ruza mashg‘ulotlari**

2-jadval

<b>Nº</b>	<b>Mavzular nomi</b>	<b>Ma’ruz a</b>
1	<b>Mavzu: WEB dasturlashga kirish. Asosiy tushunchalar. WWW rivojlanish bosqichlari</b> Veb texnologiyaga kirish. Asosiy tushunchalar. WWW ning rivojlanish bosqichlari. Kliyent-server texnologiyalari, internet protokollar. Veb ilovaning tarkibiy qismlari. Frontend va backend texnologiyalar.	2
2	<b>Mavzu: HTML tiliga kirish. HTML xujjatining umumiyligi strukturasi va uning asosiy teglari</b> HTML ga kirish. HTML-hujjatlar tuzilishi. HTML ning asosiy teglari, atributlari. Sarlavha teglari. Rasm va jadvallar bilan ishlash teglari. Havolalar va ro‘yxatlar bilan ishlash teglari. HTML formalar.	2
3	<b>Mavzu: HTML5 yangi standartlari va ularning vazifalari</b> HTML5 yangi standartlar. HTML5 da yangi teglar va atributlar. Audio, video va grafika bilan ishlash. Vizual effektlar.	2
4	<b>Mavzu: CSS ga kirish. HTMLga ulanish usullari. Selektorlar turlari</b> CSS ga kirish. CSS sintaksisi. CSS bilan HTML hujjatni bog‘lash. Selektorlar turlari (oddiy, kombinasiyalashgan, psevdo-element, psevdo-sinf, atribut selektorlar). O‘lchov birliklari.	2
5	<b>Mavzu: CSS xususiyatlari. CSS sahifalash</b> CSS rang xususiyatlari. CSSda chegara xususiyatlari. CSSda orqa fon xususiyatlari. CSS shrift xususiyatlari. Margin xususiyatlari. Padding xususiyatlari. Box model tushunchasi. Box model qoidalari.	2
6	<b>Mavzu: CSS3 asoslari. Effektlar, transformasiya, animasiya.</b> CSS3 asoslari. CSS matn effektlari. CSS da ikki o‘lchovli transformasiya. CSS da uch o‘lchovli transformasiya. CSS animasiyalar. CSS preprocessorlari (sass, less).	2
7	<b>Mavzu: JavaScript asoslari. JavaScript ssenariyli tilining xususiyatlari</b> JavaScript asoslari. JavaScript ssenariylarini HTML-xujjat bilan bog‘lash. O‘zgaruvchilar. Ma’lumotlar turlari. Massivlar. Operatorlar.	2

№	Mavzular nomi	Ma’ruz a
8	<b>Mavzu: JavaScript funksiyalari va ob'yektlari</b> JavaScriptda funksiyalar. JavaScriptda ob'yektlar. JavaScript funksiyalari yordamida hodisalarni qayta ishlash. JavaScriptda doimiy ifodalar.	2
9	<b>Mavzu: JavaScript brauzer va veb-hujjat ob'yektlari modeli bilan ishlash</b> Hujjatning ob'yektli modeli haqida tushuncha (DOM). Hujjatning ob'yektli modelida HTML ob'yektlari bilan ishlash. Hujjatning ob'yektli modelida CSS xususiyatlari bilan ishlash. JavaScript brauzer ob'yekti modeli (BOM). Date ob'yekti usullari va xususiyalari	4
10	<b>Mavzu: JQUERY ga kirish. Asosiy tushunchalar</b> JQuery ga kirish. JQuery sintaksisi. JQuery dan foydalanish usullari. JQuery selektorlari. JQuery hodisalari. JQuery UI.	2
11	<b>Mavzu: Angular JS ga kirish. Animasiya va formalar bilan ishlash</b> AngularJS ga kirish. AngularJS da direktiva-lar. AngularJS da ifodalar. AngularJS da modullar. AngularJS da animasiyalar. Forma elementlari bilan ishlash.	4
12	<b>Mavzu: Bootstrap freymvork texnologiyasi. Bazaviy shablonlarni sozlash. Komponentlar.</b> Bootstrap freymvork texnologiyasi. Freym-vorkni ulyash, bazaviy shablonlarni sozlash. Bootstrap freymvorkda tugmalar, ro'yxatlar, navigasiya menyulari, formalar bilan ishlash. Bootstrap komponentlari	4
13	<b>Mavzu: PHP asoslari. Ma'lumotlar turlari</b> PHP ga kirish. PHP asoslari, sintaksisi. PHP versiyalari. O'zgaruvchilar, konstantalar, ma'lumotlar turlari. Operatorlar.	2
14	<b>Mavzu: PHP funksiyalari va ob'yektlari</b> PHP funksiyalari va ob'yektlari. PHPda massivlar, satrlar va fayllar bilan ishlash. Formalar bilan ishlash. PHP da xatolarni qayta ishlash	4
15	<b>Mavzu: PHPda global o'zgaruvchilar</b> PHPda global o'zgaruvchilar: POST, GET, COOKIE, SESSION, SERVER, REQUEST, ENV, FILES. O'zgaruvchilarning ko'rinish sohalari. Fayllar ustida amallar. Kataloglar bilan ishlash.	2
16	<b>Mavzu: PHPda ob'yektga yo'naltirilgan dasturlash. MVC texnologiyasi</b> PHP da sinflar va ob'yektlar. MVC texnologiyasi haqida. Model. View. Controller.	2

№	Mavzular nomi	Ma’ruz a
17	<b>Mavzu: PHPda MySQL ma'lumotlar bazasini boshqarish tizimi bilan ishlash.</b> PHPda ma'lumotlar bazasi bilan ishlash (CRUD) amallari. MySQL ma'lumotlar bazasini boshqarish tizimi. Jadval yaratish. Jadvaldagi ma'lumotlarni o'qish, o'zgartirish, o'chirish. PHP da AJAX texnologiyasi bilan ishlash.	4
18	<b>Mavzu: PHP da autentifi-kasiya va avtorizasiyani amalga oshirish usullari</b> PHP da registrasiya va autentifikasiya sahifasini ishlab chiqish. Foydalanuvchining sahifalarga kirish ruxsatini belgilash usullari.	2
19	<b>Mavzu: Veb-saytlarni ishlab chiqish Backend texnologiya-lari: AngularJs</b> Veb-saytlarni ishlab chiqish Backend texnologiyalarida AngularJs da foydalanish. AngularJs ni MySQL bilan bog'lash.	2
20	<b>Mavzu: Veb saytlarni yaratishda CMS texnologiyasini qo'llash</b> CMS (kontentlarni boshqarish tizimlari) texnologiyasi haqida. Kontentlarni boshqarish tizimlaridan foydalanishning afzallik va kamchiliklari. CMS xususiyatlari. Mavjud zamonaviy CMS turlari.	2
21	<b>Mavzu: Freymvork platformasida veb-dasturlash.</b> <b>YII2 freymvorkka kirish.</b> Freymvork platformasida veb-dasturlash asoslari. YII2 freymvorkka kirish. YII freymvorkni o'rnatish. YII2 freymvork strukturasi. Kirish skriptlari. Model, View, Controller. Ilovalar. Komponentlar. Modul-lar, filterlar va vidjetlar.	4
22	<b>Mavzu: YII2 freymvorkda veb ilovalarni ishlab chiqish asoslari. YII2 da so'rovlarni qayta ishlash</b> YII2 freymvorkda veb ilovalarni ishlab chiqish asoslari. YII2 da so'rovlarni qayta ishlash. URL bilan ishlash. Request komponentining get(), post() usullari. YII2 da loglash tizimi. YII2 da SESSION va COOKIE bilan ishlash.	2
23	<b>Mavzu: YII2 freymvorkda ma'lumotlar bazasi bilan ishlash</b> YII2 freymvorkda ma'lumotlar bazasi bilan ishlash. YII2 da ma'lumotlar bazasi so'rovlarni hosil qilish. ActiveRecord.	4
	<b>Jami:</b>	60

#### IV. Laboratoriya mashg'ulotlari

3-jadval

<b>Nº</b>	<b>Mavzu nomi</b>	<b>Labaratoriya</b>
1	HTML hujjat tuzilishi. HTML ning asosiy teglari, teg attributlari. HTMLda rasm va jadvallar bilan ishlash teglari. Havolalar va ro‘yxatlar bilan ishlash teglari. HTML formalar bilan ishlash.	2
2	HTML5 da yangi teclar. HTML5 da audio, video va grafika bilan ishlash teglari.	2
3	CSS da selektorlar. CSS xususiyatlari. CSS3 da effektlar, transformasiya, animasiya.	2
4	JavaScript funksiyalar va ob’yektlari. JavaScript da doimiy ifodalar.	2
5	JavaScript brauzer va veb-hujjat ob’yektlari modeli bilan ishlash (BOM, DOM). HTML5 da JavaScript API lardan foydalanish.	2
6	JQuery selektorlar, hodisalar. JQUERY UI.	2
7	Angular JS da animasiyalar. Angular JS da forma elementlari bilan ishlash.	2
8	Bootstrap4 freymvorkni ulash, bazaviy shablonlarni sozlash.	2
9	PHPda satrlar, massivlar va fayllar bilan ishlash.	2
10	PHPda formalar bilan ishlash. PHPda COOKIE, SESSION bilan ishlash	2
11	PHPda ma'lumotlar bazasi bilan ishlash. PHPda ma'lumotlar bazasi bilan ishlash (CRUD) amallari.	2
12	PHPda AJAX texnologiyani qo'llash. PHPda avtorizasiya va autentifikasiyanı amalgaloshish usullari.	2
13	Veb-saytlarni ishlab chiqishning Backend tomonida AngularJs dan foydalanish.	2
14	YII2 freymvorkda veb ilovalarni ishlab chiqish. “Model, View, Controller” ni boshqarish.	2
15	YII2 freymvorkda ma'lumotlar bazasi bilan ishlash. YII2 freymvorkda foydalanuvchi rollarini boshqarish usullari	2
	<b>Jami:</b>	<b>30</b>

## **V. Mustaqil ish va topshiriqlarning mavzulari hamda axborot - uslubiy ta'minoti**

Mustaqil ishni bajarishdan asosiy maqsad – professor-o‘qituvchilar-ning bevosa raahbarligi va nazorati ostida talabalarni semestr davomida fanlarni uzluksiz o‘rganishini tashkil etish, olgan bilim, ko‘nikmalarini chuqur o‘rganib mustahkamlash, kelgusidagi darslarga tayyoragarlik ko‘rish, aqliy mehnat madaniyatini, yangi bilimlarni mustaqil ravishda izlab topish va qabul qilishni shakllantirishdan iborat.

Mustaqil ish mazmuni o‘quv jarayonida olingan bilim va ko‘nikmalarni darsliklar, o‘quv qo‘llanmalari, qo‘srimcha adabiyotlar va boshqa axborot uslubiy vositalarini mustahkamlashdan

iborat.

### Mustaqil ish mavzulari

Nº	Mavzu nomi	Soat
11.	HTMLda gipermurojaatlari. HTML formalar bilan ishlash. HTMLda freymlarning roli;	10
12.	Java Script o'zgaruvchilari va arifmetik amallar. Java Script yangi ob'yekt yaratish usullari, Java Script operatorlari bilan ishlash;	8
13.	JavaScript tilida sahifa dizaynini yaratish. Cascading Style Sheets (CSS) fayllar yaratish;	8
14.	Foydalanuvchi interfeysi mukamallashtirish uchun ishlataladigan freymvorklar tahlili;	10
15.	PHP Web dasturlash tilida klasslar yaratish. PHP Web dasturlash tilida grafika bilan ishlash;	8
16.	PHP Web dasturlash tili operatorlari ustida amallar PHP Web dasturlash tilida massivlar bilan ishlash, Massiv elementlarini tartiblash va ularni boshqarish;	8
17.	Saxifani MySQL ma'lumotlar bazasi bilan bog'lash va jadvallar ro'yxatini ko'rish, MySQL ma'lumotlar bazasi operatorlari bilan ishlash va ularni boshqarish;	8
18.	MySQL ma'lumotlar bazasiga ma'lumotlarni jadvallar yaratish va ularni to'ldirish, MySQL ma'lumotlar bazasida jadvallar yaratishda va ularni boshqarishda qo'llaniladigan operatorlar;	10
19.	Ma'lumotlar bazasi bilan bog'langan xolda tester yaratish. Bazada berilgan ma'lumotlarni eksport qilish.	10
20.	Web saytda qidiruv tizimini optimallashtish arxitekturasini shakllantirish.	10
	<b>Jami</b>	<b>90</b>

### VI. Fan o'qitilishining natijalari (shakillanadigan kompetensiyalar)

Fanni o'zlashtirish natijasida talaba:

- predmet sohasini ajratish va uning modelini yaratish; internet asoslari va uning arxitekturasi, algoritm tushunchasi va hech bo'limganda bir dasturlash tilini yoki ushbu dastur asosida yaratilgan freymvorkni, hamda dasturlash texnologiyalarini **bilishi kerak**;
- loyihani bajarish uchun maqbul dasturlash tili (yoki uning freymvorki) va dizayn ilova vositalarini (freymvorklarini) to'g'ri tanlash;
- ishlab chiqilgan loyiha konsepsiysi asosida web ilovani yaratish;
- loyihani ma'lumotlar bazasida so'rovlari yaratish uchun yuqori bosqichli tillarni qo'llash va sayt arxitekturasini to'g'ri shakllantira olish **ko'nikmalariga ega bo'lishi kerak**;
- mavjud dasturlash tillari va ularning tarkibidagi ma'lumotlar bazasi bilan ishslash xarakteristikalarini tahlil qilish, web dasturlashda foydalanuvchilarga dizayn va funksionalligini optimal parametrlarini aniqlay bilishi, tahlil eta olishi va hozirda keng tarqalgan va ommalashgan web-saxifalarni tahlil qilish **malakasiga ega bo'lishi kerak**.

### VII. Fan bo'yicha talabalarning ballini aniqlash jadvali

**Maksimal ball - 100****Saralash ball- 60**

<b>Nº</b>	<b>Nazorat turlari</b>	<b>Topshiriqlar soni</b>	<b>Bir topshiriq uchun maksimal ball</b>	<b>Jami ball</b>
1.	Laboratoriya ishlari bajarish va hisobot topshirish	10	3	30
2.	Mustaqil ishlarni bajarish va hisobot topshirish	5	3	15
3.	Oraliq nazorat (yozma ish yoki test)	1	5	5
	<b>Jami oraliq nazoratlar:</b>	<b>16</b>		<b>50</b>
4.	<b>Yakuniy nazorat topshirish</b>	1	50	<b>50</b>
	<b>Jami:</b>			<b>100</b>

Talabalarga laboratoriya mashg‘ulotlari uchun ajratilgan umumiy 30 ball baholash miqdori har bir laboratoriya mashg‘uloti uchun 3 balldan taqsimlangan.

Har bir laboratoriya mashg‘uloti uchun ajratilgan baholash miqdori 3 ball holatda olinib, quyidagi omillar bo‘yicha baholanadi:

1. Uyga berilgan topshiriqn ni to‘liq va mustaqil bajarib algoritmlar va dasturlar tuzganligi: 3 ball
2. Har bir laboratoriya mashg‘uloti talablar bo‘yicha bajarilganligi va dasturlardan natijalar olinganligi: 2 ball.

Mustaqil ish uchun bajarilgan har bir ishga 3 balldan ajratilgan va ballarni qo‘yish jarayonida quyidagilar e’tiborga olinadi:

1. Mustaqil ish hajmining va rasmiylashtirish qoidalarining talabga muvofiqligi;
2. Referat berilgan mustaqil ish mavzusiga mosligi;
3. Berilgan mavzuning to‘liq yoritilganligi;
4. Mustaqil ishda yangi ma’lumotlardan foydalanilganligi;
5. Mustaqil ishni bajarishda qo‘srimcha adabiyotlardan va maqolalardan foydalanilganligi;
6. Mustaqil ishni bajarish davomida dasturiy mahsulotlar yaratilganligi va tahlil qilinganligi.

O‘quv kursining bir semestri davomida bir marta oraliq nazorat olinadi va maksimal 5 ballgacha baholanadi.

**Laboratoriya (tajriba) ishlari uchun ballarning taqsimlanishi:**

<b>Tajriba ishi tartib raqami</b>	<b>Nº</b>	<b>laboratoriya ishlari mavzulari</b>	<b>Soat</b>	<b>Ajratilgan ball</b>	<b>Deadline (muddat)</b>
1-Tajriba ishi	1.	HTML hujjat tuzilishi. HTML ning asosiy teglari, teg attributlari. HTMLda rasm va jadvallar bilan ishslash teglari. Havolalar va ro‘yxatlar bilan ishslash teglari.	2	3	16.10.2021

<b>Tajriba ishi tartib raqami</b>	<b>Nº</b>	<b>laboratoriya ishlar mavzulari</b>	<b>Soat</b>	<b>Ajratilgan ball</b>	<b>Deadline (muddat)</b>
		HTML formalar bilan ishlash.			
2-Tajriba ishi	2.	HTML5 da yangi teglar. HTML5 da audio, video va grafika bilan ishslash teglari.	2	3	20.10.2021
3-Tajriba ishi	3.	CSS da selektorlar. CSS xususiyatlari. CSS3 da effektlar, transformasiya, animasiya.	2	3	7.11.2021
4-Tajriba ishi	4.	JavaScript funksiyalar va ob'yektlari. JavaScript da doimiy ifodalar.	4	3	21.11.2021
	5.	JavaScript brauzer va veb-hujjat ob'yektlari modeli bilan ishslash (BOM, DOM). HTML5 da JavaScript API lardan foydalanish.			
5-Tajriba ishi	6.	JQuery selektorlar, hodisalar. JQUERY UI.	4	3	28.11.2021
	7.	Angular JS da animasiyalar. Angular JS da forma elementlari bilan ishslash.			
6-Tajriba ishi	8.	Bootstrap4 freymvorkni ulash, bazaviy shablonlarni sozlash.	2	3	05.12.2021
7-Tajriba ishi	9.	PHPda satrlar, massivlar va fayllar bilan ishslash.	4	3	12.12.2021
	10.	PHPda formalar bilan ishslash. PHPda COOKIE, SESSION bilan ishslash			
8-Tajriba ishi	11.	PHPda ma'lumotlar bazasi bilan ishslash. PHPda ma'lumotlar bazasi bilan ishslash (CRUD) amallari.	2	3	19.12.2021
9-Tajriba ishi	12.	PHPda AJAX texnologiyani qo'llash. PHPda avtorizasiya va autentifikasiyanı amalga oshirish usullari.	4	3	25.12.2021
	13.	Veb-saytlarni ishlab chiqishning Backend tomonida AngularJs dan foydalanish.			
10-Tajriba ishi	14.	YII2 freymvorkda veb ilovalarni ishlab chiqish. "Model, View, Controller" ni boshqarish.	4	3	8.01.2022
	15.	YII2 freymvorkda ma'lumotlar bazasi bilan ishslash. YII2 freymvorkda foydalanuvchi rollarini boshqarish usullari			
			<b>30</b>	<b>30</b>	

**Mustaqil ishlarning mavzular, soatlar va ballar bo'yicha taqsimlanishi:**

<b>Mustaqil ish tartib raqami</b>	<b>№</b>	<b>Mustaqil ish mavzulari</b>	<b>Soat</b>	<b>Ajratilgan ball</b>	<b>Deadline (muddat)</b>
<b>1-mustaqil ish</b>	1.	HTMLda gipermurojaatlar. HTML formalar bilan ishlash. HTMLda freymlarning roli;	<b>26</b>	<b>3</b>	16.10.2021
	2.	Java Script o'zgaruvchilari va arifmetik amallar. Java Script yangi ob'yekt yaratish usullari, Java Script operatorlari bilan ishlash;			
	3.	JavaScript tilida sahifa dizaynnini yaratish. Cascading Style Sheets (CSS) fayllar yaratish;			
<b>2-mustaqil ish</b>	4.	Foydalanuvchi interfeysi mukamallashtirish uchun ishlatiladigan freymvorklar tahlili;	<b>10</b>	<b>3</b>	6.11.2021
<b>3-mustaqil ish</b>	5.	PHP Web dasturlash tilida klasslar yaratish. PHP Web dasturlash tilida grafika bilan ishlash;	<b>16</b>	<b>3</b>	4.12.2021
	6.	PHP Web dasturlash tili operatorlari ustida amallar PHP Web dasturlash tilida massivlar bilan ishlash, Massiv elementlarini tartiblash va ularni boshqarish;			
<b>4-mustaqil ish</b>	7.	Saxifani MySQL ma'lumotlar bazasi bilan bog'lash va jadvallar ro'yxatini ko'rish, MySQL ma'lumotlar bazasi operatorlari bilan ishlash va ularni boshqarish;	<b>18</b>	<b>3</b>	20.12.2021
	8.	MySQL ma'lumotlar bazasiga ma'lumotlarni jadvallar yaratish va ularni to'ldirish, MySQL ma'lumotlar bazasida jadvallar yaratishda va ularni boshqarishda qo'llaniladigan operatorlar;			
<b>5-mustaqil ish</b>	9.	Ma'lumotlar bazasi bilan bog'langan xolda tester yaratish. Bazada berilgan ma'lumotlarni eksport qilish.	<b>20</b>	<b>3</b>	8.01.2022
	10.	Web saytda qidiruv tizimini optimallashtish arxitekturasini shakllantirish.			
			<b>90</b>	<b>15</b>	

### **VIII. Fan bo'yicha talabalar bilimining baholash mezonlari**

Talabalarning fan bo'yicha o'zlashtirish ko'rsatkichlarini nazorat qilish quyidagi namunaviy mezonlar asosida amalga oshiriladi.

Ball	Baho	Talabalar bilim darajasi quyidagilarga javob berishi kerak:
90-100	A'lo	Xulosa va qaror qabul qilish; Ijodiy fikrlar olish; Mustaqil mushohida yurita olish; Olgan bilimlarini amalda qo'llay olish; Mohiyatini tushinish; Bilish, aytib berish; Tasavvurga ega bo'lish.
70-89	Yaxshi	Mustaqil mushohada yurita olish; Olgan bilimlarini amalda qullay olish; Moxiyatini tushinish; Bilish, aytib berish; Tasavvurga ega bo'lish.
60-69	Qoniqarli	Moxiyatini tushinish; Bilish, aytib berish; Tasavvurga ega bo'lish.
0-59	Qoniqarsiz	Aniq tasavvurga ega bo'lmaslik; Bilmaslik.

### **IX. Kreditlarni olish uchun talabalar:**

Fanga oid nazariy va uslubiy tushunchalarini to'la o'zlashtirish, tahlil natijalarini to'g'ri aks ettira olish, o'r ganilayotgan jarayonlar haqida mustaqil mushohada yuritish va joriy, oraliq nazorat shakillarida berilgan vazifa va topshiriqlarni bajarish, yakuniy nazorat bo'yicha yozma ishni topshirish.

### **X. AXBOROT – USLUBIY TA'MINOT**

#### **Asosiy adabiyotlar**

- Дронов В.А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов (pdf+epub) – СПб.: БХБ Петербург 2016. 688 стр.

2. Мартишин С.А. Базы данных. Практическое применение СУБД SQL –и NoSQL – типа для проектирования информационных систем: учеб. Пособие // - Москва: ИД «ФОРУМ» - ИНФРА-М, 2019, – 368 с.
3. Rahul Batra. SQL Primer An Accelerated introduction to SQL Basics.// Gurgaon, India. 2019. –P 194.
4. Крис Аквино, Todd Ганди HTML и CSS, JavaScript. Изд. Питер. 2018 гг. Стр. 512
5. Nazirova E.Sh., Sadullaeva Sh.A., Abidova Sh.B., Tajiev J.A. Web ilovalarni yaratish / T.: “Aloqachi”, 2018, 356 b.
6. Niederst, Jennifer. Learning Web Design. A Beginner's Guide to HTML, Graphics, and Beyond / J. Niederst. - Beijing ; Mumbai : O'reilly, SPD, 2006. - 454 s.
7. Bayross, Ivan. Web Enabled Commercial Applications Development Using Java 2. New Delhi : BPB Publications, 2013. - 646 p.
8. Bayross, Ivan. Web Enabled Commercial Applications Development Using HTML, JavaScript, DHTML and PHP: - New Delhi : BPB Publications, 2013. - 539 p.

#### **Qo'shimcha adabiyotlar**

9. O'zbekiston Respublikasi prezidentining 2017 yil 7 fevraldag'i PF-4947-sон "O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasi to'g'risida"gi Farmoni.
10. Mirziyoyev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. Toshkent. «O'zbekiston», NMU, 2017. – 488 b.
11. Разработка WEB приложение. К.Ф, Керимов, Ш.К. Камолов, Ш.Ш. Мухсинов, Н.П. Сиддикова. Ташкент, «Алокачи», 2019, -382 с.
12. Fundamentals of database systems sixth edition. Ramez Elmasri. Department of Computer Science and Engineering The University of Texas at Arlington. 2011. – 261 s.
13. Введение в Oracle 10г. Перри Джеймс, Пост Джералд. 697 стр 2013
14. Диго С.М. Базы данных Проектирование и использование. издательство "Финансы и статистика". 2005 г. – 592 с.
15. Конноли Т., Брегк К. Базы данных, проектирование, реализация и сопровождения, теория и практика, Университет Пейсли, Шотландия, изд. М.- СПб.- Киев. 2003. – 264 с.
16. E.Sh. Nazirova, D. K. Sharipov. Web dizayn: fanidan talabalarning mustaqil ta'lmini bajarish bo'yicha uslubiy ko'rsatma / TATU. 2017. - 30 b.

#### **Axborot manbaları**

17. [www.intuit.ru](http://www.intuit.ru)
18. <https://yiiframework.com>
19. [www.library.tuit.uz](http://www.library.tuit.uz)
20. [www.ziyonet.uz](http://www.ziyonet.uz)
21. <http://www.tutorialspoint.com/>
22. <http://www.php.net>
23. <http://www.phpframework.com>
24. <http://www.natahaus.ru/>
25. [www.w3schools.com/](http://www.w3schools.com/)
26. [www.tutorialspoint.com/php/](http://www.tutorialspoint.com/php/)
27. <https://www.codecademy.com/tracks/php>
28. [www.freewebmasterhelp.com › Tutorials › PHP](http://www.freewebmasterhelp.com › Tutorials › PHP)