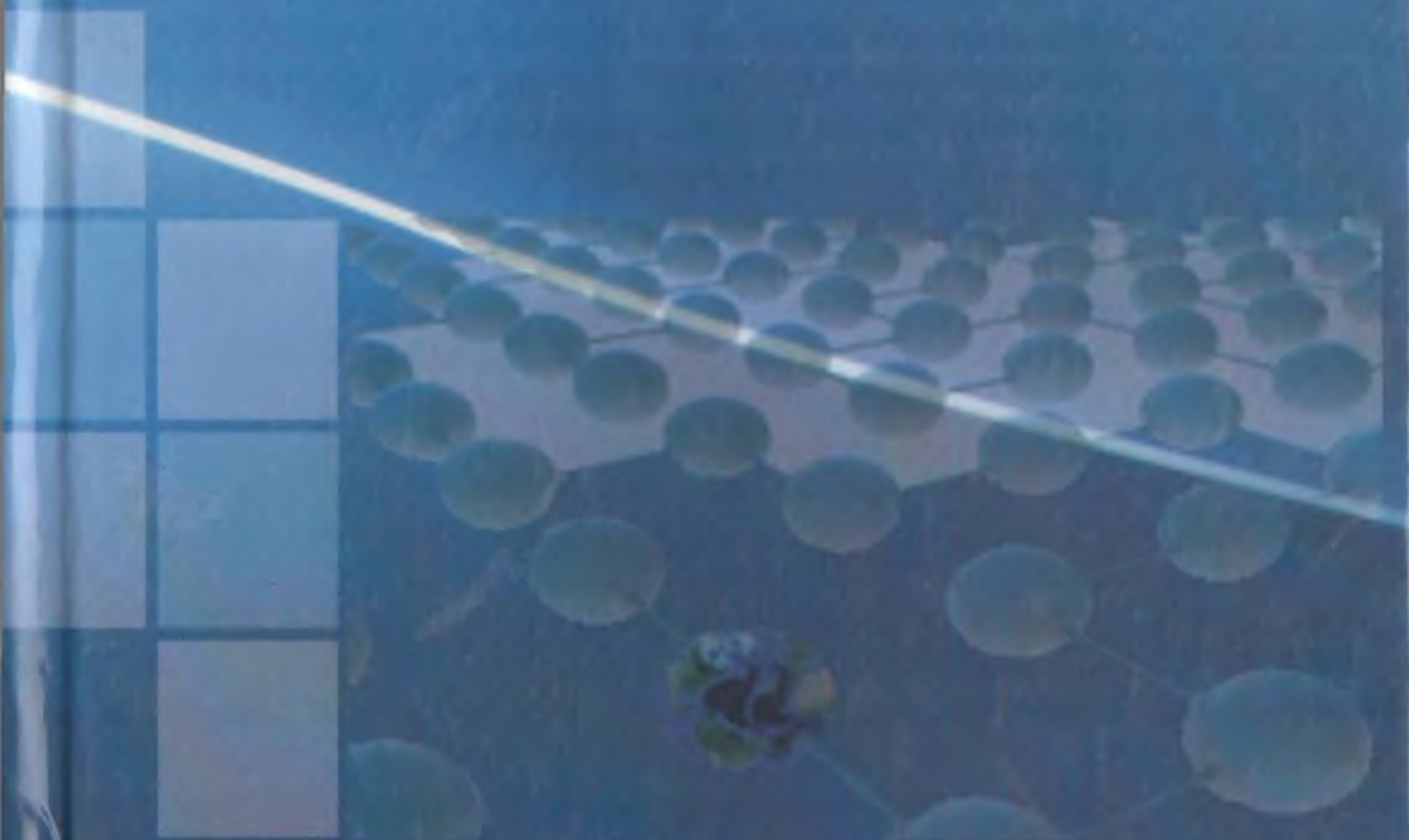


03. 973
#-75

M. ARIPOV,
M. FAYZIYEVA, S. DOTTOYEV

WEB TEKNOLOGIYALARI



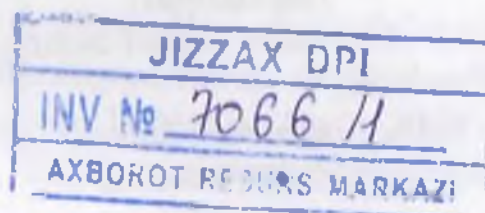
TOSHKENT – 2013

A-75
O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS
TA'LIM VAZIRLIGI

M. ARIPOV, S. DOTTOYEV, M. FAYZIYEVA

WEB TEKNOLOGIYALARI

*Oliy va o'rta maxsus ta'lim vazirligi tomonidan
5A130202 — Amaliy matematika va axborot
texnologiyalari mutaxassisligi yo'nalishi talabalari
uchun o'quv qo'llanma sifatida tavsiya etilgan*



O'zbekiston faylasuflari milliy jamiyati nashriyoti
Toshkent — 2013

UO‘K: 004(075)
KBK: 32.973.26-018.2

Aripov M.

A-75 WEB texnologiyalari: o‘quv qo‘llanma / M.Aripov, S.Dottoyev, M.Fayziyeva; O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta‘lim vazirligi. — Toshkent: O‘zbekiston faylasuflari milliy jamiyati nashriyoti, 2013. — 280 b.

UO‘K: 004(075)
KBK: 32.973.26-018.2

Bugungi kunda ta‘lim tizimida axborot kommunikatsion texnologiyalaridan, jumladan Web texnologiyalar va ma‘lumotlar bazasini boshqarish tizimlaridan foydalanish, shuningdek, ma‘lumotlarni boshqarishni avtomatlashtirish muhim vazifalardan hisoblanmoqda. Ushbu o‘quv qo‘llanmada ana shu masalalar xususida: mavjud muammolar, ularni bartaraf etish yo‘llari va usullari haqida fikr yuritilgan.

O‘quv qo‘llanmada Apache+PHP+MySQL birgalikda ishlatiluvchi zanjir tizimi to‘liq yoritilgan bo‘lib, ushbu uchlik yordamida Webga mo‘ljallangan ma‘lumotlar bazasini yaratish va ulardan foydalanish bo‘yicha ma‘lumotlar berilgan. Shuningdek, qo‘llanmada terminologik masalalarga ham e‘tibor qaratilgan va yetarlicha yoritishga harakat qilingan.

O‘quv qo‘llanma bakalavr va magistratura mutaxassisligi talabalari uchun mo‘ljallangan bo‘lib, Web texnologiyalari va Web dasturlash fanlarining asosiy mazmunini ochib berishga qaratilgan. Qo‘llanmada talabalar mustaqil o‘zlashtirishlari uchun topshiriqlar, ularning yechimlari va dasturiy kodlar berilgan.

Taqrizchilar:

A. Xoljigitov — fizika-matematika fanlari doktori, professor, TATU ilmiy ishlar bo‘yicha prorektori,

B. Abduraximov — fizika-matematika fanlari doktori, professor, O‘zMU “Informatika va tadbqiqiy dasturlash” kafedrası mudiri.

ISBN 978-9943-391-64-2

© O‘zbekiston faylasuflari milliy jamiyati nashriyoti, 2013

KIRISH

Internetning paydo bo'lishi o'zaro muloqotni amalga oshirish uchun yangi kommunikatsion texnologiyalar hamda ma'lumotlar bazasi bilan ishlovchi texnologiyalarni ishlab chiqishni taqozo etdi. Shu kungacha internetda sahifalar yaratuvchi va ular bilan ishlovchi oddiy texnologiyalar ishlab chiqilgan edi. PHP – MBBT (ma'lumotlar bazasini boshqarish tizimi) bilan amalda oson va tez bog'lana olish imkoniyatini beruvchi Web dasturlashning yangi texnologiyalaridan biri hisoblanadi.

PHP Webda ishlovchilar uchun html-sahifalarni tez va oson, dinamik holda ishlashi va o'zgartirish uchun imkoniyatlar yaratib beruvchi texnologiyadir. PHP texnologiyasining ajoyib xususiyatlaridan biri bu – ochiq dasturiy kodga egaligi va bepul tarqatilishidir. PHP platformaga bog'liq emasligi, deyarli barcha platformalarda ishlay olish imkoniyati bilan ajralib turadi. Shuning uchun dunyoda ko'plab Web sayt yaratuvchilar ushbu texnologiyani ma'qul ko'rishadi. Boisi PHP kuniga milliondan ortiq murojaatlarga xizmat ko'rsata oladi, unda skriptlarni yaratish va sozlash boshqa dasturlash tillariga nisbatan ancha osondir.

Ushbu qo'llanma uch qismdan iborat bo'lib: birinchi qismda PHP texnologiyasi va uning imkoniyatlari haqida so'z boradi. Asosiy funksiyalar, standart funksiyalar, obyektga yo'naltirilgan dasturlash borasidagi imkoniyatlar haqida ma'lumotlar berilgan. PHP yordamida yaratilgan dasturlarda HTML ning ba'zi elementlarini ishlatish haqida ham batafsil to'xtab o'tilgan.

Ikkinchi qism ma'lumotlar bazasini boshqarish tizimlarining (MBBT) imkoniyatlari va foydalanish uchun funksiyalar to'plami, PHP texnologiyasining MySQL MBBT bilan integratsiyasiga bag'ishlangan. Ushbu bo'limda PHP va MySQL MBBT yordamida dinamik WEB saytni loyihalash va dastur kodlari misollar yordamida tushuntirib o'tilgan.

Uchinchi qism PHP texnologiyasining Web tizimlardan foydalanishda avtorizatsiyani tashkil qilishga bag'ishlangan. Bu o'rinda ham PHP o'zining sodda interfeys va keng imkoniyatlarini namoyon qiladi.

1-qisim. PHP TEXNOLOGIYASI

I bob. PHP VA MYSQL

Bu bobda PHP ning yaratilish tarixi (PHP/FI dan PHP5 gacha); uning imkoniyatlari (samaradorligi, ma'lumotlar bazasi bilan integratsiyasi, biriktirilgan kutubxonalari, o'rganishda soddaligi, dastur kodi); ishlatish usullari; MySQL ma'lumotlar bazasi va uning yaratilish tarixi, imkoniyatlari (samaradorligi, soddaligi, xavfsizligi, texnik imkoniyatlari va h.k.); Apache+PHP dasturiy ta'minotini o'rnatish va sozlash kabilar o'rganiladi.

1.1. PHP tarixi va uning imkoniyatlari

PHP texnologiyasi Rasmus Lerdorf (Rasmus Lerdorf) tomonidan 1994-yilning kuz oylarida yaratilgan. PHP texnologiyasining birinchi versiyasi Perl da yozilgan bo'lib, CGI obolochka sifatida bo'lgan. PHP texnologiyasi yaratilganiga uncha ko'p vaqt bo'lmasdan, 1995-yilning o'rtalariga kelib esa forma yordamida ma'lumotlar bilan ishlovchi Form Interpreter (FI) qo'shildi. Bu versiya ma'lumotlar bazasi bilan ishlay oldi, ya'ni mSQL ni qo'llay oldi. 1998-yili shu texnologiyaning avvalgilariga qaraganda mukammalroq va qaytadan yozilgan PHP3 versiyasi yaratildi. Shundan so'ng PHP dunyo bo'yicha yanada ommalashib ketdi.

NetCraft ning hisob-kitoblariga ko'ra 1999-yilga kelib PHP dunyo bo'yicha 1 milliondan ortiq saytda ishlatilgan. 2001-yilning yanvarida esa bu son 5 millionga yetgan. Bugungi kunga kelib, dasturchilarning 100 mingdan ortig'i PHP dan foydalanishmoqda. PHP esa Apache serverida ishlovchi 14,5 mln ta saytning taxminan 52% ida ishlatilmoqda.

PHP texnologiyasining imkoniyatlariga batafsil to'xtalamiz.

Samaradorligi. PHP samaradorligi yuqori bo'lgan WEB sahifalar yaratishda yuqori natija beruvchi yagona texnologiyadir. U bir kunda milliondan ortiq murojaatlarga xizmat qila oladi. PHP da skriptlarni yaratish va sozlash boshqa dasturlash tillariga qaraganda osonroqdir.

Ma'lumotlar bazasi bilan integratsiyasi. PHP bir qancha ma'lumotlar bazasini boshqarish tizimlari bilan aloqa o'rната oladi. U MySQL ga qo'shimcha ravishda PostgreSQL, mSQL, Ora-

cle, dbm, Hyperware, Informix, Interbase va Sybaselar bilan ham bog'lanishi mumkin.

Biriktirilgan kutubxonalarining mavjudligi. Turli xil masalalarni yechishda ko'pgina kichkina CGI – ssenariylarini yozishga hojat yo'q, chunki bu sahifani yuklanishini sekinlashtiradi va sayt sahifalarini yaratishda xalaqit beradi. PHP texnologiyasi WEB bilan bog'lanish va bir qancha umumiy masalalarni hal qilish imkoniyatini yaratuvchi, qo'shimcha tashqi kutubxonalar yordamida kengaytirilgan turli foydali funksiyalar to'plamiga egadir.

Narxi. PHP paketi mutlaqo bepuldir. Uning oxirgi versiyalarini <http://www.php.net> saytidan bepul yozib olish mumkin.

PHP ni o'rganish. PHP sintaksisi boshqa dasturlash tillariga asoslangan, birinchi navbatda C va Perl ga. Agar foydalanuvchi C yoki Perl bilan yoki C++ yoxud Java bilan tanish bo'lsa, PHPda dastur yozishda tezda yuqori samaradorlikka erishishi mumkin.

Ko'chuvchanligi. PHP paketini turli operatsion tizimlar boshqaruvida ishlatish mumkin. PHP kodini Linux va FreeBSD singari bepul, Solaris va IRIX singari kommersion UNIX operatsion tizimlar muhitida, Microsoft Windows ning turli versiyalarida yaratish imkoniyati mavjud. Odatda, dasturlar turli xil PHP o'rnatilgan operatsion tizimlarda hech qanday o'zgarishlarsiz ishlaydi.

Hajmi. PHP texnologiyasida yaratilgan har qanday WEB sayt xotiradan uncha katta hajmni egallamaydi va u ochiq tizim sifatida ishlovchi ilg'or texnologiya.

PHP texnologiyasi Perl, Microsoft ning Active Server Pages(ASP), Java Server Pages(JSP) va Allaire Cold Fusion texnologiyalariga raqib bo'la oladi va MySQL MBBT bilan birgalikda funksional, dinamik WEB saytlarni yarata oladi.

1.2. MySQL: tarix va imkoniyatlar

SQL (Structured Query Language — Strukturalashgan so'rovlar tili) — relyatsion MB bilan ishlashda qo'llaniladigan so'rovlar tili.

MySQL(My Structured Query Language) — relyatsion ma'lumotlar bazasini boshqarish tizimidir. MySQL ning tarixiga nazar solsak, bu tizim Mixael Videnius tomonidan yaratilgan va dasturiy ta'minot bozoriga 1996-yili chiqarilgan bo'lsa-da, aslida bu tizimdan 1979-yildan buyon keng foydalanilmoqda.

MySQL ning imkoniyatlari bilan tanishib chiqamiz.

Samaradorligi. MySQL – foydalanish uchun juda tezkor, ishonchli, oson va oddiy tizimdir. MySQL client/server texnologiyasi asosida ishlaydi. O‘zining yengil qo‘llanilishi bilan MySQL tizimi Internet tarmog‘ida juda keng tarqalgan.

Arzon narxliligi va foydalanishdagi soddalik

Dastur kodining ochiqqligi. MySQL paketi ochiq kodli dasturiy ta‘minot hisoblanadi, uni bunday nomlanishining sababi bu – har bir foydalanuvchi dasturni ishlata olishi va kodini o‘qib chiqqan holda ichki kodini o‘rganishi va o‘ziga moslashtirib o‘zgartirishi mumkin. Xuddi PHP kabi MySQL ning dasturiy kodini qchib ko‘rish va o‘zgartirish mumkin.

Arzon narxliligi. MySQL kabi ochiq kodli dasturlarni Internet tarmog‘idan ko‘chirib olish va undan bepul foydalanish mumkin. MySQL paketiga talab bo‘lgan paytlarda ochiq kodli dasturiy ta‘minot litsenziyasiga asosan bepul tarqatiladi.

Sozlash va o‘rganishda soddaligi. Ko‘pgina zamonaviy MBBT-larida SQL qo‘llanilgan. Agar foydalanuvchi birorta MBBTlari bilan tanish bo‘lsa, MySQL ga o‘tishda u hech qanday qiyinchilikka duch kelmaydi. Xuddi ko‘pgina dasturlar singari MySQL ni sozlash va o‘rnatish ham sodd.

Xavfsizligi. Ma‘lumotlar xavfsizligini yuqori darajada ta‘minlaydi. MySQL tizimi ma‘lumotlarga ruxsat berishni boshqaradi, ma‘lumotlar bilan ishlash uchun bir vaqtning o‘zida bir nechta foydalanuvchiga tezkor ruxsatni ta‘minlaydi va foydalanuvchilarga faqatgina ruxsat berilgan ma‘lumotlar bilan ishlashiga kafolat beradi.

Parollar va huquqlarning mukammal tizimi. Tizim parol va vakolatlarga (priviligiyalarga) asoslangan yuqori darajali, mukammal va xavfsiz bo‘lib, xost vositalari orqali tekshiruvni tashkil qilishga imkon yaratadi. Parollar himoyalangan, bog‘lanish vaqtida barcha trafik parollari shifrlanadi. Parollar jadvalida 16 tagacha kalit mavjud bo‘lib, har bir kalit 15 tagacha maydondan foydalana olishi mumkin.

Turli operatsion tizimlarga mosligi. Bir qancha turli operatsion tizimlarda va platformalarda ishlaydi. MySQL dan UNIX, Microsoft Windows va boshqa turli muhitlarda ham foydalanish mumkin.

MySQL ning ayni vaqtda juda ko'p muxoliflari mavjud bo'lib, ular qatoriga mSQL, dbm, Hyperware, Informix, Interbase, Sybase, PostgreSQL, Microsoft SQL Server va Oracle lar kiradi.

1.3. XAMPP dasturiy ta'minotini o'rnatish va sozlash

Ushbu bo'limda eng yaxshi serverlardan biri bilan tanishish mumkin. Bu Apache serveridir. Apache —Internet da keng tarqalgan web-serverlardan biridir. Hozirgi vaqtda Apache dasturiy ta'minoti taxminan butun jahon Web-uzel(tugun)larining yarmida o'rnatilgan. Shuningdek, ba'zi ilovalar ham tavsiya etiladi. Bular PHP va mashhur MySQL MBBT interpretatoridir. Kitobning bu qismi bilan tanishgandan va internetdan distributivlarni olgandan (hammasi bepul) keyin Web da professional darajada ishlash uchun zarur barcha dasturlarga ega bo'linadi.

PHP dasturlash tilini o'rganishni boshlash uchun Apache virtual serverini barcha zarur modullarini hamda MySQL ma'lumotlar bazasini o'rnatish va sozlash lozim.

Buning eng sodda va qulay usuli — Apache, MySQL, PHP lar birlashtirilgan XAMPP paketini o'rnatishdir.

XAMPP — o'zida Apache, MySQL, PHP interpretatori, Perl dasturlash tili va bir qancha qo'shimcha kutubxonalarni jamlagan krossplatformali yig'ilgan Web server.

XAMPP — bu :

- X (ixtiyoriy to'rtta operatsion tizim),
- Apache,
- MySQL,
- PHP,
- Perl.

XAMPP barcha 32-razryadli Microsoft operatsion tizimlarida(98/2000/XP/2003/Vista/7) hamda Linux, Mac OS X va Solaris tizimlarida ishlaydi. Dastur GNU(General Public License) litsenziyasiga asosan barchaga ochiq va bepul tarqatiladi. Bu dastur web serverlar bilan ishlash, dinamik sahifalar yaratish uchun qulay.

XAMPP ni o'rnatish uchun zip, tar yoki exe formatidagi faylni yuklab olish zarur, dastur komponentlari sozlashni talab etmaydi. Dastur doimiy tarzda tarkibiga Apache/MySQL/PHP va Perl larning yangi versiyalarini olib yangilanib turadi. Dasturning foyda-

lanuvchi interfeysi sodda tuzilgan. XAMPP ni o'rnatish uning komponentlarini har birini alohida o'rnatishga nisbatan kam vaqt oladi.

O'rnatish va o'chirish

- XAMPP ni o'rnatishning eng oson usuli — o'rnatgichdan foydalanish(38Mb).

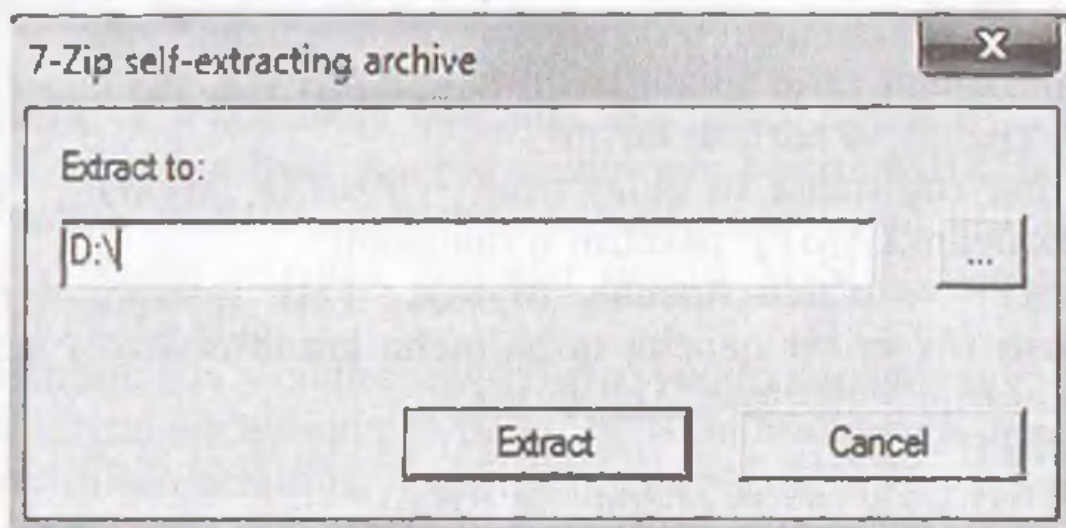
- O'rnatgichsiz ham o'rnatish mumkin.

- O'chirish uchun barcha serverlar va XAMPP xizmatlari yopilishi talab etiladi va XAMPP papkasi o'chiriladi (agar XAMPP o'rnatgich yordamida o'rnatilgan bo'lsa, dasturni va reyestrtdagi yozuvlarni o'chirish uchun o'chirish dasturi ishga tushiriladi).

Dasturni o'rnatish ikki bosqichda amalga oshiriladi: birinchi bosqich — dasturni o'rnatish, ikkinchi bosqich — dasturni testlash.

XAMPP dasturiy ta'minotini o'rnatish

Birinchi bosqich



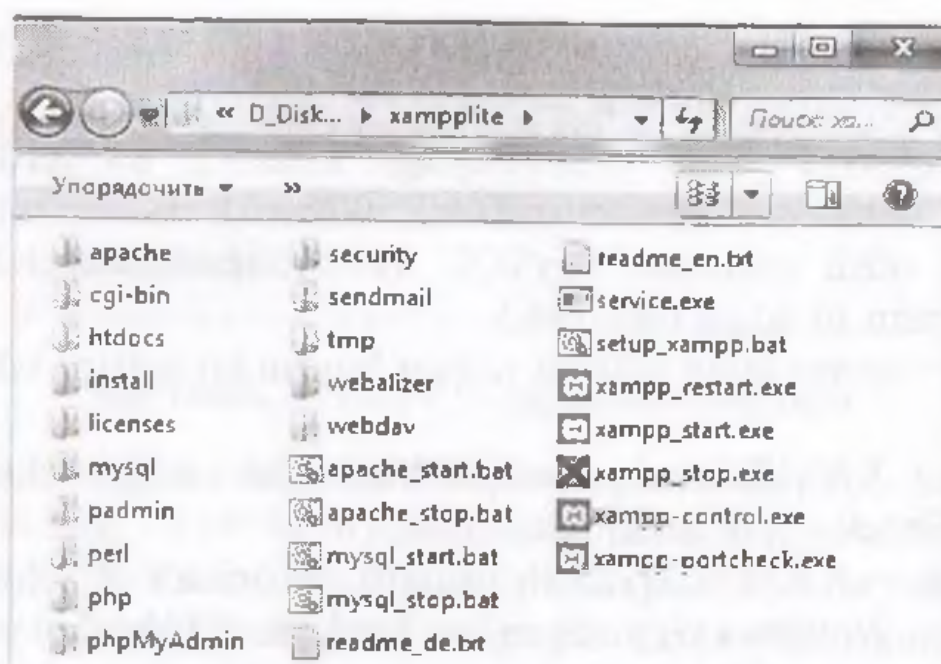
1.1-rasm. XAMPP dasturiy ta'minotini o'rnatish

XAMPP ni o'rnatgichdan foydalanib o'rnatish.

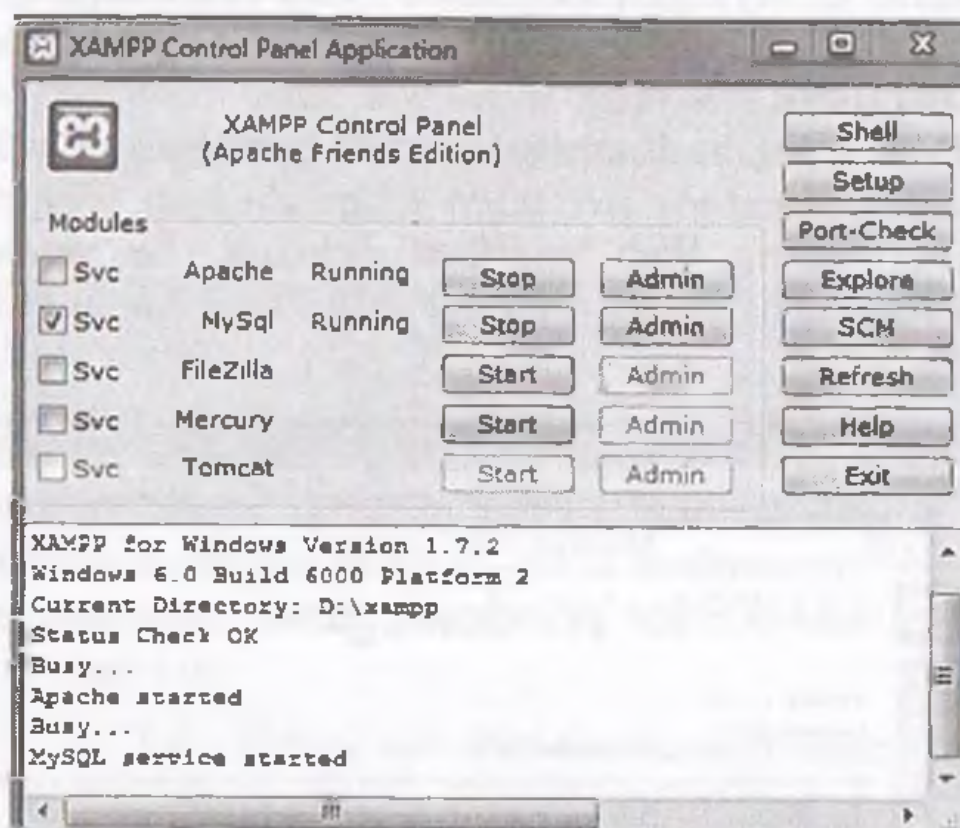
1. Fayllari o'zi o'rnatiluvchi hajmi kichikroq 7-ZIP arxiv (Installer va ZIP arxivini ham yuklab olish mumkin) dasturiy ta'minoti yuklab olinadi. Dasturiy ta'minot yuklab bo'lingandan so'ng, exe fayli ishga tushiriladi. Hosil bo'lgan oynada o'rnatish uchun disk tanlanadi " D: \":

O'rnatgich barcha fayllarni o'rnatadi.

2. O'rnatgandan so'ng ko'rsatilgan manzil bo'yicha qidiriladi (D:/xampplite katalogidagi xampp-control.exe fayli yordamida ishga tushirishingiz mumkin).



1.2-rasm. XAMPP dasturiy ta'minoti o'rnatilgan papka



1.3-rasm. XAMPP DT ning boshqaruv paneli

Checkboxes (Svc) – Windows xizmatchilari sifatida modullarni o'rnatadi (agar siz biror-bir modulni (Apache, MySQL va sh.k.) xizmatchi sifatida ishga tushirmoqchi bo'lsangiz, u holda mos bayroqchani belgilaymiz).

Admin – administrator modullarini ishga tushiradi (Apache moduli qarshisidagi Admin – XAMPP oynasini ochadi, tilni tanlaymiz. XAMPP Web – interfeysiga kiramiz, bu yerda Demos bo'limida demonstratsion misollarni tekshirish va modullar haqida ma'lumot olish mumkin. MySQL moduli qarshisidagi Admin – phpMyadmin ni ishga tushiradi).

Shell – server bilan ishlash uchun buyruqlar satrini ishga tushiradi;

Setup – XAMPP sozlash uchun buyruqlar satrini ochadi;

Port-Check – portlarni tekshirish;

Explore – XAMPP o'rnatilgan papkani ochadi (D:\xampp);

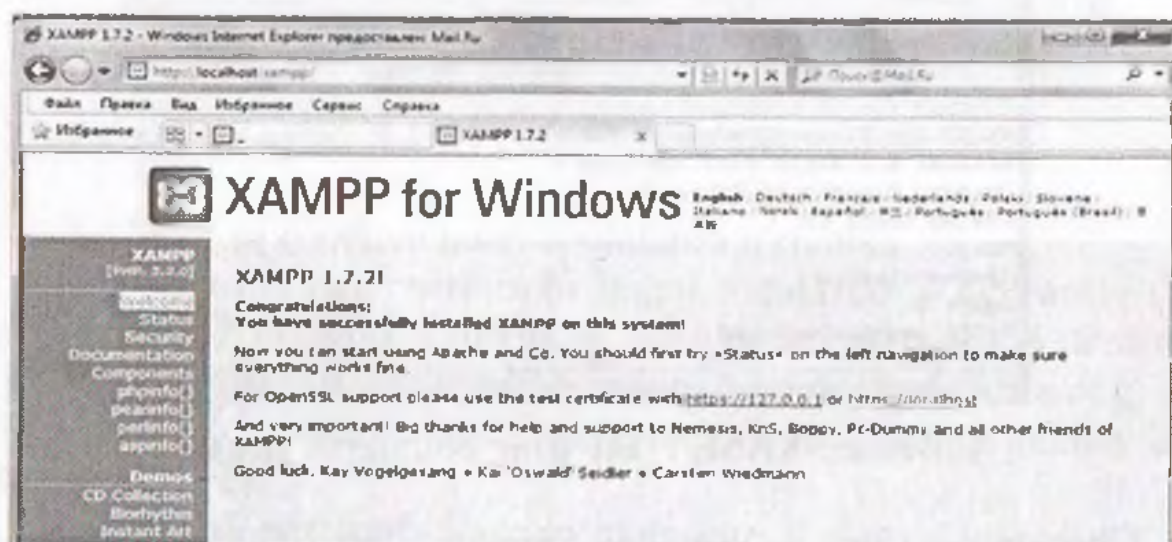
SCM – Windows xizmatlarining boshqaruv oynasini ochadi;

Refresh – modullar holatini yangilash (обновление);

3. Apache va MySQL modullarini ishga tushiramiz (Start tugmasi), brauzerda localhost ni teramiz (yoki Apache moduli qarshisidagi Admin ni bosamiz).

4. XAMPP ning bosh sahifasi ochiladi, berilgan tillardan biror-tasini tanlaymiz.

5. XAMPP ning Web interfeysi ochiladi, bu yerda Demos bo'limida turli hujjatlar va ularning misollari bilan tanishishingiz mumkin va b.



1.4-rasm. XAMPP ning Web interfeysi

6. Security bo'limida xavfsizlikni sozlash qismlari joylashgan.

To fix the problems for mysql, phpmyadmin and the xampp directory simply use

=> <http://localhost/security/xamppsecurity.php> <= [allowed only for localhost]

Some other important notes:

1.5-rasm. XAMPP ning security bo'limi

Ushbu <http://localhost/security/xamppsecurity.php>, ko'rsatilgan manzilga kiring, bu yerda MySQL ma'lumotlar bazasi administratori uchun parol berish mumkin va lokal tarmoqdan foydalanishni chegaralash uchun XAMPP direktoriyasiga parol o'rnatish mumkin:

Ikkinchi bosqich

XAMPP Web serveri bilan tanishib chiqdik va o'rnatdik, endi matnli sahifa yaratish va uni proyektga joylashtirishni ko'rib o'tamiz.

1 Lokal Web serverda sayt yaratish uchun, D:\xampplite\htdocs\ papkasida sayt nomi bilan bog'liq bo'lgan nom bilan papka yaratamiz, masalan, Talabalar portalini yaratish uchun talaba papkasini yaratamiz. Shu papkada matn muharriri yordamida php-skript ni yozamiz:

```
<?php  
echo 'Hello world!';  
?>
```

Ushbu faylni index.php deb saqlaymiz.

2. Brauzerni ishga tushirib manzil qatorida <http://localhost/talaba/> manzilini yozib, sahifani ishga tushiramiz. Agar barcha ish to'g'ri bajarilgan bo'lsa, brauzerda "Hello world!" yozuvini ko'rishimiz mumkin.

1.4. Virtual serverlar (xostlar) yaratish

Asosiy konfiguratsion fayli httpd.conf — web uzelni boshqarish uchun ishlatiladi. U yordamida asosiy amallar (server lokal resurslar bilan qanday ishlash mumkin, so'rovlarga javob berish, foydalanuvchi qaysi fayllar bilan amallarni bajarishi mumkinligi) aniqlanadi.

U orqali Apache ni ishlashini boshqarish mumkin. Web serverning konfiguratsion faylini sozlash — uni o'rnatishdagi eng muhim

ishlardan biridir. Server konfiguratsion faylni ishga tushganda o'qiydi. Agar server ishlayotgan bo'lsa, u holda fayl konfiguratsiyasini o'zgartirgandan so'ng uni qayta yuklash lozim.

httpd.conf faylida direktivalar asosiy uchta bo'limga guruhlanadi:

Section 1. Global Environment – Apache ning umumiy ishini boshqaruvchi direktivalar bo'lib, Apache ning asosiy funksiyalariga ta'sir ko'rsatadi;

Section 2. Main Server Configuration – serverning asosiy parametrlarini aniqlovchi va so'rovlarga javob beruvchi direktivalar. Bu direktivalar virtual serverlar uchun ham aniqlanishi mumkin;

Section 3. Virtual Hosts – virtual serverlarni o'rnatish uchun.

Virtual serverlar (xostlar) – bu bir nechta web serverlarning turli IP-manzillar bilan bitta nusxadagi Apache dasturini ishlatishidir.

Virtual serverlarni qo'llash doirasi:

- turli tashkilotlar, firmalar va individual foydalanuvchilar uchun o'zlarining manzillari bilan alohida web serverlar yaratish;

- firma bo'limlari uchun virtual serverlarni tashkil etish, har bir bo'lim o'zining domen nomi va web serveriga ega bo'lsin;

- ikkita virtual server ko'rinishidagi umumiy foydalanuvchili va ichki web serverlarni tashkil etish;

- web serverni qayta ishlash va tekshirish uchun virtual serverlarni ishlatish.

Apache dasturining httpd.conf faylida barcha direktivalar bitta faylda httpd.conf ning o'zida saqlanadi.

Xampp paketining httpd.conf faylidagi direktivalar turli fayllarga bo'lib saqlangan bo'lib, ular httpd.conf fayliga Include yordamida chaqirib olinadi.

```
# xampplite specific settings
Include conf/extra/httpd-xampp.conf
# Server-pool management (MPM specific)
# Include conf/extra/httpd-mpm.conf
# Multi-language error messages
Include conf/extra/httpd-multilang-errordoc.conf
# Fancy directory listings
Include conf/extra/httpd-autoindex.conf
# Language settings
Include conf/extra/httpd-languages.conf
# User home directories
Include conf/extra/httpd-userdir.conf
```

```
# Real-time info on requests and configuration
Include conf/extra/httpd-info.conf
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
# Local access to the Apache HTTP Server Manual
# Include conf/extra/httpd-manual.conf
# Distributed authoring and versioning (WebDAV)
Include conf/extra/httpd-dav.conf
# Various default settings
Include conf/extra/httpd-default.conf
# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
```

Qoidaga asosan papkalar D:\xampp\htdocs\ (masalan, talaba) direktoriyasi ichida yaratiladi va shu papkaga Web sahifa fayllarida joylashtiriladi, so'ng brauzer yordamida <http://localhost/talaba/> murojaatini yozib Web sahifa ishga tushiriladi.

Bundan ham qulay yo'li mavjud bo'lib, Web sahifalar D:\xampp\htdocs\ da emas, balki boshqa papkada saqlanadi yoki saytning lokal versiyasiga www.talaba.local manzili bilan kiriladi. Buning uchun esa virtual xostlarni yaratish va sozlash lozim.

1. Virtual xost yaratish uchun avval xost joylashadigan papka yaratiladi:

D: diskning home papkasida (D:\home) talaba.local nomli papkani yaratamiz, uning ichida www va logs papkalarini yaratamiz.

Bu Windows fayllar tizimida quyidagi ko'rinishda bo'ladi:



2. D:\xampp\htdocs\ da bo'lgan barcha papkalarni D:/home/talaba.local/ www ga o'tkazamiz.

3. apache ning konfiguratsion fayliga o'zgartirish kiritamiz.

Matn muharririda D:\xampp\apache\conf\extra katalogida joylashgan joylashgan httpd-vhosts.conf faylini ochamiz.

httpd-vhosts.conf faylining tarkibiga hech qanday o'zgartirish kiritmaymiz, faqatgina yangi xost qo'shamiz (quyidagi kod nusxasini oling va fayldagi kodlarning oxiriga qo'ying):


```

NameVirtualHost *:80
<VirtualHost *:80>
    DocumentRoot "D:\xampplite\htdocs"
    ServerName localhost
</VirtualHost>
<VirtualHost *:80>
    ServerAdmin webmaster@talaba.local
    DocumentRoot "D:\home\talaba.local\www"
    ServerName talaba.local
    ServerAlias www.talaba.local
    ErrorLog "D:\home\talaba.local\logs\error.log"
    CustomLog "D:\home\talaba.local\logs\access.log" combined
<Directory "D:\home\talaba.local\www">
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
</VirtualHost>

```

Yuqoridagi boshqa rang bilan belgilangan virtual xost localhost orqali kirish uchun imkon yaratadi (XAMPP lokal sayti).

4. Brauzer virtual xostlarni aniqlash uchun C:\Windows\System32\drivers\etc\hosts manzilida joylashgan hosts fayliga ikkita qator qo'shish talab etiladi (server papkasida buning yorlig'ini ham yaratishni tavsiya etamiz):

127.0.0.1	talaba.local
127.0.0.1	www.talaba.local

5. Kerakli fayllarni D:\home\talaba.local\www manziliga ko'chiramiz yoki index.html faylini yaratamiz.

```

<html>
<head>
<title>Talabalar portali</title>
</head>
<body>
"Hello world!"
</body>
</html>

```

6. XAMPP boshqaruv paneliga (XAMPP Control Panel) kiramiz va Apacheni ishga tushiramiz. `www.talaba.local` manzili orqali ishlashi kerak bo'lgan saytimizni tekshiramiz. Agar barchasi to'g'ri bajarilgan bo'lsa, brauzerda "Hello world!" xabarini ko'rishimiz mumkin.

Shu kabi bir qancha virtual xostlarni qo'shish mumkin.

`httpd-vhosts.conf` fayliga qo'shilgan satrlarni yoritib chiqamiz:

- **ServerAdmin** `webmaster@talaba.local` - e-mail-Web-server administratori manzili;
- **DocumentRoot** "`D:\home\talaba.local\www`" – sayt orqali papkaga kirish yo'li;
- **ServerName** `talaba.local` – brauzerga kiritiladigan sayt nomi, ixtiyoriy qiymatni oladi, masalan, `talaba` yoki `talaba.local`;
- **ServerAlias** `www.talaba.local` – sayt ochilishi kerak bo'lgan manzil;
- **ErrorLog** "`D:\home\talaba.local\logs\error.log`" – xatolar jurnali;
- **CustomLog** "`D:\home\talaba.local\logs\access.log`" combined – saytdan foydalanish jurnali, kim va qachon saytga kirdi, qaysi brauzerdan kabilar yoritilgan.

II bob. O'ZGARUVCHILAR VA MA'LUMOTLAR TIPI

Bu bobda o'zgaruvchilar qanday yaratiladi va ularga qanday nom berish mumkin va mumkin emas; o'zgaruvchilar qiymatlarini tenglashtirish operatori; o'zgaruvchi tiplari kabilar bilan tanishishingiz mumkin.

2.1. PHP texnologiyasi haqida qisqacha ma'lumot

PHP tili (texnologiyasi) Internetda ishlash uchun mo'ljallangan bo'lib, Si tili kabi universal va aniq sintaksisga ega va o'zida Perl va Si tili imkoniyatlarini mujassamlashtirib olgan. Bu til yaqinda paydo bo'lgan bo'lsada (aniqrog'i uni interpretatori), u millionga yaqin serverlarga o'rnatilgan va o'rnatilishda davom etmoqda. PHP tili HTML fayliga dinamik tarkibni yaratuvchi PHP tili instruksiyalarini qo'yish imkoniyatini beradi. PHP dasturlari faqatgina PHP tilining konstruksiyalaridan yoki PHP va HTML tilining konstruksiyalaridan tashkil topgan bo'lishi mumkin.

PHP texnologiyasining asosiy vazifasi — Webda ishlovchilarga html-sahifalarni oson va tez hosil qilish, ularni dinamik holda ishlashi va o'zgartirish uchun imkoniyatlar yaratishdan iborat.

An'anaviy ravishda dasturlash haqidagi barcha kitoblar "Hello world!" dasturidan boshlanadi. Biz ham ushbu an'anani davom ettirgan holda misol keltiramiz:

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

Ssenariyni brauzerda ishga tushiring. Uni ishlayotganligiga guvoh bo'lasiz. Ko'rinib turganidek, PHP — dasturining strukturasini SI va Beysik aralashmasini eslatadi va HTML tarkibiga ham kiritiladi. Biroq biz yuqorida PHP tilidagi oddiy dastur misolini ko'rib chiqdik. Bu til haqidagi umumiy tasavvurni bermaydi. Bu bo'limda PHP tili konstruksiyalarini o'rganamiz, ya'ni til asoslarini o'rganamiz.

2.2. PHP deskriptorlarining ishlatilishi

PHP kodlari "<?" belgisi bilan boshlanib, "?>" belgisi bilan tugatiladi, xuddi HTML deskriptorlari "kichik" (<) belgisi bilan boshlanib va "katta" (>) belgisi bilan tugaganidek. Bu belgilar PHP deskriptorlari deb ataladi va ular WEB serverga PHP kodlari qayer-

da boshlanib, qayerda tugaganligi haqida xabar beradi. PHP kodlari bloki WEB serverda qayta ishlanib brauzerga uzatiladi.

Bu deskriptorlarning turli xil ko'rinishlari mavjud bo'lib, ko'rsatib o'tilgani qisqartirilgan formasidir.

PHP deskriptorlarning to'rt xil turi mavjud:

Qisqartirilgan turi:

<? ?>

Bu tur eng sodda tur bo'lib, SGML instruksiyasiga mos keladi.

XML turi:

<?php ?>

Deskriptorning bu turidan XML hujjatlarda foydalanish mumkin.

SCRIPT turi:

<SCRIPT LANGUAGE='php'> </SCRIPT>

Deskriptorning bu turi JavaScript va VBScript dasturlari bilan ishlaganlar uchun juda yaxshi tanish.

ASP turi:

<% %>

2.3. PHP operatorlari

PHP interpretatori bajarishi kerak bo'lgan biror amal PHP operatori yordamida ko'rsatiladi. Quyidagi misolda PHP ning ma'lumotlarni chiqazish operatorlarini ko'rib chiqamiz. PHP da ma'lumotlarni ekranga chiqazish operatori echo va print operatorlaridir.

echo konstruksiyasi judayam oddiy ishlaydi. U brauzer oynasiga qo'shtirnoq ichidagi yozuvni chiqazadi. Amal bajarilgandan so'ng albatta nuqta-vergul qo'yilishi kerak, agar nuqta-vergul unutilib qoldirilsa, u holda dasturda xatolik yuzaga keladi. Masalan,

```
<? echo "WEB sayt yaratamiz!"; ?>
```

Yoki

```
<html><head><title>Misol</title></head>  
<body><? echo "WEB sayt yaratamiz!"; ?></body>  
</html>
```

2.4. Izohlar

Izohlar foydalanuvchilarga dastur kodlarini o'qishda, dasturda ishlatilayotgan kod nimani anglatishini izohlashda, uning qismlarini

nomlashga, ya'ni bu kod bloki dasturning qaysi qismi ekanligini ajratishga yordam beradi. PHP sahifasida izohlarning bir necha xilini ishlatish mumkin. Masalan, C, C++, Unix dagi kabi ko'rinishlarini.

Izohlarning turlari:

```
// bir satrli izohlar
# bir satrli izohlar
/* ko'p satrli izohlar */
// va # belgisi bilan boshlanuvchi izohlar satrning oxirigacha tugatilishi kerak, ya'ni keyingi satrga o'tmasligi kerak.
```

2.1-misol. PHP da izohlarning ishlatilishi.

```
<? php
/* bu ko'p satrli izoh.
   Bu yerda bir nechta satrlarni yozish mumkin*/
echo "Xodimlar ro'yxati"; // bu C++ stilidagi
// bir satrli izoh
echo "Xodimlar ro'yxati"; # bu Unix stilidagi
# bir satrli izoh
?>
```

2.5. O'zgaruvchi tiplari

O'zgaruvchilar haqida siz maktab kurslaridan tushunchaga egasiz. O'zgaruvchi bu o'zida ma'lumotni, ya'ni biror-bir so'zni, biror-bir sonni saqlovchi belgi.

O'zgaruvchilar dastur bajarilayotgan vaqtda, ustida amallar bajarish mumkin bo'lgan ma'lumotlarni o'zida jamlagan xotira maydoni nomini ifodalaydi. PHP da o'zgaruvchilar \$(dollar) belgisi va o'zgaruvchi nomini yozish orqali yaratiladi. O'zgaruvchi nomi biror-bir harf yoki so'z bilan ifodalanadi, lekin son bilan ifodalanishi mumkin emas. Masalan: \$name, \$n, \$a2 va h.

O'zgaruvchiga biror qiymat berish uchun uni o'sha qiymatga tenglashtirish lozim. Tenglashtirish "=" amali yordamida amalga oshiriladi. Bir narsani yoddan chiqarmang, PHP da "=" amali aynan teng ma'nosini emas, balki tenglashtirish ma'nosini bildiradi. Agar bu ikki tushuncha aralashtirilib yuborilsa, dastur kodida xatoliklar yuzaga keladi.

Endi o'zgaruvchini biror-bir qiymatga tenglashtiramiz. Masalan, belgili qiymatga, u holda belgili qiymat "" ichida yozilishi kerak. Va har bir satr " ; " belgisi bilan yakunlanishi kerak.

```
<? $name="Lola"; ?>
```

Agar o'zgaruvchiga son qiymat beradigan bo'lsak, u holda "" belgisi qo'yilmasa ham bo'ladi.

```
<?
$name="Lola";
$age=22;
?>
```

2.2-misol. O'zgaruvchiga qiymat berish

```
<?php
$first = 'Matn'; // $first ga 'Matn' qiymatini tenglaymiz
$second = $first; // $second ga
// $first o'zgaruvchisi qiymatini tenglaymiz
$first = 'Yangi matn'; // $first ning qiymatini
// 'Yangi matn' ga almashtiramiz
echo " first nomli o'zgaruvchi " . " $first qiymatiga teng <br>";
// $first ning qiymatini chiqazamiz
echo " second nomli o'zgaruvchi " . " $second qiymatiga teng";
// $second ning qiymatini chiqazamiz
?>
```

Natija:

first nomli o'zgaruvchi yangi matn qiymatiga teng.

second nomli o'zgaruvchi Matn qiymatiga teng.

O'zgaruvchi tipi o'zida saqlanuvchi ma'lumotlar tipiga bog'liq.

PHP quyidagi ma'lumot tiplarini qo'llaydi:

Skalyar tiplar:

- boolean (mantiqiy)
- integer, int (butun)
- float, real, double (haqiqiy)
- string (satrli)

Aralash tiplar:

- array (massiv)
- object (obyekt)

Maxsus tiplar:

- resource (resurs)

NULL

Skalyar tipdagi o'zgaruvchilar

Skalyar tipdagi o'zgaruvchilarni ma'lumotlarni saqlovchi bir quti desak ham bo'ladi. U o'ziga sonli, belgili va matnli qiymatlarni

qabul qiladi. Skalyar tipdagi o'zgaruvchilar haqida yanada aniqroq tasavvur hosil qilish uchun quyidagi sxemani ko'rib chiqamiz.

\$a = 12

\$b = Bilim, bilim, bilim, bilim, bilim, bilim, bilim,
bilim, bilim, bilim, bilim, bilim, bilim, bilim,
bilim, bilim, bilim, bilim, bilim, bilim, bilim

Boolean (bulev yoki mantiqiy) tipi

Bu oddiy tip qiymatni rostligini aniqlaydi, bu tipdagi o'zgaruvchi faqatgina ikkita qiymat qabul qiladi – rost (TRUE) yoki yolg'on (FALSE).

Mantiqiy tipni aniqlash uchun, ikkita kalit so'z TRUE yoki FALSE ishlatiladi.

2.3-misol. Mantiqiy tip

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

integer yoki int (butun) tipi

Integer (butun) tipi butun sonlar to'plamida ($Z=\{..., -2, -1, 0, 1, 2, ...\}$) gi qiymatlar uchun ishlatiladi. Butun tiplar o'nlik, o'n oltilik va sakkizlik sanoq sistemalaridagi qiymatlarni “-” yoki “+” belgilarini ishlatgan holda ham qabul qiladi.

Agar sakkizlik sanoq sistemasi ishlatilgan bo'lsa, raqamdan avval 0(nol) sonini, o'n oltilik sanoq sistemasida esa raqamdan avval 0x qo'shib yozish kerak.

```
<?php  
$a = 1234; // o'nli sanoq sistemasidagi son  
$a = -123; // manfiy son  
$a = 0123; # sakkizlik sanoq sistemasidagi son  
$a = 0x1A; # o'n oltilik sanoq sistemasidagi son  
?>
```

Butun tipning eng katta qiymati 2 milliardga yaqin (bu 32 bit belgi). Belgisiz butun qiymatlarni PHP qabul qilmaydi.

Float, real, double (nuqta bilan ajratilgan son qiymati) tipi

Nuqta bilan ajratilgan son qiymati (haqiqiy son qiymati) tipi quyidagicha sintaksislarga ega.

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

Double(ikkilik qiymat) float, real tiplari bir-biriga o'xshash bo'lib, haqiqiy son qiymatlari uchun ishlatiladi.

String (satrli) tipi

Satr – bu belgilar to'plami. PHP da boshqa dasturlardagi kabi bayt bu turli 256 ta belgini qabul qiladi. String tipi satrli qiymatlar uchun ishlatiladi.

PHP da satr turli xil usullar bilan beriladi:

- bittalik qo'shtirnoq yordamida
- ikkitalik qo'shtirnoq yordamida
- heredoc – sintaksisi yordamida

Bittalik qo'shtirnoq

Satrlarni aniqlashni oddiy usuli – bu uni bittalik qo'shtirnoq "" ichiga olib yozish kerak. Satr ichida bittalik qo'shtirnoq ishlatish uchun boshqa dasturlash tillaridagi kabi, uning oldidan teskari chiziq belgisi qo'yilishi kerak. Agar teskari chiziq belgisini satrning oxirida bittalik qo'shtirnoq oldidan qo'yishga to'g'ri kelsa, u holda uni yana bittaga oshirib qo'yish kerak "\".

Agar bittalik qo'shtirnoq ichiga olingan satr ichida teskari chiziq "\" belgisi ixtiyoriy belgi oldidan ("\"" va "" dan farqli) uchrasa, u holda bu oddiy belgi sifatida qaraladi va ekranga shu ko'rinishda chiqadi. Shuning uchun teskari chiziq satr oxirida kelgan bo'lsa, unga yana bitta teskari chiziq qo'shib yozish kerak. PHP ning teskari chiziq bilan boshlanuvchi belgilar to'plami mavjud bo'lib, ular boshqaruvchi ketma-ketliklar deb ataladi va maxsus qiymatlarga ega(/n, :// va h). Shunga ko'ra, ikkita turli sintaksislar bilan farq qiluvchi o'zgaruvchilar va maxsus belgilar uchun boshqaruvchi ketma-ketliklar satrlarda bittalik qo'shtirnoq ichida uchrasa, ular qayta ishlanmaydi.

Ikkitalik qo'shtirnoq

PHP da ikkitalik qo'shtirnoq "" bilan chegaralangan satrlarda maxsus belgilar uchun bir qancha boshqaruvchi ketma-ketliklar ishlatiladi. Ulardan ba'zilari 2.1-jadvalda keltirilgan.

2.1-jadval

Satrlar uchun ishlatiladigan boshqaruvchi ketma-ketliklar

Boshqaruvchi ketma-ketliklar	
Ketma-ketlik	Qiymati
\n	Yangi satr (ASCII da LF yoki 0x0A (10))
\r	Kursorni qaytishi
\t	Gorizontal tabulyatsiya (ASCII da HT yoki 0x09 (9))
\\	Teskari qiyshiq chiziq
\\$	Dollar belgisi
\"	Ikkitalik qo'shtirnoq

Heredoc sintaksisi

Satrlar katta hajmdagi ma'lumotlar (jadvallar, satrlar, murojaatlar) ham bo'lishi mumkin. Bunday satrlarni echo operatoridan foydalanib ekranga chiqazish bir qancha muammolar keltirishi mumkin. Ushbu holatlarda satriy o'zgaruvchilarni heredoc sintaksisi usulidan foydalanib aniqlash mumkin. Bu usulda satrlar <<< markeri bilan boshlanib, so'ng identifikator so'zi yoziladi va satr yana shu identifikator so'zi bilan tugatiladi.

Sintaksisi:

echo <<<MATN

... ..

MATN;

Heredoc matni qo'shtirnoq ichida joylashgan matn kabi bo'ladi va heredoc sintaksisi ishlatilganda qo'shtirnoq ishlatishga hojat yo'q. Heredoc matni ichida o'zgaruvchilarni ham berish mumkin va bu holda heredoc ichidagi o'zgaruvchilar ham qayta ishlanadi.

2.4-misol. heredoc sintaksisining ishlatilishi

```
<?php
$str = <<<EEE
<table cellpadding="1" width="100%" cellspacing="1" border="0" >
<tr><td> <div class="body_txt">
<a href="#"><li>$fak fakultetining tarixi</a> </div>
```

```
</td></tr>
</table>
EEE; // EEE – bu yerda identifikator.
?>
```

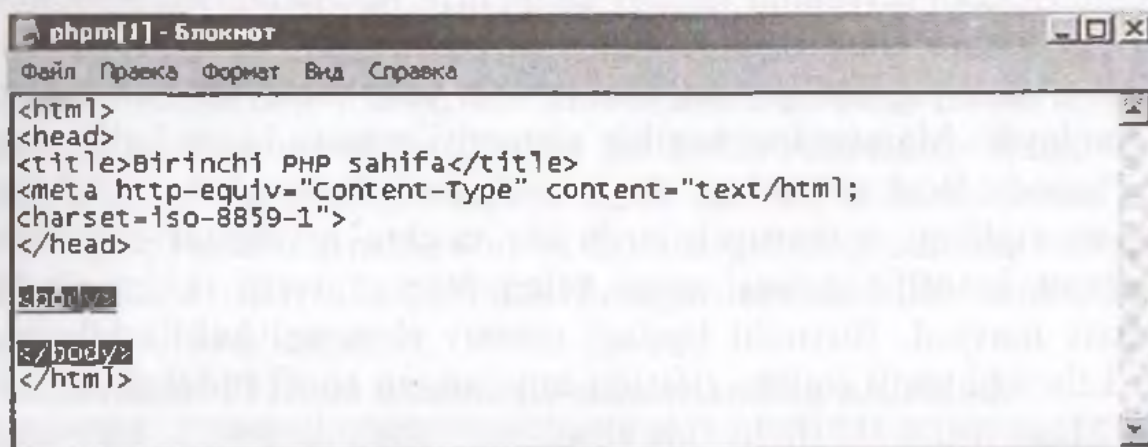
Skalyar tipdagi o'zgaruvchilarning boolean, integer, float, string turlarini ko'rib o'tdik.

2.5-misol. O'zgaruvchilarning qiymatlarini chiqazish

```
<? $name="Lola"; $age=22;
$price=2.11; ?>
```

Bu yerda, \$name o'zgaruvchisi string tipdagi, \$age o'zgaruvchisi integer tipdagi, \$price o'zgaruvchisi esa double tipdagi qiymatlarni qabul qiladi.

Endi sahifani saqlab, brauzerda ko'ramiz. Sahifada hech qanday yozuv yoki ma'lumot yo'q. Ichki (html) kodini ko'radigan bo'lsak.



Biz yozgan PHP kodlar brauzerda ham, ichki kodda ham chiqmadi. Nimaga? Chunki biz faqatgina o'zgaruvchilarga qiymat berdik xolos, brauzerga chiqazishga esa buyruq bermadik.

echo operatoridan foydalanib, o'zgaruvchilarning qiymatlarini brauzerga chiqazamiz.

```
<? $name="Lola";
$age=22;
echo "$name"; ?>
```

Fakultet so'zini ham brauzerga chiqazmoqchimiz, u holda

```
<? $name="Lola";
$age=22;
echo "$name";
echo "Fakultet"; ?>
```


Bu yerda qiymatlar ketma-ket holda chiqdi. Ularni ajratish uchun bizga tanish bo'lgan
 tegidan foydalanishimiz mumkin.

```
<? $name="Lola";  
$age=22;  
echo "$name";  
echo "<br>Fakultet"; ?>
```

Aralash tipdagi o'zgaruvchilar

Array tipi (Massivlar)

Skalyar tipdagi o'zgaruvchi — bu yagona qiymat saqlanadigan katakcha xotirasining nomi. Massiv tipidagi o'zgaruvchi esa qiymatlar to'plami saqlanadigan katakcha xotirasining nomi. Boshqacha qilib aytganda, bir nechta tipdagi ma'lumolarning bir tipdagi o'zgaruvchida saqlanishi array (massiv) deb ataladi. Massivda saqlanayotgan qiymatlar massiv elementlari deb ataladi. Bitta massiv bir nechta elementlardan tashkil topgan bo'lib, har bitta element o'zida yagona qiymat (matn, son yoki biror boshqa massiv) ni saqlaydi. Massivning har bir elementi massiv bilan kalit orqali bog'lanadi. Boshqa tipdagi o'zgaruvchilarda bitta qiymat saqlansa, massiv tipidagi o'zgaruvchilarda bir nechta qiymatlar saqlanadi. Element identifikatsiyasi usuli bilan farq qiluvchi ikkita tipdagi massiv mavjud. Birinchi tipdagi massiv elementi kalitlari ketma-ket keluvchi sonli indeks sifatida aniqlangan sonli indeksli massivlar. Massivning ikkinchi tipi assotsiativ shaklda bo'lib, elementga murojaat etish uchun qiymat bilan mantiqiy bog'langan kalitlar ishlatiladi. O'zida boshqa massivlarni saqlagan massiv ko'p o'lchovli massiv deb ataladi.

Massivlar array() konstruksiyasi yordamida e'lon qilinadi.

Sintaksisi:

```
array ([kalit] => qiymat,  
[kalit1] => qiymat1, ... )
```

array() massivi juftlik kalit=> qiymat parametrlarini vergul bilan ajratib qabul qiladi. => belgisi qiymat va uning kaliti orasidagi bog'lanishni o'rnatadi. Kalit tipi butun va satrli, qiymat esa PHP ning ixtiyoriy tipida bo'lishi mumkin.

Masalan, \$name=array("Lola", "Karim", "Vasila", "Akmal", "Botir");

Sonli indeksli massivlar

\$a o'zgaruvchisini nashr qilishga bersak, bitta qiymatni ekranga chiqazadi. \$name o'zgaruvchisi massiv bo'lganligi sababli, kalitni ko'rsatishimiz kerak. Ya'ni Akmalni ekranga chiqazmoqchi bo'lsak, \$name[3] deb kalitini ko'rsatishimiz kerak. Sonli indekslangan massivlar noldan boshlanadi.

```
$name=array (0=>"Lola", 1=>"Karim", 2=>"Vasila",  
3=>"Akmal", 4=>"Botir");
```

```
yoki $name=array("Lola", "Karim", "Vasila", "Akmal",  
"Botir");
```

```
yoki $name[0]= "Lola";$name[1]="Karim";$name[2]= "Vasi-  
la"; $name[3]= "Akmal";$name[4]= "Botir";
```

```
Yoki $name[ ]= "Lola";$name[ ]="Karim";$name[ ]= "Vasila";  
$name[ ]= "Akmal";$name[ ]= "Botir";
```

Assotsiativ massivlar

Ko'pgina dasturlash tillarida massiv indeksleri 0 va 1 dan boshlanuvchi sonlardan iborat. Bu dasturlash tillaridan farqli ravishda PHP assotsiativ massivlarni ham qo'llaydi. Assotsiativ massivlarni sonli indekslangan massivlardan farqi assotsiativ massivlarda kalit ixtiyoriy qiymatlar (matn, son va h.k.)ni qabul qiladi. Bu massivda massiv elementlari kalit sifatida berilgan sonlar bilan emas, balki satrlar bilan bog'lanadi.

2.6-misol. PHP da assotsiativ massivlarning ishlatilishi

```
<? $poytaxt["Uzbekiston"]="Toshkent";  
$poytaxt["Rossiya"]="Moskva";  
$poytaxt["Fransiya"]="Parij";  
echo $poytaxt["Fransiya"]; ?>  
Natija: Parij
```

Ko'p o'lchovli massiv

Massiv kalit va qiymatlarni o'zida mujassamlashtirgan oddiy ro'yxatdan iborat bo'lishi shart emas, balki massiv yacheykasi o'zida boshqa massivlarni ham saqlashi mumkin. Bunday massivlar ko'p o'lchovli massivlar deyiladi. Bu massivda aralash (sonli va assotsiativ) indekslashga ruxsat beriladi.

Ko'p o'lchovli massiv sifatida, ikki o'lchovli massivlarni ham yaratish mumkin. Ikki o'lchovli massiv kengligi va balandligi satrlar va ustunlarga akslanuvchi matritsa yoki jadval ko'rinishida bo'ladi.

2.7-misol. Ko'p o'lchovli massivlarning ishlatilishi

\$auto massivini qaraydigan bo'lsak, Nexia indeksli massiv qiymati boshqa bir massivga teng. Demak, bu ko'p o'lchovli massivning PHP kodini yozib chiqamiz.

```
<? $auto["nexia"] = array("yili" => "1999", "rangi" => "oq");  
$auto["damas"] = array("yili" => "2003", "rangi" => "qora");  
$auto["matiz"] = array("yili" => "2005", "rangi" => "oq");  
$auto["tiko"] = array("yili" => "1999", "rangi" => "qizil");  
echo "Matiz avtomobilini ishlab chiqarilgan yili - ". $auto["matiz"]  
["yili"]; ?>  
Natija.  
Matiz avtomobilini ishlab chiqarilgan yili - 2005
```

Massiv kalitlari

Agar kalit berilmagan bo'lsa, u holda kalit o'rniga kalitning eng katta qiymatiga 1 sonini qo'shib yozib ketiladi. Kalit biror-bir qiymatga teng bo'lsa, u holda o'sha qiymat o'zi yoziladi. PHP ning 4.3.0 dan keyingi versiyalarida kalitning eng katta qiymati manfiy son bo'lsa, massivning keyingi qiymati nol(0) deb olinadi.

Agar kalit o'rnida TRUE yoki FALSE qiymatlari ishlatilsa, u holda kalit tipi integer bo'lgan 0 yoki 1 qiymatni beradi. Kalit o'rnida NULL ishlatilsa, bo'sh satrni beradi. Kalit o'rnida bo'sh satr ham ishlatilishi mumkin, faqat uni qo'shtirnoq ichiga olib yozish kerak yoki alohida yozilganda bo'sh to'rtburchak qavs ishlatiladi. Kalit o'rnida obyekt va massivlar ishlatilmaydi.

2.8-misol. PHP da massivlarning ishlatilishi

```
<? // $arr va $arr1 massivlari ekvivalent  
$arr = array(-3 => 43, 32, 56, 86, 97, "b" => 12);  
$arr1 = array(-3 => 43, 0 => 32, 1 => 56, 2 => 86, 3 => 97, "b" => 12,  
"s" => true, " " => 28);  
Echo $arr1[null];?>
```

Agar yangi kalit va yangi qiymat berilsa (\$name["new_key"]="new_value"), u holda massivga yangi element qo'shilgan hisoblanadi. Agar biz kalitni ko'rsatmasak, faqatgina qiymatga tenglasak, \$name[]="new_value", u holda massivning yangi elementi sonli kalit bo'lib, mavjud kalitning eng katta qiymatidan bir qiymatga oshgan bo'ladi.

```
<? $name=array(2=>5, 3=>6, 4=>15);  
$name[] = value1;
```

```
/* qiymati valuel kaliti 5 (bizda kalitning eng katta qiymati 4 bo'lganligi uchun)ga teng massiv kiritdik */  
echo $name; ?>
```

Massivning elementini o'zgartirish uchun, kalitini yangi qiymatga tenglashtirish yetarli. Massiv kalitini o'zgartirish mumkin emas, faqatgina element (kalit/qiymat juftligini)ni o'chirib yuborish va yangisini kiritish mumkin. Massiv elementini o'chirish uchun, `unset()` funksiyasidan foydalaniladi.

```
<?php  
$name=array(2=>5, 3=>6,4=>15); $name[] = valuel;  
/* qiymati valuel kaliti 5 (bizda kalitning eng katta qiymati 4 bo'lganligi uchun)ga teng massiv kiritdik */  
echo $name; $name[] = "Sobir";  
/*kaliti(indeksi)5 ga teng. $name[5] = "Sobir" ga ekvivalent element yaratildi */ Echo $name.<br>; $name["starosta"] = "Komila";  
/* Kaliti "starosta", qiymati "Komila" bo'lgan yangi massiv elementini yaratadi */  
unset($name[4]);  
// massivdan kaliti 4 ga teng bo'lgan elementni o'chiradi  
Echo $name.<br>;  
unset ($name); // massivni to'liq o'chiradi  
Echo $name; ?>
```

Object tipi (obyektlar)

Obyektlar — obyektga yo'naltirilgan dasturlashdan (OYD) olingan ma'lumotlar tipi. OYD prinsipiga asoslanib, klass — bu xususiyati va ishlash metodi aniqlangan obyektlar to'plami, obyekt esa klassdan olingan bir nusxa.

Biror-bir vazifani bajaruvchi amallardan tashkil topgan klassni e'lon qilamiz. Klass e'lon qilib bo'lingandan so'ng, klassni a'zosi bo'lgan alohida aniq element — obyektни yaratamiz. Bu jarayon klassning nusxasini yaratish deb nomlanadi. Obyekt new kalit so'zi yordamida yaratiladi. Buning uchun, obyekt qaysi klassni nusxasi ekanligini va barcha parametrlarni ko'rsatish kerak. Obyektlar bilan ishlashda klasslarni chaqirish uchun `->` operatoridan foydalaniladi.

Misol uchun, dasturchilar klassi — bu dastur tuza oladigan, kompyuterda ishlay oladigan va bundan tashqari boshqa insonlar kabi ism va familiyaga ega insonlar klassi. Endi, agar birorta aniq bir dasturchi, masalan, Odil Karimovni oladigan bo'lsak, u holda

u boshqa dasturchilar kabi dastur tuza oladigan, ism-familiyaga va boshqa xususiyatlarga ega dasturchilar klassining obyekti hisoblanadi.

2.9-misol. PHP da obyektlar.

```
<?php
//insonlar klassini e'lon qilamiz class person
{ function prog_learn($name)
  { echo "$name PHP da dastur tuza oladi"; }
  function lang_learn($name1)
    {echo "$name1 ingliz tilida gaplasha oladi";} }
// klassni obyektlarini yaratamiz
$a = new person();
$a->prog_learn("Lola")
$b=$a->lang_learn("Karim");
$c= $a-> prog_learn(); ?>
Klass har bitta obyekt yaratilganda quyidagi natijani chiqaradi:
Lola PHP da dastur tuza oladi
Karim ingliz tilida gaplasha oladi
PHP da dastur tuza oladi
```

Maxsus tipdagi o'zgaruvchilar

resource tipi (resurslar)

Resurs – bu tashqi resursga (masalan, ma'lumotlar bazasi bilan bog'lanishga) murojaatni ta'minlovchi maxsus o'zgaruvchi. Resurslar maxsus funksiyalar (masalan, `mysql_connect()`, `pdf_new()` va boshqalar) bilan yaratiladi va ishlatiladi.

Null tipi

Maxsus qiymat `NULL` o'zgaruvchi qiymatga ega emasligini anglatadi.

O'zgaruvchi `NULL` hisoblanadi, agar:

u `NULL` konstantasiga tenglashtirilgan bo'lsa (`$var = NULL`);

u hali biror-bir qiymatga tenglashtirilmagan bo'lsa;

`unset()` yordamida o'chirilgan bo'lsa.

Faqatgina bitta registrga bog'liq bo'lmagan `NULL` kalit so'zli `NULL` tipidagi qiymat mavjud.

2.6. Identifikatorlar

Umumiy termin "identifikator" foydalanuvchi tomonidan aniqlangan o'zgaruvchilar, funksiyalar va boshqa obyektlar uchun qo'llaniladi. PHP identifikatorlari bir nechta shartlarni qoniqtirishi

kerak: identifikatorlar bir yoki bir nechta belgilardan tashkil topgan bo'lib, harflar yoki ostki chiziq belgisi bilan boshlanadi; identifikatorlar faqatgina harflar, sonlar, ostki chiziq belgilari va 127 dan 255 gacha bo'lgan ASCII kodi belgilaridan iborat bo'lishi mumkin.

Masalan:

Ruxsat etilgan identifikatorlar	Ruxsat etilmagan identifikatorlar
my_f	Thi&th
Siz	!coun
_somet	4wa

Identifikatorlarda belgilar registrga sezgir. Misol uchun, \$rec nomli o'zgaruvchi \$Rec nomli o'zgaruvchidan farq qiladi. Identifikator uzunligi chegaralanmagan. Identifikator PHP ning birorta standart kalit so'ziga mos kelishi mumkin emas.

2.7. O'zgaruvchilar bilan ishlovchi funksiyalar

PHP ning o'zgaruvchilar bilan ishlovchi funksiyalari bilan tanishamiz. Bu funksiyalar kutubxonasi o'zgaruvchilarni tekshirish va ularni turli usullar bilan o'zgartirish uchun ishlatiladi.

O'zgaruvchilarning tiplarini tekshirish va o'rnatish

Sintaksisi:

string gettype(mixed var);

int settype(string var, string type);

gettype() funksiyasi unga uzatilgan o'zgaruvchining tipini aniqlab, tip nomini yoki agar o'zgaruvchi tipi standart tiplardan(integer, double, string, array yoki object) birortasi ham bo'lmasa, "unknown type" ni qaytaradi.

settype() funksiyasiga parametrlar sifatida o'zgaruvchi va standart tiplardan biri beriladi. Bu funksiya parametr sifatida berilgan o'zgaruvchi tipini parametr sifatida berilgan yangi tipga o'zgartiradi.

Misol:

```
<? $a = 56;
echo gettype($a)."<br>";
settype ($a, "double");
echo gettype($a)."<br>"; ?>
```

Natija:

integer

gettype() funksiyasiga murojaat etganda, \$a o'zgaruvchisining tipi integer ekanligini chiqazadi. settype() funksiyasiga murojaat etgandan so'ng, uning tipi double ga o'zgaradi.

PHP ning tiplarni tekshirish uchun bir qancha funksiyalari mavjud. Bu funksiyalarning har biri parametr sifatida o'zgaruvchini qabul qiladi va

true yoki false qiymatlarini qaytaradi.

- is_array()
- is_double(), is_float(), is_real() (bir xil funksiya)
- is_long(), is_int(), is_integer() (bir xil funksiya)
- is_string()
- is_object()

O'zgaruvchi holatini tekshirish

PHP ning o'zgaruvchi holatini tekshiruvchi bir qancha funksiyalari mavjud. Ulardan biri — isset().

Sintaksisi:

```
int isset(mixed var);
```

Bu funksiya parametr sifatida o'zgaruvchi nomini oladi va natija sifatida agar o'zgaruvchi mavjud bo'lsa, true, aks holda false ni qaytaradi.

O'zgaruvchini unset() funksiyasi yordamida o'chirish mumkin. Uning sintaksisi quyidagicha:

```
int unset(mixed var);
```

bu funksiya o'zgaruvchini o'chiradi va natija sifatida true qiymatini qaytaradi.

empty() funksiyasi qiymati bo'sh va nol bo'lmagan o'zgaruvchining mavjudligini aniqlaydi va mos holda true yoki false qiymatlarini qaytaradi.

Sintaksisi:

```
int empty(mixed var);
```

Ushbu uchta funksiyaga misol ko'ramiz. Vaqtinchalik yaratilgan sahifaga ushbu kodlarni kiritamiz:

```
<? echo isset($bor);  
echo isset($yuq);  
echo empty($bor);  
echo empty($yuq); ?>
```

isset() funksiyasi \$bor o'zgaruvchisi uchun forma maydonida qiymat kiritilgan yoki kiritilmaganligiga bog'liq bo'lmagan holda, true qiymatini qaytaradi. empty() funksiyasi esa kiritilgan qiymatga bog'liq holda natijani qaytaradi.

\$yuq o'zgaruvchisi mavjud emas, shuning uchun bu o'zgaruvchi uchun isset() funksiyasi false qiymatini, empty() funksiyasi esa — true ni qaytaradi.

Ko'rib o'tilgan funksiyalar formadagi mavjud maydonlar foydalanuvchi tomonidan to'ldirilganligini tekshirish uchun ishlatiladi.

O'zgaruvchilarni qayta interpretatsiyalash

Sintaksisi:

int intval(mixed var);

double doubleval(mixed var);

string strval(mixed var);

Ushbu har bir funksiya parametr sifatida o'zgaruvchini qabul qiladi va mos tipga o'zgartirilgan o'zgaruvchi qiymatini qaytaradi.

III bob. AMALLAR

Bu bobda satrlar bilan ishlash (satrlarni satrlar bilan birlashtirish, o'zgaruvchilar bilan satrlarni birlashtirish), arifmetik amallar (o'zgaruvchi qiymatlari ustida qo'shish, ayirish, ko'paytirish va bo'lish amallarini to'g'ri bajarish), mantiqiy amallar, tenglik amallari, munosabat amallari, xatolarni yashirish amallari bilan tanishish mumkin.

3.1. Satrlar bilan ishlash amallari

O'zgaruvchilar va literallar. O'zgaruvchi ma'lumotlar belgisi. \$name — o'zgaruvchi yoki biror-bir ma'lumotni o'zida saqlovchi belgi. Satrlar ham ma'lumotlar. "talaba" — bu literal, u qanday bo'lsa shunday qabul qilinadi.

Satrlarni birlashtirish

PHP da satrlar bilan ishlash uchun eng qulayi konkatenatsiya (satrlarni birlashtirish) amali. Konkatenatsiya ikki yoki undan ortiq obyektlarni bir butun qilib birlashtirish degan ma'noni bildiradi. Satrlar bilan ishlash uchun ikkita amal mavjud: konkatenatsiya amali(.) va konkatenatsiya bilan tenglik amali(=).

3.1-jadval

Satrlar bilan ishlash amali

Belgisi	Nomi	Misol	Natija
.	Konkatenatsiya	\$a="abc"."def"	\$a o'zgaruvchisi abc va deflarning birlashtirish natijasiga teng.
=	Konkatenatsiya bilan tenglik	\$a.="ghijkl"	\$a o'zgaruvchisi uning mavjud qiymati bilan "ghijkl" satrini birlashtirish natijasiga teng.

Konkatenatsiya amali yordamida satrlar bilan satrlarni, o'zgaruvchilar bilan satrlarni birlashtirish mumkin. Bu amaldan brauzerga matnlarni chiqazishda tez-tez foydalaniladi va echo buyrug'ini bir necha marotaba berishga zarurat tug'ilmaydi.

Satrlar bilan ishlash amallariga misollar keltiramiz:

3.1-misol. Satrlar bilan ishlash.

```
<? $a="Olma"." va Nok"; // $a "Olma va Nok" satrli qiymatga teng  
$a.=" mevalar"; // $a "Olma va Nok mevalar" satrli qiymatga teng ?>
```

Sonli o'zgaruvchilar ustida ham konkatenatsiya amalini bajarish mumkin.

3.2-misol. Satrlar bilan ishlashda konkatenatsiya amali

```
<? $a=39; $b=10;  
$n=$a.$b;  
echo $n; ?>  
Natija: 3910
```

Demak, (.) konkatenatsiya amali \$a va \$b o'zgaruvchilarni qiymatlarini bir-biriga qo'shib qo'ymaydi, balki ularni satriy kattalik sifatida ketma-ket joylashtiradi.

Endi o'zgaruvchi qiymatlari, matnlarni va HTML teglarini birgalikdagi kombinatsiyasini chiqazishini ko'ramiz.

3.3-misol. Satrlar bilan ishlash

```
<? $name="Lola"; $age=22;  
echo "<br>$name Fizika-matematika  
a fakultetida o'qiydi. <br>Uni yoshi $age da." ?>  
Natija:  
Lola Fizika-matematika fakultetida o'qiydi.  
Uni yoshi 22 da.
```

O'zgaruvchilarga ba'zi hollarda qiymat berilayotganda quyidagi kabilarga duch kelish mumkin.

```
<? $a= "Lola "Ulug'bek stipendiyasi" sovrindori";  
echo $a; ?>
```

Agar o'zgaruvchiga shu tartibda qiymat beriladigan bo'lsa, u holda brauzerda xatolik haqida xabar chiqaziladi, ya'ni ikkitalik qo'shtirnoq ichida ikkitalik qo'shtirnoq bu holatda ishlatilmaydi. Agar ikkinchi qo'shtirnoqning ochilishi va yopilishi oldidan \ belgisi qo'llanilsa, u holda xatolik yuzaga kelmaydi.

```
<? $a= "Lola \"Ulug'bek stipendiyasi\" sovrindori"; echo $a; ?>
```

PHP sahifada HTML teglarining qo'llanilishini ko'rib chiqamiz.

```
<? echo "<p align=\"center\">Fizika-matematika fakulteti</p>"; ?>
```

HTML tegini PHP sahifada bunday holda qo'llashimiz xatolikka olib keladi. Bu yerda yuqoridagi singari ikkinchi qo'shtirnoq ochilishi va yopilishi oldidan \ belgisi ishlatilsa bo'ladi, lekin uni o'rniga quyidagi holda yozish ham mumkin.

```
<? echo "<p align='center'>Fizika-matematika fakulteti</p>"; ?>
```


3.2. Arifmetik amallar

Biz ikkita amal bilan tanishmiz: (=) tenglik amali va (.) satrlarni birlashtirish amali. Endi arifmetik amallar bilan tanishib chiqamiz. O'zgaruvchilar ustida barcha arifmetik amallarni bajarish mumkin.

3.2-jadval

Arifmetik amallar

Belgisi	Ko'rinishi	Nomi
+	$a+b$	qo'shish
-	$a-b$	ayirish
*	$a*b$	ko'paytirish
/	a/b	bo'lish
%	$a\%b$	modul olish

Qo'shish va ayirish amallari oddiy bajariladi. Ularni natijasi a va b o'zgaruvchilarda saqlanayotgan qiymatlarning yig'indisi va ayirmasiga teng. Ayirish amalidan unar amal sifatida ham foydalanilsa bo'ladi. Bunda o'zgaruvchiga manfiy qiymat beriladi, $a=-1$.

Ko'paytirish va bo'lish amallari ham oddiy ishlatiladi. E'tibor bering, belgilar an'anaviy matematikadagi belgilarni, ya'ni yulduzcha ko'paytirish amali belgisini va egilgan tayoqcha bo'lish amali belgisini beradi.

3.4-misol. Modul olish amali.

```
<? $a=39; $b=10;  
$natija=$a%$b;  
/* Bu yerda $natija o'zgaruvchisi 39 ni 10  
a bo'lingandagi qoldiq 9 ga teng bo'ladi. */ ?>  
Natija: 9
```

3.5-misol. Arifmetik amallarning ishlatilishi

```
<? $a=10; $b=17;  
$qo'shish=$a+$b; $ayirish=$a-$b;  
echo $qo'shish." - $a va $b ning yig'indisi.<br>";  
echo $ayirish." - $a va $b ning ayirmasi. <br>";  
$a=20; $b=5;  
$kupay=$a*$b; $bo'lish=$a/$b;  
echo $ko'pay."- $a va $b ni ko'paytmasi.<br>";  
echo $bo'lish."- $a ni $b ga bo'linmasi."; ?>  
Natija:  
27 - 10 va 17 ning yig'indisi.
```

- 7 - 10 va 17 ning ayirmasi.
- 100 - 20 va 5 ni ko'paytmasi.
- 4 - 20 ni 5 ga bo'linmasi.

Arifmetik amallarni tenglashtirish amali bilan birga boshqacha ko'rinishda ham ishlatish mumkin. Masalan,

$\$b = 12 + (\$a = 4);$

Bu holda $\$b$ o'zgaruvchisi 16 ga teng bo'ladi. Chunki, $\$a$ o'zgaruvchisi 4 qiymatni qabul qilyapti va 12 ga qo'shilyapti, shuning uchun $12 + 4 = 16$ ga teng.

Oddiy tenglashtirish amallaridan tashqari kombinatsiyalangan tenglashtirish amallari yoki qiymat uzatish amallari ham mavjud.

Qiymat uzatish amali har bir arifmetik amal va konkatenatsiya amali uchun ham mavjud.

3.3-jadval

Qiymat uzatish amali

Belgisi	Ishlatilishi	Ekvivalenti
$+=$	$\$a += \b	$\$a = \$a + \$b$
$-=$	$\$a -= \b	$\$a = \$a - \$b$
$*=$	$\$a *= \b	$\$a = \$a * \$b$
$/=$	$\$a /= \b	$\$a = \$a / \$b$
$\% =$	$\$a \% = \b	$\$a = \$a \% \$b$
$. =$	$\$a . = \b	$\$a = \$a . \$b$

Masalan, $\$a += 8$ yozuvi $\$a = \$a + 8$ yozuviga ekvivalent.

inkrement(++) va dekrement(--) yordamchi amallari dasturda o'zgaruvchi qiymatini 1 ga kamaytirish yoki oshirish uchun xizmat qiladi.

3.4-jadval

Inkrement va dekrement amallari

Misol	Nomi	Natija
$++\$a, \$a++$	Inkrement	$\$a$ o'zgaruvchisini 1 ga oshiradi
$--\$a, \$a--$	dekrement	$\$a$ o'zgaruvchisini 1 ga kamaytiradi

Prefiksli va suffiksli inkrement(++) va dekrement(--) amallari $+=$ va $-=$ amallariga o'xshash. Masalan,

`<? $a=4; echo ++$a; ?>`

Natija:5

Ikkinchi satrda prefiksli inkrement amali ishlatilyapti, chunki \$a dan oldin (++) belgisi mavjud. Natijada \$a ning qiymati 1 ga oshiriladi va oshirilgan natija chiqaziladi. Bu holda, \$a ning qiymati 5 ga oshiriladi va 5 ekranga chiqaziladi, ya'ni \$a o'zgaruvchining qiymati avval oshirilib, keyin natijaga beriladi.

Agar (++) belgisi \$a dan keyin ishlatilsa, demak suffiksli inkrement amali ishlatilyapti, u holda natija boshqacha bo'ladi. Masalan,

```
<? $a=4; echo $a++; ?>
```

Natija:4

Bu holda amal teskari tartibda bajariladi. Avval \$a qiymatni qabul qiladi va natijaga chiqazadi, shundan so'ng uning qiymati 1 taga oshiriladi. Bunda natijaga 4 soni chiqaziladi, keyin \$a ning qiymati 5 ga teng bo'ladi.

Inkrement va dekrement amallari bir-biriga o'xshash. Faqatgina, prefiksli va suffiksli inkrement amalida \$a ning qiymati 1 taga oshirilsa, dekrement amalida esa \$a ning qiymati 1 taga kamaytiriladi.

3.3. Mantiqiy binar amallar

Mantiqiy amallar PHP da ixtiyoriy dastur tuzish jarayonida, o'zgaruvchi qiymatlari bilan bog'liq bo'lgan hollarda muhim rol o'ynaydi. Mantiqiy amallar dasturda buyruqlar bajarilish tartibini boshqarish imkoniyatini beradi va boshqaruvchi konstruksiyalar (shartli if operatori va for, while sikllari) bilan tez-tez ishlatiladi.

Mantiqiy shartlarning natijalarini birlashtirish uchun mantiqiy amallar ishlatiladi. Misol uchun \$a o'zgaruvchisi 0 va 100 orasidagi qiymatlarni qabul qiladi deylik. Bu holda AND(VA) amalini ishlatib, \$a>=1 va \$a<=100 shartlarni tekshiramiz.

```
$a>=1 && $a<=100
```

PHP texnologiyasi AND(VA), OR(YOKI), XOR(yoki mustasno) va NOT(EMAS) kabi mantiqiy amallarni ishlatadi.

3.5-jadval

Mantiqiy amallar

Belgisi	nomi	ishlatilishi	natija
&&	Va	\$a&&\$b	\$a va \$b o'zgaruvchilarning ikkalasi ham rost qiymat qabul qilsa, rost, aks holda yolg'on
AND	Va	\$a AND \$b	&& kabi mantiqiy amal

	Yoki	$\$a \parallel \b	$\$a$ yoki $\$b$ o'zgaruvchilarning ikkalasidan birortasi rost qiymat qabul qilsa, rost, aks holda yolg'on
OR	yoki	$\$a \text{ OR } \b	kabi mantiqiy amal
XOR	yoki	$\$a \text{ XOR } \b	o'zgaruvchilarning ikkalasidan faqatgina bit-tasi rost bo'lsa, rost
!	Emas	$!\$a$	agar $\$a$ yolg'on bo'lsa, natija rost va aksincha

3.4. Tenglik amali

Tenglik amali = =(ikkita teng belgisi) ikkita qiymatni bir-biriga taqqoslaydi va ularning tengligini tekshiradi. Masalan,

$\$a = \b ;

$\$a$ va $\$b$ o'zgaruvchilarda saqlanayotgan qiymatlarning tengligini tekshirish uchun ishlatiladi. Agar ikkala o'zgaruvchining qiymati bir-biriga teng bo'lsa, natija true ni, aks holda false ni beradi.

Bu amalni tenglashtirish amali bilan adashtirish mumkin. Bu vaqtda dasturda xatolik yuzaga kelganligi haqida xabar chiqmasligi mumkin, lekin istalgan natijaga erishib bo'lmaydi. Umumiy hollarda nol bo'lmagan qiymat true ni, nol qiymatlar esa — false ni beradi. Quyidagi holatni ko'rib chiqamiz:

$\$a = 5$; $\$b = 7$;

$\$a = \b ni tekshiradigan bo'lsak, natija true ni beradi. Nima uchun?

3.6-jadval

Tenglik amallari

Amal belgisi	Nomi	Ishlatilishi	Natija
$=$	Teng	$\$a = \b	$\$a$ va $\$b$ teng bo'lsa, rost
$==$	Aynan teng	$\$a == \b	$\$a$ va $\$b$ aynan teng (teng va bir xil tipli) bo'lsa, rost
$!=$	Teng emas	$\$a != \b	$\$a$ va $\$b$ teng bo'lmasa, rost
0	Teng emas	$\$a 0 \b	$\$a$ va $\$b$ teng bo'lmasa, rost

Chunki $\$a = \b qiymatga ega, ya'ni o'ng tomondagi operandning qiymati ayni vaqtda 7 ga teng, bu esa nol bo'lmagan qiymat. Shuning uchun ifoda natijasi true ni beradi.

Agar $\$a == \b ifodani tekshiradigan bo'lsak, bu holda natija false ni beradi, chunki $\$a$ va $\$b$ ning qiymati bir-biriga teng emas.

Demak, $=$ va $=$ belgilari ikki xil natija berdi. Shuning uchun dastur kodlari yozayotgan vaqtda qaysi taqqoslash amali kerakligini aniqlab, keyin bu belgilarni ishlatish kerak.

3.5. Munosabat (taqqoslash) amali

Taqqoslash amali ikkita qiymatni bir-biri bilan solishtirish uchun ishlatiladi. Taqqoslash amallari xuddi mantiqiy amallar kabi ishlaydi, dastur mantiq'ini boshqaradi va ikki yoki undan ortiq o'zgaruvchilarni taqqoslab xulosa qabul qiladi. Bu amallar ishlatilganda taqqoslash natijasi true(rost) yoki false(yolg'on) mantiqiy qiymatlarga teng bo'ladi.

3.7-jadval

Taqqoslash amallari

Amal belgisi	Nomi	Ishlatilishi	Natija
<	Kichik	$a < b$	a o'zgaruvchisi b dan kichik bo'lsa, rost
>	Katta	$a > b$	a o'zgaruvchisi b dan katta bo'lsa, rost
<=	Kichik yoki teng	$a \leq b$	a o'zgaruvchisi b dan kichik yoki teng bo'lsa, rost
>=	Katta yoki teng	$a \geq b$	a o'zgaruvchisi b dan katta yoki teng bo'lsa, rost

3.6. Ternar amali

Sintaksisi:

shart ? shart rost bo'lgandagi qiymat: shart yolg'on bo'lgandagi qiymat

Ternar amali if-else operatoriga o'xshash.

3.6-misol. Ternar amalining ishlatilishi

```
<? ($natija>50 ? "Imtihondan o'tdingiz" : " Imtihondan o'tmadingiz") ?>
```

Bu ifoda talabalarning imtihondan o'tgan yoki o'tmaganligini aniqlovchi shart sifatida ishlatiladi.

IV bob. PHP YORDAMIDA SO'ROVLARNI QAYTA ISHLASH

Bu bobda serverga ma'lumotlarni jo'natish metodlari va ularni PHP yordamida qayta ishlash o'rganiladi. Mijoz-server texnologiyasi-ning asosiy tushunchalari ko'riladi. Post va Get metodlarining qisqacha tavsifi beriladi. HTML formadan ma'lumotlarni olish mexanizmi va ularni PHP yordamida qayta ishlash bilan tanishiladi.

4.1. Mijoz-server texnologiyasi asoslari

Biz ma'ruzaning boshida PHP server yordamida qayta ishlanadigan skriptli til ekanligini ta'kidlab o'tgandik. Hozir biz server nima, u qanday funksiyalarni bajaradi va umuman qanaqa serverlar bo'lishini aniqlab olamiz. Gap server haqida ketayotganida o'z-o'zidan mijoz tushunchasi esga tushadi. Chunki bu ikki tushuncha bir-biri bilan uzviy bog'liq. Ularni mijoz-server kompyuter arxitekturasini birlashtirib turadi. Odatda, server haqida gapirilayotganida mijoz-server arxitekturasini nazarda tutiladi, mijoz haqida gapirilayotganida ham shu arxitektura nazarda tutiladi. Uning mazmuni shundaki, ikki ichki sistema funksiyasini ajratish qandaydir harakatlarni bajarayotgan so'rovni jo'natayotgan mijoz va shu so'rovni bajarayotgan server yordamida amalga oshiriladi.

Har xil jarayonlarning bajarilishini nazorat qilib turadigan dasturlar to'plamiga server deyiladi. Ushbu dasturlar mos ravishda qandaydir kompyuterga o'rnatilgan bo'ladi. Ko'pincha server o'rnatilgan kompyuterni server deb atashadi. Mijoz so'rovi bo'yicha qandaydir aniq jarayonni ishga tushirish va ishining natijasini mijozga jo'natish kompyuter-serverning asosiy funksiyasiga kiradi.

Serverni istalgan xizmatidan foydalanilgan jarayonga mijoz deyiladi. Foydalanuvchi ham dastur, ham mijoz bo'la olishi mumkin. Mijozning asosiy vazifasi ilova talab qilayotgan vaqtda server bilan aloqada ilovani bajarish va uni amalga oshirishdir. Shu bilan birga mijoz foydalanuvchiga ishining mantig'ini va kerak bo'lganda uni masalasini serverga jo'natishni hamda ilova bilan ishlash uchun interfeys yaratib berishni ta'minlashi kerak.

Mijoz va server o'rtasidagi o'zaro aloqa mijoz tashabbusiga ko'ra boshlanadi. Mijoz xizmat turini tanlaydi, seansni o'rnatadi, o'ziga kerak bo'lgan natijani oladi va ish tugagani haqida xabar qiladi.

Ko'pincha bir nechta mijozlar bir vaqtning o'zida bitta server xizmatlaridan foydalanadi. Shuning uchun har bir server ma'lumotlar xavfsizligini ta'minlashi va katta ishlab chiqarishga ega bo'lishi kerak.

Server kompyuterga ulanayotgan vaqtda lokal yoki global tarmoq bo'lsa maqsadga muvofiqroq bo'ladi. Serverni alohida turadigan kompyuterga ham ulash mumkin (unda u bir vaqtning o'zida ham server, ham mijoz vazifasini bajaradi).

Odatda, kompyuter-serverda bir vaqtning o'zida bir nechta server dasturlari ishlaydi. Biri elektron pochta bilan, ikkinchisi fayllarni taqsimlash bilan, uchinchisi web sahifalar bilan ishlaydi. Hamma serverlar turidan bizni ko'proq WWW serveri qiziqtiradi. Odatda, uni http-server, web server yoki oddiy qilib server deb atashadi. Web server o'zida nimani mujassamlashtiradi? Birinchidan, bu axborot resurslar ombori. Ikkinchidan, bu resurslar foydalanuvchilarga internet standartlariga mos holda beriladi va saqlanadi (masalan, HTTP ma'lumotlarini uzatish protokoli singari). Bu protokol bilan bog'liq holda ma'lumotlarni qanday tasvirlash mumkinligini keyinroq ko'rib o'tamiz. Web serverdagi sahifalar bilan ishlash brauzer dasturlari (masalan, IE, Opera, Mozilla) yordamida amalga oshiriladi. Bunda brauzer dasturi HTTP protokoliga mos ravishda yaratilgan so'rovlarni serverga jo'natadi. Server esa topshiriq, boshqacha aytganda, so'rovni qayta ishlaydi va bajaradi, o'z navbatida qayta ishlash jarayonida boshqa serverlar (bu yerda dastur serverlari nazarda tutilyapti, masalan, MBBT serveri, fayl serveri, pochta serveri kabilar) bilan aloqa o'rnatishi mumkin.

Bundan keyin "server" deganda web-serverni tushunamiz.

Web serverga misol qilib, Apache guruhining Apache serverini, Microsoft kompaniyasining Internet Information Server (IIS), Sun Microsystems firmasining Sun, Rambler kompaniyasi uchun Igor Sisoyev tomonidan yaratilgan nginx(engine x), Google kompaniyasining Google, NCSA (National Center for Supercomputing Applications) tashkilotining NCSA HTTPd, lighttpd(lighty) va boshqalarni keltirish mumkin.

Mijoz-server texnologiyasini qo'llash tarmoqqa birlashtirilgan kompyuterlarga asoslanadi, bu kompyuterlardan biri maxsus boshqaruv funksiyalarini bajaradi (tarmoq serveri bo'ladi).

Mijoz-server arxitekturası quyidagicha qurilgan:

Ma'lumotlar bazasi fayllar to'plami shaklida maxsus kompyuterning (tarmoq serveri) qattiq diskida joylashadi.

MBBT ham tarmoq serverida joylashadi.

Mijoz kompyuterlaridan iborat bo'lgan lokal tarmoq mavjud bo'lib, har bir kompyuterda MB bilan ishlovchi mijoz dasturi o'rnatilgan.

Har bir mijoz kompyuterda foydalanuvchilar dasturini ishga tushirish imkoniga ega. Dasturning foydalanuvchi interfeysi yordamida foydalanuvchilar ma'lumotlarni tanlash/yangilash uchun serverdagi MB ga murojaat qiladi. Murojaat qilish uchun maxsus SQL so'rovlar tili ishlatiladi, ya'ni serverga faqat so'rov matni jo'natiladi.

MBBT da serverdagi MB ning fizik strukturasi haqidagi ma'lumotlar saqlanadi.

MBBT serverdagi ma'lumotlarga murojaatni qayd qiladi va serverda ma'lumotlarni qayta ishlash amallari bajariladi va natija mijoz kompyuteriga yuboriladi. Shu tarzda MBBT natijalarni mijoz dasturiga yuboradi.

Dastur esa o'zidagi foydalanuvchi interfeysi yordamida natijalarni tasvirlab beradi.

Bajariladigan funksiyalar server va mijoz o'rtasida qanday taqsimlanganini ko'rib chiqamiz:

- mijoz dasturi funksiyalari:

So'rovlarni serverga jo'natish.

Serverdan olingan so'rov natijalarini talqin qilish.

Natijalarni biror shaklda foydalanuvchiga ko'rsatish (foydalanuvchi interfeysi).

- Server funksiyalari:

Mijoz-dasturlardan so'rovlarni qabul qilish.

So'rovlarni talqin qilish.

MB dagi so'rovlarni optimallashtirish va bajarish.

Natijalarni mijoz-dasturga jo'natish.

Xavfsizlik tizimini va murojaatni chegaralash.

MB butunligini boshqarish.

Ko'p foydalanuvchili ish rejimi stabililigini ta'minlash.

4.2. HTTP protokoli va ma'lumotlarni serverga jo'natish metodlari

Internet foydalanuvchi va tarmoq o'rtasidagi interfeysni amaliy darajagacha ta'minlaydigan, fizik daraja bilan ikkilamchi ma'lumotni uzatishda fizik aspektlarga bog'liq ko'p darajali prinsip asosida qurilgan.

HTTP — bu, Internetda gipermatnli ma'lumotni almashtirish uchun ishlangan amaliy daraja protokoli.

HTTP serverga jo'natadigan to'liq so'rovlar ko'rsatmalari metodlari yig'indisidir. Bu metodlar resursni ko'rsatish uchun jo'natmalar (ssilkalar) tartibiga asoslanadi, ushbu metod qo'llanilishi kerak bo'lganda resursning joylashgan joyini (Universal Resource Locator, URL) yoki uni universal nomini (Universal Resource Name, URN) topadigan universal identifikatorlari (Universal Resource Identifier, URI) qo'llaniladi.

HTTP protokoli qo'llanilayotgan vaqtda tarmoq ichida xabar jo'natish xuddi Internet (RFC-822) pochta xabari formatiga yoki MIME (Multipurpose Internet Mail Exchange) xabarlar formatiga o'xshash formatda yuboriladi.

HTTP SMTP (elektron pochta protokoli), NNTP (xabarlar jo'natish protokoli), FTP (fayllar jo'natish protokoli), Gopher va WAIS kabi mavjud Internet protokollariga ruxsat beruvchi har xil foydalanuvchi dasturlari va shlyuz dasturlari kommunikatsiyasi uchun qo'llaniladi. HTTP shunday shlyuzlarga oraliq server-dasturlari (proxy) da ma'lumotlarni yo'qotishlarsiz jo'natish uchun yaratilgan.

Protokol so'rov/javob prinsipini amalga oshiradi. Mijoz so'rayotgan dastur javob berayotgan dastur-serverdan quyidagilarni so'raydi:

- Ruxsat metodi;

- URI manzili;

- Protokol versiyasi;

- Uzatilayotgan ma'lumotlar tipi haqidagi xabarni (MIME dagi formaga o'xshash), so'rov jo'natgan mijoz haqida ma'lumotni;

- Server javobi:

- Protokol versiyasi va qaytish kodi (muvaffaqiyatli yoki xatoli) chiquvchi qator holati;

- Server haqidagi ma'lumot, metama'lumot (xabar mundarijasi haqida ma'lumot) va xabar tanasini mujassamlashtirgan xabar.

Mijoz va server o'rtasida aloqa o'rnatilayotganda kim ochishi yoki yopishi protokolda ko'rsatilmaydi. Qoidaga asosan aloqa o'rnatishda mijoz ochadi, server bo'lsa javob jo'natilgandan keyin o'rtadagi aloqani uzadi.

Serverga so'rov qanday shaklda jo'natilishini to'liqroq ko'rib chiqamiz.

4.3. Mijoz so'rovi shakli

Mijoz serverga ikki xil: to'liq yoki qisqartirilgan shaklda so'rov jo'natadi. Ushbu so'rovlar mos holda to'liq so'rov va oddiy so'rov deb aytiladi.

Oddiy so'rov ruxsat metodi va resurs manzilini mujassamlashtiradi. Buni to'liqroq shunday yozish mumkin:

< Oddiy-so'rov > := <Metod> <probel belgisi>

<So'ralayotgan -URI> <yangi qator belgisi>

Metod tariqasida GET, POST, HEAD, PUT, DELETE va boshqalarni ko'rsatish mumkin. Ular ichida kengroq tarqalganlari haqida keyinroq aytib o'tamiz.

Oddiy so'rovga misol:

GET http://phpbook.info/

Bu yerda GET — ruxsat metodi, ya'ni so'ralayotgan resursga qo'llanilishi kerak bo'lgan metod, http://phpbook.info/ esa so'ralayotgan resursning URL-manzili.

To'liq so'rov esa qator holati, bir nechta sarlavhalar (so'rov sarlavhasi, umumiy sarlavha va mundarija sarlavhasi) va so'rov tanasidan iborat bo'lishi mumkin. To'liq so'rovning umumiy ko'rinishini quyidagicha yozish mumkin:

< to'liq so'rov > := < qator holati >

(<umumiy sarlavha> < so'rov sarlavha>

<mundarija sarlavha>)

<yangi qator belgisi>

[<so'rov mundarijasi>]

Bu yerda to'rtburchak qavslar muqobil variantning vertikal chiziqlar ustida ko'rsatilgan sarlavhaning keraksiz elementlarini anglatadi. Bu yerda <qator holati> So'rov metodi va URI resursini (xuddi oddiy so'rov kabi) va bundan tashqari HTTP protokolining qo'llanilgan versiyasini o'z ichiga oladi. Masalan, tashqi dasturni chaqirish uchun quyidagi qator holatiga ta'sir qilish mumkin:

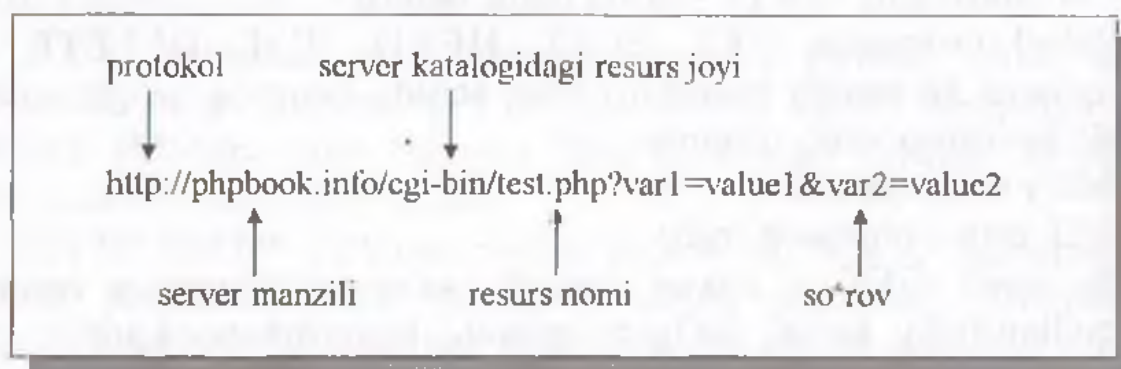
POST http://phpbook.info/cgi-bin/test HTTP/1.0

Bu holda POST metodi va HTTP protokolining 1.0 versiyasi qo'llanilyapti.

So'rovning ikki shaklida ham so'ralayotgan resursning URI si muhim o'rin egallaydi. Ko'pincha URL-resurs manzili sifatida URI ishlatiladi. Serverga murojaat qilayotgan vaqtda URLning to'liq shaklini ham, qisqartirilgan shaklini ham qo'llash mumkin.

To'liq shakli protokolga ruxsat toifasini, resurs serveri manzilini va serverda manzil resursini o'z ichiga oladi (6.2-rasm).

Qisqartirilgan shaklda esa protokol va server manzili server asosidan manbaning faqatgina qayerda joylashganligini ko'rsatadi. To'liq shaklda agar mumkin bo'lsa, so'rovni boshqa serverga jo'natib yuborish ham qo'llaniladi. Agar bitta server bilan ishlanayotgan bo'lsa ko'pincha qisqartirilgan shakl ishlatiladi.



4.1-rasm. URLning to'liq shakli.

Endi so'rovlar jo'natishning keng tarqalgan metodlarini ko'rib chiqamiz.

4.4. Metodlar

Yuqorida aytib o'tganimizdek, mijozning serverga istalgan so'rovi metodni ko'rsatish bilan boshlanishi kerak. Metod mijoz so'rovining maqsadi haqida xabar beradi. HTTP protokoli juda ham ko'p metodlarni qo'llaydi, lekin ularning ichidan faqat uchta ko'p qo'llaniladi: POST, GET va HEAD. GET metodi resurs so'rovida URI yordamida identifikatsiya qilingan istalgan ma'lumotni olishi mumkin. Agar URI dasturga ko'rsatayotgan bo'lsa, unda dastur ishining natijasi sifatida dasturning matni emas o'zi qaytadi (agar matn dastur ishi-

ning natijasi bo'lmasa). Qo'shimcha ma'lumot so'rovni qayta ishlash uchun so'rovning o'ziga qo'yiladi. GET metodini resurs qismi maydonida qo'llayotgan vaqtda so'ralgan ma'lumot mos ravishda qaytadi (masalan, HTML matni hujjati).

GET metodining har xil shakllari mavjud — shartli GET. Bu metod agar shart bajarilgan bo'lsa, so'rov sarlavhasi maydonida if-Modified-Since ning mavjudligi haqida so'rovga javob yuborish kerakligi xabar beradi. Aniqroq gapiradigan bo'lsak, resurs if-Modified-Since da ko'rsatilgan vaqtdan keyin o'zgartirilgan bo'lsa, resurs maydoni so'rovga javob kabi beriladi.

HEAD metodi ham GET metodiga o'xshash, faqat u resurs tanasini qaytaradi va shartli analogga ega emas. HEAD resursi resursdan ma'lumot olish uchun qo'llaniladi.

POST metodi serverga resurslar annotatsiyasi, yangiliklari va pochta xabarlar, kerakli va katta hajmdagi ma'lumotni jo'natish uchun ma'lumotlar bazasida qo'shimchalar qo'shish uchun yaratilgan. POST metodida GET va HEAD metodlaridan farqli ravishda formalar maydonidan yoki boshqa manbadan olingan ma'lumotni, ya'ni resurs tanasini uzatadi.

Shu vaqtgacha biz faqat asosiy tushunchalar va nazariyalar bilan tanishdik. Endi ularni amaliyotda qo'llashni o'rganamiz. Quyida biz serverga qanday qilib so'rov berish va uning javobini qayta ishlashni o'rganamiz.

4.5. HTML formasining serverga ma'lumotlarni jo'natish uchun qo'llanilishi

Ma'lumotlarni serverga jo'natish uchun HTML tilida maxsus konstruksiya — formalar mavjud. Formalar foydalanuvchidan ma'lumot olish uchun yaratilgan. Masalan, foydalanuvchining logini va paroli uni sayt sahifasidan foydalanishiga ruxsat berishi uchun kerak. Yoki foydalanuvchining u bilan aloqa qilishi uchun shaxsiy ma'lumotlari kerak. Formalar aynan shunday ma'lumotlarni olish uchun ishlatiladi. Formalarga matn kiritish yoki ro'yxat ichidan kerakligini tanlab kiritish mumkin. Formalarga yozilgan ma'lumotlar serverning maxsus dasturiga qayta ishlash uchun jo'natiladi (masalan, PHP skriptiga). Foydalanuvchi kiritgan ma'lumotlarga qarab bu dastur ularni har xil web-sahifa ko'rinishida formatlashi,

ma'lumotlar bazasiga so'rovni jo'natishi, har xil izohlar kiritishi va hokazolarni bajaradi.

HTML formasining sintaksisi bilan tanishamiz. Ko'pchilik bu bilan tanish bo'lishi mumkin, lekin biz uning asosiy joylarini qaytarib o'tamiz.

Formaga ma'lumotlarni chiqazish uchun turli boshqaruvchi elementlar ishlatiladi. Bu elementlardan birida foydalanuvchi ma'lumotlarni klaviatura yordamida kiritadi, boshqasida esa u sich-qoncha yordamida kerakli variantni tanlaydi. Formada foydalanuvchi tomonidan o'zgartirib bo'lmaydigan yashirin maydonlar ham qatnashadi. Bitta sahifa tarkibida bir nechta forma bo'lishi mumkin, shuning uchun bir formani boshqasidan ajratib olish imkonini beruvchi vosita zarur. Shunga ko'ra, foydalanuvchi forma bilan ishlayotganda, formadagi ma'lumotlar qaysi sahifaga o'tishi kerakligi ko'rsatilishi kerak (ya'ni, jo'natish tugmasi bosilganda forma ma'lumotlarni qayerga jo'natishi lozim bo'lgan manzil ko'rsatilishi kerak). Barcha masalalar quyidagicha HTML teglari bilan berilgan forma ichiga olib yechiladi:

`<form action = amal method = "metod">` — forma elementlari-`</form>`

Berilgan fragmentdan ko'rinib turibdiki, forma tegida ikkita zarur element: amal va metod ko'rsatiladi. Amal, forma ma'lumotlari jo'natiladigan va qayta ishlanadigan sahifa nomini, metod esa ma'lumotlarni ushbu sahifaga qanday uzatish metodini aniqlaydi. Ikkita metod mavjud:

get metodi formaning barcha ma'lumotlarini URL tugashida uzatadi. Bu metod maxsus tillar bilan bog'liq turli chegaralashlarni va ma'lumot uzunligini kamdan kam hollarda qabul qiladi;

post metodi formadagi barcha ma'lumotlarni so'rov tanasida uzatadi. Bu metod get ga nisbatan ko'p qo'llaniladi.

Klaviatura bilan kiritishga yo'naltirilgan forma elementlari

Formani yaratish klaviatura bilan kiritishga yo'naltirilgan elementlar bilan boshlanadi. Bunday elementlar ikkita — matnli maydon (input) va matnli soha (text area).

Matnli maydon forma elementi

Matnli maydonda, odatda, qisqa matnli ma'lumotlar kiritiladi, masalan, ismi, manzili yoki elektron pochta manzili.

Sintaksisi:

```
<input type="text" name="o'zgaruvchi_nomi" size="N"
maxlength="N" value="">
```

Matnli maydonni aniqlash beshta atributni o'z ichiga oladi:

- type — element tipi (matnli maydon uchun— text);
- name — kiritilgan ma'lumot saqlanadigan o'zgaruvchi nomi (o'zgaruvchi_nomi);
- size — matnli maydonning brauzerdagi umumiy uzunligi (N);

maxlength — matnli maydonga kiritiladigan belgilarning eng katta soni;

value — odatda matnli maydonda chiqadigan qiymat.

Matnli maydonlarning alohida turi bu parollarni kiritish maydoni. U xuddi oddiy matnli maydon kabi ishlaydi, bitta farqi kiritilayotgan belgilar yulduzchaga almashadi. Parollarni kiritishga forma maydoni yaratish uchun type="text" o'rniga type="password" ni kiritish kerak, ya'ni:

```
<input type="password" name="o'zgaruvchi_nomi" size="N"
maxlength="N" value="">
```

Matnli soha forma elementi

Matnli soha (textarea) bir nechta katta hajmdagi oddiy ismlar, elektron pochta manzillari, chegaralanmagan matnlarni kiritish uchun ishlatiladi.

Sintaksisi:

```
<textarea name="o'zgaruvchi_nomi" rows="N" cols="N"
value=""></textarea>
```

Matnli sohani aniqlash uchta atributni o'z ichiga oladi:

- name — kiritilgan ma'lumot saqlanayotgan o'zgaruvchi_nomi;
- rows — matnli sohani satrlar soni;
- cols — matnli sohani ustunlar soni.

Formaning boshqa elementlarida foydalanuvchi oldindan berilgan variantlardan birini sichqoncha yordamida tanlaydi. Bular bayroqchalar (checkbox), yoqib-o'chirgichlar (radio button) va ochiladigan ro'yxatlar (select).

Checkbox forma elementi

Bayroqchalar (checkbox) foydalanuvchi tayyor to'plamdan bir yoki bir nechta variantni tanlash holatlarida ishlatiladi.

Sintaksisi:

```
<input type="checkbox" name="o'zgaruvchi_nomi" value="qiymat">
```

Bayroqchani aniqlash o'z ichiga uchta atributni oladi:

- type – element tipi (bayroqcha uchun – checkbox);

name – kiritilgan ma'lumot saqlanadigan o'zgaruvchi_nomi (hozirgi holatda – element holati);

value – odatga ko'ra o'zgaruvchiga teng bo'lgan qiymat. Agar bayroqcha o'rnatilgan bo'lsa, aynan ushbu qiymat ko'rsatilgan nomdagi o'zgaruvchiga teng bo'ladi. Agar bayroqcha o'rnatilmagan bo'lsa, value atributi qiymati ishlatilmaydi.

radio button forma elementi

Yoqib-o'chirgichlar (radio button) bayroqchani bir turi hisoblanadi. U amalda quyidagicha ishlatiladi: bir vaqtning o'zida faqatgina bitta yoqib-o'chirgich o'rnatiladi, qolganlari esa istisno qilinadi.

Sintaksisi:

```
<input type="radio" name="o'zgaruvchi_nomi" value="qiymat">
```

Yoqib-o'chirgichlarni aniqlashda uchta atribut ishtirok etadi:

- type – element tipi (yoqib-o'chirgichlar uchun – radio);

name – kiritilgan ma'lumotlar saqlanadigan o'zgaruvchi_nomi (ayni paytda – element holati);

value – odatga ko'ra o'zgaruvchi tenglanadigan qiymat. Agar yoqib-o'chirgich o'rnatilgan bo'lsa, aynan ushbu qiymat ko'rsatilgan nomdagi o'zgaruvchiga tenglanadi. Agar yoqib-o'chirgich o'rnatilmagan bo'lsa, value atributining qiymati ishlatilmaydi.

select forma elementi

Agar mumkin bo'lgan variantlar ro'yxati ko'p bo'lsa va foydalanuvchi ulardan bitta variantni tanlashi kerak bo'lgan holatlarda ochiladigan ro'yxatlar (select)ni qo'llash alohida qulay hisoblanadi.

Sintaksisi:

```
<select name="o'zgaruvchi_nomi">
```

```
<option value="qiymat1">
```

...

...

...

```
<option value="qiymatN">
```

```
</select>
```

Ochiladigan ro'yxatlarni aniqlashda uchta atribut ishtirok etadi:

- name — kiritilgan ma'lumot saqlanadigan o'zgaruvchi_nomi (ayni paytda — ro'yxatdan tanlangan satr);
- value — odatga ko'ra ro'yxatda chiquvchi qiymat.

Yashirin maydon forma elementi

Yashirin maydon brauzerda ko'rinmaydi va odatda sahifalar o'rtasida ma'lumotlarni uzatish uchun ishlatiladi. Yashirin maydonlarni aniqlash sintaksisi matnli maydonlarni aniqlash sintaksisiga juda o'xshash bo'lib, faqatgina tip atributi bilan farq qiladi.

Sintaksisi:

```
<input type="hidden" name="o'zgaruvchi_nomi" value="qiymat">
```

Yashiringan maydonni aniqlashda uchta atribut ishtirok etadi:

- type — element tipi (yashirilgan maydon uchun — hidden);
- name — yashirilgan ma'lumot saqlanadigan o'zgaruvchi_nomi;
- value — odatga ko'ra yashirilgan maydonda saqlanayotgan qiymat.

Yashirin maydon brauzerda ko'rsatilmasa ham, foydalanuvchi View Source buyrug'ini ishlatib formada qanday yashirin qiymat saqlanayotganligini ko'rishi mumkin.

Ma'lumotni jo'natish tugmasi

Ma'lumotlarni jo'natish tugmasi <form> tegidagi action atributida berilgan amalni bajaradi. Ushbu atributda ko'rsatilgan sahifaga formadagi ma'lumotlarni jo'natadi.

Sintaksisi:

```
<input type="submit" value="Tugma_uchun_matn">
```

Tugmani aniqlashda ikkita atribut ishtirok etadi:

- type — element tipi (ma'lumotlarni jo'natish tugmasi uchun — submit);
- value — Tugma_uchun_matn.

Formadagi ma'lumotlarni tozalash (sbros) tugmasi

Tozalash tugmasi forma elementidagi barcha kiritilgan ma'lumotlarni o'chiradi. Web formalarda bu tugmani tez-tez uchratish mumkin.

Sintaksisi:

```
<input type="reset" value="Tugma_uchun_matn">
```

Tugmani aniqlashda ikkita atribut ishtirok etadi:

type — element tipi (tozalash tugmasi uchun— reset);

value — Tugma_uchun_matn.

Tozalash tugmasi ko'rinishi xuddi jo'natish tugmasi kabi, shuning uchun bu tugmalarga mos nomlar beriladi.

Forma elementlari bilan tanishib chiqdik. Endi shu forma elementlari yordamida misol ko'ramiz.

4.1-misol. Webtest dasturining testni boshlash formasini yarata-miz.

6.1-listing. index.html

```
<h2>Ma'lumotlarni kiriting</h2>
<form action="start.php" method=POST> <!--formani.yaratamiz-->
<!--berilgan formani start.php fayli qayta ishlaydi,
forma ma'lumotlarini jo'natishda POST usuli qo'llaniladi-->
Ismingiz<br><input type=text name="first_name" value="Ismingizni
kiriting"><br>
Familiyangiz<br><input type=text name="last_name"><br>
E-mail <br><input type=text name="email"><br>
<P>Fanni tanlang:<br>
<select name="fan">
<option value="tarix"> Tarix </option>
<option value="falsafa"> Falsafa </option>
<option value="matematika"> Matematika </option>
<option value="informatika"> Informatika </option> </select>
<P><input name="confirm" type=checkbox checked> To'ldirganligingizni
tasdiqlang <br>
<input type=submit value="Jo'natish">
<input type=reset value="Tozalash"> </form>
```

Bu faylning brauzerdagi ko'rinishi:

HTML formasi shu kabi ko'rinishlarda yaratiladi. Ko'rinib turganidek, formada ma'lumotlarni jo'natish metodi ko'rsatilgan bo'lishi kerak. Endi GET va POST metodlari farqini ko'ramiz.

GET metodi

GET metodi yordamida forma ma'lumotini jo'natish uchun forma tarkibidagi URL ga so'roq belgisidan so'ng & ampersand yordamida birlashtirilgan nomi=qiymati juftliklari (so'rov qismi) qo'shiladi:

action?name1=value1&name2=value2&name3=value3

bu yerda action — dasturning formani qayta ishlashi kerak bo'lgan URL manzili (bu form tegi bilan action atributiga berilgan dastur

yoki agar atribut tushirib qoldirilgan bo'lsa, joriy dasturning o'zi). name1, name2, name3 nomlar forma elementlari nomi, value1, value2, value3 lar esa ushbu elementlarning qiymatidir. Hamma maxsus belgilar = va & ham parametrlar nomiga qarab tushirib qoldiriladi. Shuning uchun bu belgilar va identifikatordagi kirillcha belgilar nomlashda qo'llanilmaydi.

4.2-rasm. HTML formasiga misol.

Agar maydonga qandaydir xizmatchi belgi kiritilgan bo'lsa, u holda u o'n oltitalik sanoq sistemasida beriladi. Masalan, \$ belgisi %24 ga almashadi. Harflar ham xuddi shu kabi beriladi.

Matn va parol kiritish maydoni (bu type=text va type=password atributli INPUT elementi)da qiymat foydalanuvchi tomonidan kiritiladi. Agar foydalanuvchi ushbu maydonga hech narsa kiritmasa unda bu qatorda name= hosil bo'ladi.

checkbox va radio button forma elementlarida value qiymati, belgilangan value atributiga teng bo'ladi. Belgilanmagan forma elementlari so'rov tuzilayotgan vaqtda inkor etiladi. Agar zarur bo'lsa, checkbox forma elementi bitta atribut name (va har xil value) ga ega bo'lishi mumkin. radio button elementi bo'lsa, hamma taklif qilin-

gan variantlardan bittasiga tegishli va shuning uchun bir xil name atributiga va har xil valuelarga ega bo'lishi kerak.

Umuman olganda, GET metodida ma'lumotlarni jo'natish uchun HTML formasini yaratish shart emas. Adres qatordagi URLda kerakli o'zgaruvchilar va ularning qiymatlarini qo'shib qo'yish yetarli.

4.2-misol. URLda ma'lumotlar uzatilishi.

```
http://phpbook.info/test.php?id=J0&user=pit
```

Shu bilan birga ma'lumotlarni GET metodida jo'natishni bir kamchiligi bor — parametrlar qiymatini biror kim o'zgartirib qo'yishi mumkin. Shuning uchun dastur yoki server ishi xavfsizligiga daxldor parol bilan himoyalangan sahifalarda ma'lumotni jo'natish uchun bu metoddan foydalanishni maslahat berilmaydi. Bundan tashqari foydalanuvchi tomonidan o'zgartirish taqiqlangan ma'lumotlarni jo'natish uchun GET metodidan foydalanish yaramaydi.

Shuncha kamchiliklarga qaramasdan, skriptlarni sozlashda (bu holda jo'natilayotgan o'zgaruvchilar nomini va qiymatini ko'rish mumkin) va xavfsizlikka daxlsiz bo'lgan parametrlarni jo'natishda GET metodidan foydalanish yetarli darajada qulay.

POST metodi

POST metodida formalar xuddi GET metodidagidek kodlanadi, lekin so'rov qismi URLga qo'shilmaydi, balki POSTning amallari, ya'ni berilganlar bloki qismidek yuboriladi. Agar action atributi mavjud bo'lsa, unda URL ushbu berilganlar blokini qayerga jo'natishni aniqlaydi. Bu metod yuqorida aytib o'tilgandek katta hajmdagi ma'lumotlar blokini jo'natish uchun qo'llaniladi.

Jo'natilayotgan faylning uzunligi content_length doirasidagi o'zgaruvchilarga jo'natiladi, ma'lumotlar tipi esa content_type o'zgaruvchisiga jo'natiladi.

Ma'lumotlarni post metodi bilan jo'natishni faqat HTML formasida yordamida amalga oshirish mumkin, chunki ma'lumotlar GET-dagidek so'rov sarlavhasiga emas, balki so'rov qismiga jo'natiladi. Shunga ko'ra faqatgina formaga kiritilgan qiymatni o'zgartirish orqali parametr qiymatini o'zgartirish mumkin. Foydalanuvchi POSTdan foydalanayotganda serverga berilayotgan ma'lumotlarni ko'rmaydi.

POST so'rovlarining asosiy xususiyati uning xavfsizligi kattaligi va funksionalligidir. Shuning uchun POST metodidan katta hajmdagi va kerakli ma'lumotlarni jo'natishda foydalaniladi. Lekin bu

mexanizmga ham butunlay ishonib bo'lmaydi, chunki unda ham POST so'rovlarini o'zgartirish mumkin. Masalan, html-faylini o'z mashinamizda yaratib uni kerakli ma'lumotlar bilan to'ldirilsa kifoya. Bundan tashqari POST metodidan hamma mijozlar foydalana olmaydilar, chunki uning ishlatish variantlari cheklangan.

Ma'lumotlarni serverga istalgan yo'l bilan jo'natishda faqat ma'lumotlar emas, balki foydalanuvchi tomonidan kiritilgan bir qator doiraviy o'zgaruvchilar nomli o'zgaruvchilar qatori, mijozni xarakterlaydigan parametrlar, uning ishi haqidagi ma'lumotlar, fayllarga yo'l va h.k.lar ham beriladi. Quyida doiraviy o'zgaruvchilardan ayrimlarini keltiramiz:

REMOTE_ADDR – xost (kompyuter)ning so'rov jo'natilayotgan IP-manzili;

REMOTE_HOST – so'rov jo'natilgan xostning nomi;

HTTP_REFERER – joriy skriptga murojaat (ssilka) etiladigan qator manzili;

REQUEST METHOD – so'rovni jo'natishda qo'llanilgan metod;

QUERY_STRING – URLda so'roq belgisidan keyin joylashgan so'rov qismi;

SCRIPT_NAME – bajarilishi kerak bo'lgan virtual dasturga yo'l;

HTTP_USER_AGENT – mijoz foydalanayotgan brauzer haqida ma'lumot.

4.6. So'rovlarni PHP yordamida qayta ishlash

Mijoz so'rovining serverda maxsus dasturlar yordamida qayta ishlashini ko'rib chiqqansiz. Ushbu maxsus dasturni PHP tilida yozish va unda kiritilgan ma'lumotlar bilan istalgan amalni bajarish mumkin. Bu dasturni yozishda PHP ning instrumentlar va qoidalari bilan tanishib chiqish kerak.

PHP skripti ichida mijozga HTTP protokoli asosida berilgan bir qancha ma'lumotlarga ruxsat olish yo'llari mavjud. PHP 4.1.0. versiyasigacha bunday ma'lumotlarga ruxsat jo'natilgan o'zgaruvchilar nomi (ma'lumotlar "o'zgaruvchi nomi, " =" belgisi, o'zgaruvchi qiymati" juftligi ko'rinishida berilgan) asosida amalga oshirilgan. Shu tariqa agar `first_name=Lola` berilgan bo'lsa, skript ichida Lola

qiymatli \$first_name o'zgaruvchi hosil bo'ladi. Agar ma'lumotlarni qanday yo'l bilan berilganligini ajratish talab qilingan bo'lsa, unda berilgan o'zgaruvchilar kaliti bo'lgan \$HTTP_POST_VARS va \$HTTP_GET_VARS assotsiativ massivlar qo'llanilgan. Xuddi shunday agar first_name=Lola GET metodi bilan jo'natilgan bo'lsa, unda

\$HTTP_GET_VARS["first_name"]="Lola" bo'ladi.

Dasturda o'zgaruvchilar nomini to'g'ridan to'g'ri yuborish xatarli emas. Shuning uchun PHP 4.1.0 versiyasidan boshlab HTTP so'rovlari yordamida berilgan o'zgaruvchilarga maxsus \$_REQUEST massivi yordamida murojaat etish mumkin. Bu massivlar POST va GET metodi hamda HTTP cookies yordamida berilgan ma'lumotlarni o'z ichiga oladi. Bu superglobal assotsiativ massiv, shuning uchun mos o'zgaruvchi (forma elementi) nomini kalit sifatida ishlatib, dasturning istalgan qismidan uning nomini olish mumkin.

4.1-misoldagi 4.1-listingda test topshiriladigan fanlar ro'yxatining formasi yaratildi. Endi shu forma yordamida jo'natilgan ma'lumotlarni qayta ishlovchi test.php faylining skriptini yozamiz:

4.2-listing. Bu index.html formasini qayta ishlovchi start.php fayli

```
<? $str = "Salom, ".$_REQUEST["first_name"]. " ".$_REQUEST["last_name"]."! <br>";  
$str .= "Siz test topshirish uchun ".$_REQUEST["fan"]." fanini tanladingiz";  
echo $str; ?>
```

U holda, agar biz formaga "Vohid" ismini va "Fozilov" familiyasini kiritganimizda va fanlar ichidan Informatika fanini tanlaganimizda brauzer ekranida quyidagi xabarni olamiz:

Natija:

Salom, Vohid Fozilov!

Siz test topshirish uchun Informatika fanini tanladingiz

\$_REQUEST massivini kiritish uchun \$HTTP_POST_VARS va \$HTTP_GET_VARS bir xillik massivlari mos ravishda \$_POST va \$_GET ga qayta nomlanadi. \$_POST va \$_GET massivlari boshqalardan farqli ravishda metodlar va funksiyalar ichida to'g'ridan to'g'ri ruxsatga ega superglobal massivlarga aylanadi.

Bu massivlarning qo'llanishiga misollar keltiramiz. `first_name`, `last_name`, fan nomli elementlarni o'z ichiga olgan formani qayta ishlash kerak. POST va boshqa metodlar bilan berilgan ma'lumotlarni qayta ishlashni quyidagicha amalga oshirish mumkin:

```
<? $str = "Salom,  
  $_POST ["first_name"]."  
  $_POST ["last_name"] ."! <br>";  
$str .= " Siz test topshirish uchun " . $_POST["fan"] ." fanini tanladingiz";  
echo $str; ?>
```

Agar biz yuqoridagi misoldagidek "Vohid " ismini va "Fozilov" familiyasini kiritsak va fanlar ichidan Informatika fanini tanlasak, u holda brauzer ekranida quyidagi hosil bo'ladi:

Salom, Vohid Fozilov!

Siz test topshirish uchun Informatika fanini tanladingiz

Qayta ishlangan skriptlarni xotiraga saqlash uchun PHP 4.1.0 versiyasidan oldingilarida `registr_globals` o'zgaruvchilarga bevosita nomlari bo'yicha ruxsatni taqiqlaydigan yoki taqiqlamaydigan direktivasi kiritilgan edi. Agar PHP ning konfiguratsiya fayli(`httpd.conf`)da `registr_globals=on` parametri mavjud bo'lsa, u holda serverdagi GET va POST metodi bilan berilgan o'zgaruvchilarga ularning nomlari bo'yicha murojaat etish mumkin (ya'ni, `$first_name` kabi yozish mumkin). Agar `registr_globals=off` bo'lsa, u holda `$_REQUEST["first_name"]` yoki `$_POST["first_name"]`, `$_GET["first_name"]`, `$HTTP_POST_VARS["first_name"]`, `$HTTP_GET_VARS["first_name"]` deb yozish kerak. Xavfsizlik nuqtayi nazaridan bu direktivani o'chirib qo'ygan ma'qul (`registr_globals=off`). `registr_globals` direktivasi yoqilgan vaqtda ham mijoz tomonidan berilgan ma'lumotlar yuqorida qayd etilgan massivlarni o'z ichiga oladi.

Ayrim hollarda so'rov jo'natgan kompyuterning IP manzilini yoki qandaydir doiraviy o'zgaruvchi qiymatini bilish kerak bo'ladi. Bunday ma'lumotni `getenv()` funksiyasi yordamida bajarish mumkin. U parametr sifatida berilgan doiraviy o'zgaruvchi qiymatini qaytaradi.

4.3-misol. `getenv()` funksiyasining ishlatilishi

```
<? getenv("REQUEST_METHOD"); // qo'llanilgan usulni qaytarish  
echo getenv ("REMOTE_ADDR");  
// so'rov jo'natgan foydalanuvchining IP- adresini chiqaradi ?>  
Natija:127.0.0.1
```


Masalan, test.php faylida ushbu skriptni yozsak va html forma ma'lumotlarini jo'natsak. U holda natijada ma'lumotlar jo'natilayotgan kompyuter IP manzili chiqaziladi.

Aytib o'tganimizdek, GET metodi qo'llanilganda, ma'lumotlar so'rov qatoriga "o'zgaruvchi_nomi=resursning URL manzil qiymati" juftligi qo'shilib jo'natiladi. URL da so'roq belgisidan keyin yozilganlarning barchasini

```
getenv("QUERY_STRING");  
buyrug'i yordamida olish mumkin.
```

Shu sababdan GET metodida ma'lumotlarni boshqaxil ko'rinishda berish mumkin. Masalan, qo'shish belgisi orqali bir nechta parametrlarni kiritib, skript ichida bo'lsa so'rov qatorini ajratish yoki faqat bitta parametr qiymatini berish mumkin. Bu holda \$_GET massivida ushbu qiymatga teng bo'lgan bo'sh qator hosil bo'ladi, so'rov qatorida uchragan "+"belgisi esa "_"ga almashtiriladi.

POST metodida ma'lumotlar faqat formalar yordamida beriladi va foydalanuvchi serverga qanday ma'lumotlar jo'natilayotganini ko'ra olmaydi. Uni ko'rish uchun xakker bu formani o'zining formasiga almashtirishi kerak. Bu holda server qayta ishlangan noto'g'ri formani jo'natilishi kerak bo'lgan joyga jo'natmaydi. Bunday holga duch kelmaslik uchun ma'lumotlar jo'natilgan sahifa bilan manzil sahifasini tekshirib ko'rish kerak. Buni ham getenv() funksiyasi yordamida bajarish mumkin:

```
getenv("HTTP_REFERER");
```

4.7. Formalar bilan ishlashga misollar

Quyidagi amaliy misollar PHP da formalarni qayta ishlashning turli aspektlarini o'rganishga yordam beradi. Bu misollarda turli yo'llar bilan saytning interaktiv imkoniyatlarini oshirish ko'rsatiladi.

4.1-masala. Saytning Bizga yozing... sahifasida foydalanuvchi bilan muloqot o'rnatuvchi forma yaratamiz. Bu forma orqali foydalanuvchi o'zining sayt haqidagi fikr-mulohazalarini jo'natishi mumkin.

4.3-listing

```
<form action = "process.php" method = "post">  
<b>Bizning sayt haqida fikr va mulohazalaringizni qoldiring:</b>  
<p><b>Ismingiz:</b><br>
```

```

<input type="text" name="name" size="15" maxlength="25"
value=""><br>
<b>E-mail:</b><br>
<input type="text" name="email" size="15" maxlength="45"
value=""><br>
<b>Bizning saytga necha marotaba tashrif buyurasiz?</b><br>
<select name="tashrif">
<option value="">Saytga tashriflar:
<option value="0">Bu mening birinchi kirishim
<option value="1">Bir oyda 1 marta
<option value="2"> Hafta davomida
<option value="3"> Har kuni
</select><br>
<b>Saytning qaysi bo'limlaridan ko'p foydalanasiz:</b><br>
<input type="checkbox" name="kutubxona" value="kutubxona"> Elek-
tron kutubxona<br>
<input type="checkbox" name="foto" value="foto"> Fotogalereya<br>
<input type="checkbox" name="barchasi" value="barchasi"> Barcha
sahifa<br>
<b>Saytning eng yaxshi sahifasi:</b><br>
<input type="radio" name="asset" value="kutubxona">Elektron
kutubxona<br>
<input type="radio" name="asset" value="foto">Fotogalereya<br>
<input type="radio" name="asset" value="forum">Forum<br>
<b>Izohlar:</b><br>
<textarea name="izoh" rows="3" cols="40"></textarea><br>
<input type="submit" value="Kiritish!"></form>

```

4.2-masala. Bir sahifadan ikkinchi sahifaga ma'lumotlarni uza-
tish.

Bu misoldagi formada foydalanuvchining nomi va elektron poch-
ta manzili ko'rsatiladi. Foydalanuvchi ma'lumotlarni jo'natish (Go!)
tugmasini bosganda, forma 4.5- listingda berilgan sahifaga murojaat
etadi, ya'ni foydalanuvchi ma'lumotlari birinchi sahifadan ikkinchi
sahifaga uzatiladi. O'z navbatida, berilgan so'rovga ko'ra \$name va
\$mail o'zgaruvchi qiymatlarini chiqazadi.

4.4-listing

```

<html>
<head> <title>Listing 4.4</title> </head>

```



```

<body bgcolor="#ffffff" text="#000000" link="#cbda74" link="#808040"
alink="#808040">
  <form action="listing4_5.php" method="post">
    <b>Ma'lumotlaringizni kiriting!</b><br>
    Ismingiz:<br>      <input type="text" name="name" size="20"
maxlength="20" value=""><br>
    Email manzilingiz:<br>
    <input type="text" name="email"
size="20" maxlength="40" value=""><br>
    <input type="submit" value="go!">
  </form>
</body> </html>

```

Natija

Ma'lumotlaringizni kiriting!

Ismingiz:

Email manzilingiz:

go!

4.3-rasm. Ma'lumotlarni bir sahifadan boshqasiga jo'natish

4.3-rasmdagi formada kiritilgan ma'lumotlar sahifada aks etishi uchun php.ini faylidagi register_globals=Off ni register_globals=On ga almashtirish kerak. Yoki ma'lumotlarni massiv sifatida jo'natish kerak bo'ladi.

```

<html>
<head> <title>Listing 4.5</title> </head>
<body    bgcolor="#ffffff"    text="#000000"    link="#cbda74"
link="#808040" alink="#808040">
  <? // Ismi va elektron pochta manzilini chiqazish
  print "Salom. $name!. Sizning elektron pochta manzilingiz $email";
  ?>
</body> </html>
  Natija Salom Lola! Sizning elektron pochta manzilingiz lola@mail.ru

```

Bu misolda hammasi oddiy bajariladi: foydalanuvchi forma maydonlarini to'ldiradi va ma'lumotlarni jo'natish tugmasini bosadi. Boshqaruv 4.5-listing sahifasiga uzatiladi. Bu sahifada ma'lumotlar qayta ishlanadi va oxirida ma'lumotlarni chiqazadi.

Formadagi ma'lumotlarni qayta ishlashni boshqa usullari ham mavjud. Ya'ni ma'lumotlarni kiritish, qayta ishlash va chiqazish hammasi bitta sahifada bajariladi. Bu usulning kamchiligi shundaki, hammasi bitta sahifada bo'lganligi uchun sahifa kattalashadi va uni tahrirlash va boshqarish birmuncha qiyinchilik tug'diradi.

4.3-masala. Formani qayta ishlashning alternativ yo'li (bitta sahifada).

Forma ma'lumotlarini bitta sahifada qayta ishlash deyarli oddiy amalga oshiriladi. Forma o'zgaruvchisiga qiymatni tenglashtirish bajarilganligi tekshiriladi. Agar qiymat tenglashtirilgan bo'lsa, sahifa ularni qayta ishlaydi (bizning misolda ma'lumotlarni chiqazadi), aks holda esa formani o'zini chiqazadi. Formani bitta sahifada amalga oshirish uchun dastur kodi 4.6-listingda berilgan. if shartli o'tish operatori \$seenform nomli yashirin maydon o'zgaruvchisi holatini tekshiradi. Agar \$seenform qiymati berilmagan bo'lsa, forma brauzerda chiqaziladi, agar berilgan bo'lsa, u holda forma foydalanuvchi tomonidan to'ldirilgan va kiritilgan ma'lumotlar sahifada qayta ishlanadi (bizning misolda – brauzerga chiqaziladi).

Formaga ma'lumotlarni bitta sahifada kiritish

```

<html>
<head> <title>Listing 4.6</title> </head>

```



```

<body bgcolor="#ffffff" text="#000000" link="#cbda74"
link="#808040" alink="#f808040">
  <? /* $form ichidagi qo'shtirnoqda joylashganlarning hammasi ekranga
chiqishi kerak aks holda xatolik yuzaga keladi. */
  $form = "
  <form action=\"listing4_6.php\" method=\"post\">
  <input type=\"hidden\" name=\"seenform\" value=\"y\">
  <b>Ma'lumotlaringizni kiriting!</b><br>
  Ismingiz:<br>
  <input type=\"text\" name=\"name\" size=\"20\" maxlength=\"20\"
value=\"\"><br>
  Email manzilingiz:<br>
  <input type=\"text\" name=\"email\" size=\"20\" maxlength=\"40\"
value=\"\"><br>
  <input type=\"submit\" value=\"subscribe!\"> </form>;
  // agar forma avval chiqazilmagan bo'lsa, uni chiqazish,
  /* tekshirish uchun $seenform yashirin o'zgaruvchisining qiymati ishla-
tiladi */
  if (!isset($seenform)): print "$form";
  else :
  print "Salom. $name! Sizning elektron pochta manzilingiz $email";
  endif; ?>
</body> </html>

```

4.4-masala. Mashg'ulotga qatnashuvchi o'quvchilarni ro'yxatdan o'tkazish va ro'yxatdan o'tgandan so'ng ularga xabar jo'natish formasini tuzamiz.

4.7-listing. form_yakuniy.html

```

<h2>Qatnashuvchilar ro'yxati</h2>
<form action="listing4_8.php" metod=POST>
<!--formani yaratamiz-->
<!--berilgan formani test.php fayli qayta ishlaydi,
forma ma'lumotlarini jo'natishda POST usuli qo'llaniladi-->
Ismingiz<br><input type="text" name="first_name"
value="Ismingizni kiriting"><br>
Familiyangiz<br><input type="text" name="last_name"><br>
E-mail <br><input type="text" name="email"><br>

```

```

<P> Siz qatnashmoqchi bo'lgan fanni tanlang:<br>
<input type=checkbox name='fan[]' value='Matematika'> Matematika
<br>
<input type=checkbox name='fan[]' value='Informatika'> Informatika
<br>
<input type=checkbox name='fan[]' value='Tarix'> Tarix <br>
<input type=checkbox name='fan[]' value='Falsafa'>Falsafa<br>

<P> O'zingiz haqingizda qisqacha ... <BR>
<textarea name="comment" cols=32 rows=5></textarea>
<P><input name="confirm" type=checkbox checked>To'ldirganligingizni
tasdiqlang <br>
<input type=submit value="Jo'natish">
<input type=reset value="Tozalash">
</form>

```

Bu yerda faqatgina checkbox elementi qiymatini belgilab qo'yish kerak. checkbox elementi tanlanganda uning qiymatlari mos ravishda birinchi elementi fan massivining birinchi elementiga, ikkinchi elementi massivning ikkinchi elementiga va h.k. yoziladi. checkbox elementlariga har xil nomlar ham berish mumkin, lekin bu, agar fanlar ko'p bo'lsa, ma'lumotlarni qayta ishlashini qiyinlashtiradi.

Bularning hammasi muhokama etuvchi va qayta ishlovchi skript test.php deb ataladi (bu formaning action atributida ko'rsatilgan). Ma'lumotlarni jo'natish uchun POST metodi ko'rsatiladi. Ro'yxatdan o'tgan shaxsdan olingan ma'lumotlarga qarab skript mos xabarni ishlab chiqadi. Agar ro'yxatga olingan shaxs qandaydir fanlarni tanlagan bo'lsa, unda unga ushbu fanlarni qachon bo'lishi va ularni o'qiydigan ma'ruzachilar haqida ma'lumot beradi. Agar o'quvchi hech qanday fanni tanlamagan bo'lsa, u holda mashg'ulot fanlari-ning keyingi majlisi haqida xabar beriladi.

4.8-listing. form_yakuniy.html formasini qayta ishlovchi fayl

```

<?
// mos holda mashg'ulot o'tish vaqtlari va ma'ruzachilar haqidagi
// massivni yaratamiz

```



```

$times= array("Matematika"=>"14.30", "Informatika"=>"12.00", "Ta-
rix"=>"15.00", "Falsafa"=>"14.00");
$selectors=array("Matematika"=>"Sayfulla                      Dottoyev",
"Informatika"=>"Azim      Karimov",      "Tarix"=>"Farhod      Sobirov",
"Falsafa"=>"Baxtiyor Sodiqov");
define("SIGN", "Hurmat bilan ma'muriyat");
// Xat imzosini konstanta sifatida aniqlaymiz
define("MEETING_TIME", "18.00");
// Talabalar yig'ilishi vaqtini beramiz
$date = "20 may"; // ma'ruzalar o'tish vaqtini beramiz
//Xabar matnini tuzishni boshlaymiz
$str = "Assalomu alaykum hurmatli " . $_POST["first_name"]
. " " . $_POST["last_name"]."!<br>";
$str .= "<br>Sizga shuni ma'lum qilamizki ";
$fans=$_POST["fan"]; // bu o'zgaruvchiga tanlangan
// fanlar ro'yxatini saqlaymiz
if (!isset($fans)) { // agar faqat bitta fan tanlanmagan bo'lsa
    $sevent = " talabalarning keyingi yig'ilishi";
    $str .= "$sevent $date ". MEETING_TIME . " bo'lib o'tadi <br>";
} else { // hech bo'lmaganda bitta fan tanlanganda
    $sevent = "Siz tanlagan ma'ruzalar $date sanasida bo'lib o'tadi. <ul>";
    //count funksiyasi massivdagi elementlar sonini aniqlaydi
    for ($i=0;$i<count($fans);$i++){
        // har bir tanlangan fan uchun
        $k = $fans[$i]; // fan nomini eslab qolamiz
        $select = $select . "<li> $k fani bo'yicha ma'ruza $times[$k] da";
        // xabar tuzamiz
        $select .= " (Sizning ma'ruzachingiz, $selectors[$k])";
    }
    $sevent = $sevent . $select . "</ul>";
    $str .= "$sevent";
}
$str .= "<br>". SIGN; // Imzo qo'yamiz
echo $str; // ekranga xabar chiqaramiz
?>

```

V bob. BOSHQARUVCHI KONSTRUKSIYALAR

Bu bobda tanlash operatorlari (if, else — shartli o'tish operatorlari, switch-case qayta ulovchi operatorlar), takrorlash operatorlari (while, for, foreach), o'tish operatorlari (continue, break) va qamrab olish operatorlari (include, require) kabilar haqida so'z boradi.

Boshqaruvchi konstruktsiyalar boshqacha qilib aytganda, operatorlar quyidagi turlarga bo'linadi:

— Tanlash operatorlari — shartli o'tish operatorlari (if-else) va qayta ulovchi (switch-case) operatorlar.

— Takrorlash (sikl) operatorlari — sharti oldindan berilgan operator (while), parametrli takrorlash operatori (for) va sharti keyin keluvchi operatorlar (do...while).

— O'tish operatori — boshqarishning shartsiz uzatishini ta'minlaydi. Ularga continue (takrorlashning hozirgi iteratsiyasini yakunlash), break (takrorlashdan chiqish) va return (funksiyalardan chiqish)lar kiradi.

— Qamrab olish operatorlari (include, require).

5.1. Tanlash operatorlari

if, else — shartli o'tish operatorlari

Boshqa tillardagi kabi if shartli o'tish operatori PHPning asosiy operatorlaridan biri. IF operatori shartning berilishiga qarab, kod blokini bajaradi. If shartli o'tish operatorining sharti () qavs ichiga olib yoziladi.

Sintaksisi:

if (shart) bajarish_bloki;

bu yerda shart biror-bir mazmunga ega bo'lib, skriptni qayta ishlash jarayonida mantiqiy qiymatga o'zgartiriladi. Agar shart bajarilsa, true(rost) qiymatni qabul qiladi, ya'ni shart mantiqiy true(rost) qiymatga o'zgaradi va keyingi qadam (bajarish_bloki) bajariladi. Aks holda bajarish_bloki bekor qilinadi.

Shartlarning mantiqiy qiymatlarga aylanish holatlari

1. False qiymatga o'zgaradigan holatlar:

Mantiqiy false

Butun nol (0)

Bo'sh qator va "0" qatori

Elementlarsiz massiv
O'zgaruvchilarsiz obyekt
Maxsus NULL

2. Qolgan barcha holatlar True qiymatga o'zgaradi.

5.1-misol. If shartli o'tish operatori

Ma'lumot kiritish uchun forma berilgan. Ismi kiritilganmi yoki yo'qligi tekshiriladi va mos javob qaytariladi. If shartli o'tish operatori shart yordamida foydalanuvchi logini kiritilgan yoki kiritilmaganligini tekshiradi va mos xabarni chiqazadi.

```
<? if (!$name) echo "Ismingizni kiritmadingiz!"; ?>
```

Bu misolda !\$name sharti ishlatilmoqda.

Agar \$name o'zgaruvchisining qiymati bo'sh bo'lsa, !\$name sharti true(rost) qiymatni qabul qiladi va shart bajarilganligi uchun keyingi qadamga o'tiladi.

IF shartli o'tish operatoridan so'ng faqatgina bittadan ortiq bo'lmagan bajarish_bloki bajariladi. Agar bajarish_bloki bir nechta buyruq satridan iborat bo'lsa, har bir buyruq satri oldidan if shartli o'tish operatori qo'llaniladi yoki buyruq satrlari {} qavs ichiga olib yoziladi, ya'ni bir blokka birlashtiriladi.

```
<? if (!$name)
{ echo "<font color=red>";
echo "Ismingizni kiritmadingiz";
echo "</font>"; } ?>
```

Else operatori

Odatda faqatgina shart bajarilgan holatlar qaralmaydi, balki u bajarilmay qolgan holatlarda ham biror-bir xulosa chiqariladi. Shart bajarilmagan holda, qandaydir harakatni bajarish uchun if operatori Else operatori yordamida kengaytiriladi.

If ning Else operatori yordamida kengaytirilgan sintaksisi:

```
if(shart) bajarish_bloki
else bajarish_bloki1
```

Agar shart bajarilgan bo'lsa, ya'ni shart true(rost) qiymat qabul qilsa, bajarish_bloki bajariladi, aks holda bajarish_bloki1 bajariladi.

if operatorini yakka holda yoki else operatori bilan birga ishlatish mumkin. Lekin else operatorini yakka holda ishlatish mumkin emas, faqatgina if operatori bilan birgalikda ishlatiladi.

5.2-misol. Berilgan sonning juft-toqligini tekshirish.

```
<? if ($a%2 == 0)
{ echo "$a - juft son<br>"; }
else { echo "$a - toq son<br>"; } ?>
```

5.3-misol. Talaba imtihon topshirgan yoki topshirmaganligi tekshiriladi. Agar topshirgan bo'lsa, o'tgan yoki o'tmaganligi tekshiriladi.

```
<? if ($natija == "") { echo "Talaba imtihon topshirmadi"; }
else { if ($natija<55) echo "Talaba imtihondan o'tmadi";
if ($natija>55) echo "Talaba imtihondan o'tdi"; } ?>
```

Demak, bu misolda birinchi if operatoridagi shart bajarilmagan holda ikkinchi shart uchun yana if operatori qo'llanilgan.

Elseif operatori

If operatorining yana bir kengaytmasidan biri bu elseif operatoridir. Elseif — bu else va if ning kombinatsiyasidir.

Sintaksisi:

If (shart) bajarish_bloki

Elseif (shart1) bajarish_bloki1

...

Else bajarish_blokiN

Bitta if blokda bir necha marta elseif operatorlari qatnashishi mumkin. Agar if sharti false qiymat qabul qilsa, keyingi elseif shartlar tekshiriladi. Elseif shartlarining birortasi true qiymat qabul qilsa, mos bajarish_bloki bajariladi, aks holda keyingi qadamga o'tiladi.

5.3-misolni boshqacharoq shaklda elseif operatori yordamida yozishimiz ham mumkin.

```
<? if ($natija=="") echo "Talaba imtihon topshirmadi";
elseif ($natija<55) echo "Talaba imtihondan o'tmadi";
elseif ($natija>55) echo "Talaba imtihondan o'tdi"; ?>
```

Shartli o'tish operatorlarida shart o'rnida bir nechta shartni ham tekshirish mumkin. Bu shartlar VA (AND yoki &&) va YOKI (OR yoki ||) mantiqiy amallari yordamida bog'lanadi. Quyidagi misolda VA mantiqiy amalini ishlatamiz.

5.4-misol. Berilgan o'zgaruvchilar qiymatlarini solishtirish.

```
<? $a=1; $b=3;
$c=4; $d=9;
if ($a!=$b and $c!=$d){ echo "O'zgaruvchi qiymatlari teng emas!"; }
else { echo "O'zgaruvchi qiymatlari teng!"; } ?>
```


Natija:O'zgaruvchi qiymatlari teng emas!

Bu misolda ikkala shartda ham o'zgaruvchi qiymatlari bir-biriga teng bo'lmaganligi, ya'ni shart bajarilgani uchun keyingi qadam bajarildi.

5.5-misol. Berilgan o'zgaruvchilar qiymatlarini solishtirish.

```
<? $a=1; $b=3;  
$c=9; $d=9;  
if ($a!=$b and $c!=$d)  
{ echo "O'zgaruvchi qiymatlari teng emas! "; }  
else { echo "O'zgaruvchi qiymatlari teng!"; } ?>  
Natija:O'zgaruvchi qiymatlari teng!
```

Bu misolda and amali qatnashganligi uchun ikkala shart ham bajarilgandagina keyingi qadam bajariladi, aks holda else operatori-dan keyingi qadam bajarilishi kerak. Misolni tekshiramiz, birinchi shart bajarildi, lekin ikkinchi shart bajarilmadi, shuning uchun else operatoridan keyingi qadam bajarildi.

Agar shartlar orasidagi and mantiqiy amal o'rniga or mantiqiy amalni ishlatsak natija o'zgaradi.

5.6-misol. Berilgan o'zgaruvchilar qiymatlarini solishtirish.

```
<? $a=1; $b=3; $c=9; $d=9;  
if ($a!=$b or $c!=$d){ echo "O'zgaruvchilar qiymatlari teng emas!"; }  
else { echo "O'zgaruvchilar qiymatlari teng!"; } ?>  
Natija:O'zgaruvchilar qiymatlari teng emas!
```

Bu misolda esa YOKI (or yoki ||) amali qatnashganligi uchun shartlarning birortasi rost qiymat qabul qilsa ham keyingi qadam bajariladi.

Yuqoridagi misollarda o'zgaruvchi qiymatlarini taqqoslash uchun or va and mantiqiy amallari bilan birgalikda = va != amallaridan tashqari boshqa taqqoslash amallarini: <, >, <=, >= va h.k. larni ham ishlatish mumkin.

5.7-misol. Talabalarni baholash mezonlari sxemasini tuzib chiqamiz.

55 dan past ball to'plagan talabalar – o'zlashtira olmaganlar.

55 dan yuqori 71 dan past ball to'plagan talabalar – 3 bahoga o'zlashtiruvchilar.

71 dan yuqori 86 dan past ball to'plagan talabalar – 4 bahoga o'zlashtiruvchilar.

86 dan yuqori ball to'plagan talabalar — 5 bahoga o'zlashtiruvchilar.

Bu misolni PHP kodini if va elseif operatorlari va &&(yoki AND) amali yordamida yozamiz.

```
<? if ($baho<55) echo "O'zlashtira olmagan talabalar!";  
elseif ($baho>=55 && $baho<71)  
echo "3 bahoga o'zlashtirgan talabalar!";  
elseif ($baho>=71 && $baho<86)  
echo "4 bahoga o'zlashtirgan talabalar!";  
elseif ($baho>=86) echo "5 bahoga o'zlashtirgan talabalar!"; ?>
```

E'tibor qarating! elseif operatorini else if shaklida ham yozish mumkin, ikkala holat ham to'g'ri hisoblanadi.

Ichma-ich joylashgan IF-ELSE operatori

Ichma-ich joylashgan shartlar — bu if shartli o'tish operatori ichida yana bir if shartli o'tish operatorining ishlatilishidir. Bu holda shart tekshirilgandan so'ng, bajarilishi kerak bo'lgan blok ichida biror-bir amal emas, balki yana bir shart tekshiriladi.

5.8-misol. Ichma-ich joylashgan shartlarning ishlatilishi

```
<? $a=1; $b=3;  
$c=9; $d=9;  
$talaba1="bakalavr"; $talaba2="magistr";  
if ($a==$b or $c==$d) { echo "O'zgaruvchilar qiymatlari teng!";  
if($talaba1==$talaba2)  
{ echo "talaba1 va talaba2 ikki xil yo'nalish talabasi!"; }  
else { echo "O'zgaruvchilar qiymatlari teng emas!"; } }  
Natija: O'zgaruvchilar qiymatlari teng! talaba1 va talaba2 ikki xil yo'nalish talabasi!
```

Bu misolni tahlil qilamiz. Avval o'zgaruvchi qiymatlari beriladi, so'ng If operatori yordamida shart tekshiriladi, shart bajarilsa, keyingi qadam bajariladi va shu bilan birga ikkinchi shart ham tekshiriladi, ikkinchi shart ham bajarilsa, keyingi qadamga o'tiladi.

switch-case qayta ulovchi operatori

Shartlarni tekshirish va tekshirish natijalariga ko'ra turli amallarni bajaruvchi operatorlardan biri switch operatoridir. O'zgaruvchi

qanday shartga ega ekanligiga qarab, mos turli bajarish_bloklari bajariladi.

switch operatori if va if...elseif operatoriga o'xshash, faqatgina undan farqli tomoni bu operatorida shart ikkitadan ortiq qiymat qabul qiladi. If operatorida shart true va false qiymatlarni qabul qiladi. switch operatorida esa shart oddiy tip (integer, string yoki double) bo'lgan ixtiyoriy turli qiymatlarni qabul qiladi.

Sintaksisi:

```
switch (shart yoki o'zgaruvchi) {  
  case qiymat1: bajarish_bloki1 break;  
  case qiymat2: bajarish_bloki2 break;  
  case qiymat3: bajarish_bloki3 break;  
  case qiymat4: bajarish_bloki4 break;  
  default: birorta ham holat yuzaga kelmagandagi bajarish_bloki  
  break;}
```

Bu yerda shart case(qiymat1, qiymat2, qiymat3, ...) dagi qiymatlar bilan solishtiriladi. Agar shart yoki o'zgaruvchi qiymat1, qiymat2, qiymat3, ... larning qaysidir biri bilan mos tushsa, qiymatdan keyingi ikki nuqtadan birinchi break operatorigacha (agar bo'lsa) yoki switch oxirigacha mos bajarish_bloki bajariladi. Agar shart yoki o'zgaruvchi qiymat1, qiymat2, qiymat3, ... larning bi-rortasi bilan mos tushmasa default kalit so'zidan keyingi bajarish_bloki bajariladi.

5.9-misol. Talabaning ismi, familiyasi, guruhi, to'lov-kontraktni amalga oshirish yo'lini tanlash imkoni bo'lgan maydonlardan iborat forma yaratamiz. Talaba ushbu formani to'ldirib, ma'lumotlarni formani qayta ishlovchi sahifaga jo'natadi. Formani qayta ishlovchi sahifada tanlash operatorlarini ishlatishni ko'rib o'tamiz.

```
<tr>  
<td>To'lov kontraktni qaysi yo'l bilan amalga oshirmoqchisiz?</td>  
<td><select name="usul">  
  <option value = "plastik_kartochka">Plastik kartochka yordamida...  
  <option value = "pul_o'tkazish">Pul o'tkazish yo'li bilan...  
  <option value = "naqd">Naqd to'lov orqali... </select></td></tr>
```

5.1-rasmda berilgan HTML forma kodining qismi berilgan.

Ma'lumotlarni kiriting

Ismingiz

Familiyangiz

Guruhingiz

To'lov kontraktini qaysi yo'l bilan amalga oshirmoqchisiz?

Plastik kartochka yordamida...

Pul o'tkazish yo'li bilan...

Naqd to'lov orqali...

5.1-rasm. To'lov-kontraktini amalga oshirish yo'lini tanlash formasi

Yuqorida berilgan kodni qayta ishlash uchun switch-case operatoridan foydalanamiz.

```
<? $usul="naqd";
switch ($usul) {
case "plastik_kartochka":
echo "<p>Plastik kartochka yordamida..."; break;
case "pul_o'tkazish":
echo "<p>Pul o'tkazish yo'li bilan..."; break;
case "naqd":
echo "<p>Naqd to'lov orqali..."; break;
default:
echo "<p>Talaba birorta ham yo'lni tanlamadi "; break; } ?>
Natija: Naqd to'lov orqali...
```

Agar \$usul o'zgaruvchisi birorta ham qiymatga teng bo'lmasa, u holda natija o'zgaradi, ya'ni "Talaba birorta ham yo'lni tanlamadi" yozuvi ekranga chiqaziladi.

Misolning natijasi ikkala holat (if-else va switch-case)da ham bir xil bo'ladi. Faqatgina if-elseif operatorlari yordamida yozilgan vaqtda, har bir qatordagi shart tekshiriladi. switch operatori esa

boshqacharoq tarzda ishlaydi. switch-case operatorlari ishga tushganda, shart {} blok oxirigacha tekshiriladi. Agar break operatori qo'llanilsa, unga duch kelganda, tekshirish to'xtatiladi.

5.2. Takrorlash (sikl) operatorlari

PHP ning faqatgina biror-bir shart asosida ishlovchi operatorlari emas, balki takrorlanish amallarini bajaruvchi sikllar ham mavjud. Ular while, do-while, foreach va for.

Takrorlash (sikl) operatori ikki qismdan iborat: tana va bosh qismdan iborat. Tana qismi takrorlashda bajariladigan operatorlarni o'z ichiga oladi. Bosh qism esa tanadagi operatorlarning takroriy bajarilishini ta'minlaydi.

While sikli

Sintaksisi:

```
while (shart ) {bajarish_bloki}
```

yoki

```
while (shart) bajarish_bloki endwhile;
```

While — oddiy sikl bo'lib, bu yerda bajarish_bloki shart false qiymat qabul qilgunga qadar bajariladi. While siklining if dan farqli tarafi shundaki, if operatorida shart tekshirilganda true qiymatni qabul qilsa, u holda keyingi qadam faqatgina bir marotaba bajarilar edi. While siklida esa har bir shart bajarilgan holat uchun bajarish_bloki bajariladi, ya'ni bajarilish sikli takrorlanadi. Shart har safar sikl boshida tekshiriladi, shuning uchun agar bajarish_bloki amalga oshirilayotgan vaqtda shartga o'zgartirish kiritilsa ham sikl iteratsiya oxirigacha, ya'ni bajarish_blokining barcha buyruqlari bajarilguncha to'xtatilmaydi.

5.10-misol. 1 dan 50 gacha bo'lgan sonlarning yig'indisini hisoblang.

```
<? $sum=0; $i=1;
while($i<=50) { $sum=$sum+$i; $i++; }
echo "1 dan 50 gacha bo'lgan sonlar yig'indisi - $sum"; ?>
Natija:1 dan 50 gacha bo'lgan sonlar yig'indisi - 1275
```

do...while sikli

do...while sikli while sikliga juda o'xshash, faqatgina bitta farqli tarafi shundaki, unda shartning rostligi sikl boshida emas, balki ox-

irida tekshiriladi. Shuning uchun do...while siklining bajarish_bloki hech bo'lmaganda bir marotaba bajariladi.

Sintaksisi:

do (bajarish_bloki) while (shart);

5.11-misol. do...while operatori

1 dan 10 gacha bo'lgan juft sonlarni chiqazing.

```
<? $n=1;
do { if ($n%2==0) echo $n. "<br>"; $n++; }
while ($n<=10) ?>
```

FOR sikli Sintaksisi:

for (amall; shart; amal2){bajarish_bloki;}

amall — faqatgina bir marotaba sikl boshida bajariladi. Odatda, bunda hisoblagichning boshlang'ich qiymati beriladi.

shart — shart har bir iteratsiyada tekshiriladi. Agar shart false qiymatni qabul qilsa, u holda iteratsiya to'xtatiladi. Odatda, shartda hisoblagich qiymati bilan berilgan qiymat solishtiriladi.

amal2 — hisoblagich qiymatini o'zgartirish uchun ishlatiladi va iteratsiya oxirida bajariladi.

bajarish_bloki — har bir iteratsiyada bir marotaba bajariladi. Odatda, bu yerda o'zida kodlar blokini jamlagan sikl tanasi joylashadi.

amall, shart, amal2 larning har biri bo'sh bo'lishi mumkin. Agar shart bo'sh bo'lsa, bu siklning aniqlanmagan vaqtda bajarilishi kerakligini bildiradi (bu holda php shartni hamma vaqt rost deb hisoblaydi). Bu sikl foydasizdek ko'rinishi mumkin, aslida esa bu hollarda siklni break operatorini qo'llash bilan ham to'xtatish mumkin.

5.12-misol. 1 dan 10 gacha bo'lgan toq sonlarni chiqazing.

```
<? for ($n=1; $n<=10; $n++)
{ if ($n%2!=0) echo $n. "<br>"; } ?>
```

Shu misolni \$n<=10 shartni ishlatmasdan, break operatorini qo'llab yozish mumkin.

```
<? for ($n=1;;$n++)
{ if ($n%2!=0) echo $n. "<br>";
if ($n>=10) break; } ?>
```

5.13-misol. 1 dan 50 gacha bo'lgan sonlarning yig'indisini hisoblang.


```
<? for($i=1;$i<=50;$i++) { $sum=$sum+$i; }  
echo "1 dan 50 gacha bo'lgan sonlar yig'indisi - $sum"; ?>  
Natija:1 dan 50 gacha bo'lgan sonlar yig'indisi - 1275
```

foreach

Bu operator massivlar bilan ishlash uchun mo'ljallangan bo'lib, PHP ning 4 versiyasida kiritilgan. Turli tipdagi massivlar uchun foreach operatorining ikki xil ko'rinishi mavjud.

Sintaksisi:

```
foreach ($array as $value) {bajarish_bloki}
```

yoki

```
foreach ($array as $key=> $value) {bajarish_bloki}
```

Birinchi navbatda \$array o'zgaruvchisiga berilgan massivning hamma elementlari bo'yicha sikl shakllanadi. Siklning har bir qadamida massivning joriy elementi qiymati \$value o'zgaruvchisiga yoziladi va massivning ichki hisoblagichi bir birlikka o'zgaradi (shuning uchun keyingi qadamda massivning keyingi elementi ko'rinadi). bajarish_bloki ichida massivning joriy elementi qiymati \$value o'zgaruvchisi yordamida olinadi va \$array o'zgaruvchisida nechta element bo'lsa, bajarish_bloki shuncha marta amalga oshiriladi.

foreach massivining ikkinchi tur sintaksisida esa, yuqoridagiga qo'shimcha ravishda, siklning har bir qadamida massivning joriy elementi kaliti \$key o'zgaruvchisiga yoziladi va uni ham bajarish_bloki ichida qo'llash mumkin.

foreach o'z ishini boshlaganda, massivning ichki ko'rsatkichi avtomatik tarzda birinchi elementga o'rnatiladi.

5.14-misol. Foreach operatori

```
<? $names = array("Lola", "Karim", "Kamol");  
foreach ($names as $val) {  
echo "$val <br>"; } ?>  
Natija:Lola  
Karim  
Kamol
```

Ikkinchi turi assotsiativ massivlar uchun ishlatiladi.

5.15-misol. Foreach operatori

```
<? $names = array( "Lola"=>7, "Karim"=>9, "Kamol"=>15);  
foreach ($names as $i=>$tartib_nomer)  
{ echo "$i ro'yxatda $tartib_nomer nomerida.<br>"; } ?>
```

Natija:

Lola ro'yxatda 7 nomerida.

Karim ro'yxatda 9 nomerida.

Kamol ro'yxatda 15 nomerida.

5.3. O'tish operatorlari

Ayrim hollarda sikl ishini tezda to'xtatish yoki biror fragmentdan chiqib ketish talab etiladi. Yoki siklning oxirgi instruksiyasidan keyin joylashgan dastur birinchi instruksiyaga boshqaruvni uzatishi kerak. Buning uchun break va continue operatorlaridan foydalaniladi.

Break (sikldan chiqish)

Break operatori joriy siklning (for, foreach, while, do...while, switch) ishini to'xtatadi. break operatori sonli parametr bilan birga qatnashgan hollarda, sonli parametrga qarab boshqaruvchi kerakli struktura ishini to'xtatadi (yoki sikldan chiqadi).

5.16-misol. break operatori

```
<? $i=1;
while ($i) {
    $n = rand(1, 10); // 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
    echo "$i:$n"; // iteratsiya tartibi va tanlangan sonlarni chiqaradi
    if ($n==5) break;
    /* Agar 5 soni tanlangan bo'lsa sikl ishini tugallaydi. Ushbu qatordan
    keyin bajarilishi kerak bo'lgan kodlar ishlamaydi */
    echo "Sikl ishlayapti <br>";
    $i++; }
echo "<br>sikl iteratsiyalar soni $i"; ?>
```

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo'lishi mumkin.

Agar break operatoridan keyin son ko'rsatilsa, u holda shu songa teng miqdordagi ushbu operator qatnashgan sikllar ishi to'xtatiladi. 5.16-misol bu holatga misol bo'la olmaydi, chunki unda ichma-ich joylashgan sikllar mavjud emas. Skriptga o'zgartirish kiritamiz:

```
<? $i=1;
while ($i) { $n = rand(1, 10);
// 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
switch ($n){
```



```

case 5: echo "<font color=blue> switch (n=$n)</font> dan chiqish";
break 1;
// switch ishini to'xtatamiz (break qatnashgan birinchi siklni)
case 10: echo "<font color=red> switch dan chiqish va while (n=$n)</
font>";
break 2;
// switch va while ishini to'xtatamiz (break qatnashgan ikkala siklni)
default: echo "switch ishlayapti (n=$n), "; }
echo " while ishlayapti - $i qadam <br>";
$i++; }
echo "<br> Sikl iteratsiyalar soni $i "; ?>

```

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo'lishi mumkin.

Continue (siklni davom ettirish)

Ayrim hollarda sikl ishini butunlay to'xtatmasdan, siklni yangi iteratsiyadan boshlashga to'g'ri keladi, bu holda continue operatorini qo'llab yangi iteratsiyani boshlash mumkin. continue operatori bajarish_blokining istalgan siklidan keyingi instruksiyalar ishlashini to'xtatib, yangi doirada bajarishni davom ettirishni ta'minlaydi. Continue operatorini sonli parametr bilan qo'llash mumkin va ushbu parametrda nechta ichma-ich joylashgan boshqaruvchi konstruksiyaning ishini tugatish kerakligi ko'rsatiladi. Masalan, ikkita ichma-ich joylashgan sikllar ichida Continue 2 buyrug'i berilsa, boshqaruv ikkala sikl ishlashini to'xtatadi.

5.16-misoldagi break operatorini continuega almashtiramiz va sikl qadamlari sonini 4 ta bilan to'xtatamiz.

```

<? $i=1;
while ($i<4) { $n = rand(1, 10);
// 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
echo "$i:$n"; // iteratsiya generizatsiyalagan sonni chiqaramiz
if ($n==5) { echo "<br> Yangi iteratsiya <br>"; continue;
/* agar 5 soni generizatsiya qilingan bo'lsa, siklning yangi iteratsiyasini
boshlaymiz $i o'zgarmaydi */ }
echo "sikl ishlayapti <br>"; $i++; }
echo "<br>". $i ." - siklning iteratsiyalar soni"; ?>
Natija:
1:4sikl ishlayapti

```



```
2:4sikl ishlayapti
3:5
Yangi iteratsiya
3:8sikl ishlayapti
4-siklning iteratsiyalar soni
```

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo'lishi mumkin.

Continue operatori bajarilgandan so'ng sikl ishi to'xtamaydi. Bu yerda sikl hisoblagichi 5 sonini olishda o'zgarmaydi, chunki u continue operatoridan keyin joylashgan. Aslida biz continue yordamida 5 soni paydo bo'lgan holdan qochishga harakat qilamiz. Shuning uchun continue operatorini quyidagi almashtirish bilan yozib ko'ramiz:

```
<? $i=1;
while ($i<4) { $n = rand(1, 10);
// 1 dan 10 gacha bo'lgan sonlarni tanlash
if ($n!=5) { echo "$i:$n <br>";
// iteratsiya va generizatsiya qilingan sonni chiqaramiz
$i++; } } ?>
Natija:
1:2
2:1
3:7
```

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo'lishi mumkin.

PHP da continue operatori qo'llanilishining bitta asosiy tarafi mavjud — u switch konstruksiyalarida xuddi break dagidek ishlaydi. Agar switch sikl ichida joylashgan bo'lsa va yangi iteratsiyani boshlash kerak bo'lsa, continue 2 ni qo'llash kerak.

5.4. Qamrab olish operatorlari

Oddiy, lekin juda ko'p ishlatiladigan operatorlar: require(), include() larni ko'rib o'tamiz. require(), include() operatorlari yordamida PHP sahifaga faylni yuklab olinadi. Fayl o'zida matnlarni, HTML deskriptorlarni, PHP klasslarni, PHP funksiyalarni, PHP operatorlarni jamlagan bo'ladi.

WEB sayt yaratishda faylni yuklab olish nima uchun kerak? Sayt yaratilayotgan vaqtda saytni bloklarga ajratib dastur kodlarini yozish

maqsadga muvofiq. Masalan, saytni yuqori qismi bloki, menyu qismi bloki, asosiy qismi bloki, o'ng tomoni bloki va shunga o'xshash. Bloklarga ajratib yozish, saytni biror-bir qismiga o'zgartirish kiritishda qulay bo'lib (masalan, yangi menyu qo'shish yoki mavjud menyulardan birortasini olib tashlash), u holda faqatgina o'sha blokka o'zgartirish kiritiladi, xolos. Ana shu bloklar fayllarda saqlanadi va ushbu fayllarni require(), include() operatorlari yordamida yuklab olish mumkin. require(), include() operatorlari bir-biriga o'xshash holatda ishlaydi.

include() operatori

Include operatori ko'rsatilgan fayldagi barcha kodlarni ochish uchun ishlatiladi va dasturda bu operator necha marta ishtirok etsa, u shuncha marta bajariladi.

Sintaksisi:

include 'fayl_nomi';yoki

include \$fayl_nomi;yoki

include ("fayl_nomi");

Masalan, form.php faylida funksiya yoki o'zgaruvchi qiymatlari joylashgan bo'lsin. Asosiy dastur kodini biror qismida ushbu funksiya yoki o'zgaruvchi qiymatlaridan foydalanish kerak bo'lsa include("form.php") ni qo'yamiz. Agar bir necha marta form.php dan foydalanish kerak bo'lsa, u holda asosiy dasturning kerakli qismida bir necha marta qo'llash mumkin. Sahifa bajarilayotgan vaqtda include() operatori include("form.php"); buyrug'i o'rniga form.php faylidagi mavjud kodlarni almashadi.

5.17-misol. include() qamrab olish operatorining ishlatilishi form.php fayli

```
<? $name="Nizomiy nomli TDPU";  
$bugun=date("d.m.y"); ?>
```

main.php fayli

```
<? include("form.php");  
echo "Ma'lumotingiz $name ning pochta adresiga yuborildi.<br>";  
echo "Bugun $bugun"; ?>
```

Natija shuni ko'rsatadiki, main.php ishga tushirilganda sahifa quyidagi ko'rinishga ega bo'ladi.

```
<? $name="Nizomiy nomli TDPU";  
$bugun=date("d.m.y");
```

```
echo "Ma'lumotingiz $name ning pochta adresiga yuborildi.<br>";  
echo "Bugun $bugun";  
?>
```

Natija:

Ma'lumotingiz Nizomiy nomli TDPU ning pochta adresiga yuborildi.
Bugun 26.01.11

Fayl kengaytmasi va include() operatori

PHP ning include() operatori murojaat etilayotgan faylning kengaytmasi ahamiyat qaratmaydi. Faylning nomi qulay tarzda nomlanishi mumkin. include() operatori yordamida yuklanayotgan fayl PHP faylining bir qismi sifatida bo'ladi va qanday bo'lsa, shunday bajariladi. Qamrab olinayotgan (yuklanayotgan) fayl uchun ixtiyoriy kengaytmani qo'llash mumkin, shulardan biri .inc. Agar fayl .inc kengaytmasi yoki birorta boshqa standart bo'lmagan kengaytma bilan WEB hujjatlar katalogida joylashgan bo'lsa va foydalanuvchi bu faylni brauzer yordamida ishga tushirsa, u holda fayldagi kodlarni oddiy matn shaklida ko'rish mumkin, ya'ni fayl qayta ishlanmaydi. Shuning uchun murojaat etiladigan fayllarni biror-bir katalogda saqlash tavsiya etiladi yoki standart kengaytmadan foydalanish kerak bo'ladi.

PHP operatorlari, masalan, page.html kabi nomlangan fayllarda ishtirok etsa bajarilmaydi. Aksincha, agar include() operatori yordamida page.html fayli yuklab olinsa, fayl ichida saqlanayotgan PHP operatorlari qayta ishlanadi. Qoidaga ko'ra, PHP faqatgina .php kabi kengaytmaga ega fayllardagina operatorlariga murojaat etib, boshqa fayllarni tahlil qiladi.

Demak, form.php dagi dastur kodi main.php faylining bir qismiga ekvivalent. form.php faylini PHP kod ekanligini anglatuvchi teglarsiz oddiy matn sifatida yozish mumkinmi? Yo'q, chunki faylni qo'yish vaqtida PHP ni qayta ishlash rejimidan HTML rejimiga o'tadi. Shuning uchun yuklanayotgan fayl ichidagi PHP skripti ko'rinishida qayta ishlanadigan kod mos teglar bilan yozilishi kerak.

Yuklanayotgan fayllarni qidirish quyidagi tartibda amalga oshiriladi:

1. Faylni qidirish avval joriy ish katalogiga tegishli include_path da olib boriladi.

2. Agar fayl topilmasa, u holda qidiruv joriy skript katalogiga tegishli `include_path` da amalga oshiriladi.

3. PHP sozlovlar faylida aniqlangan `include_path` parametrda yuklanadigan fayllarni qidirish mumkin bo'lgan kataloglar nomi beriladi.

Masalan, bizning misolda joriy ish katalogi bu — `/www/` katalogidir. Asosiy fayl `main.php` fayliga `include/a.php` fayli va navbati bilan `b.php` fayli yuklanayotgan bo'lsa, parser birinchi o'rinda `b.php` faylini `/www/` katalogidan qidiradi va agar bunday fayl topilmasa, u holda keyin `/www/include/` katalogidan qidiradi.

`include()` yordamida yuklanayotgan fayl tarkibidagi o'zgaruvchilar `include()` joylashgan sohaga bog'liq ravishda ishlatiladi. Yuklanayotgan fayldagi ixtiyoriy o'zgaruvchi yuklanayotgan faylni o'zida ishlatilishi mumkin. Agar `include()` funksiya ichida yuklansa, u holda yuklangan fayl tarkibidagi kod funksiya ichida aniqlanganlar kabi bo'ladi.

`include()` operatori yordamida lokal fayllardan tashqari url — manzillari ko'rsatilgan tashqi fayllarni ham yuklash mumkin. Ushbu imkoniyat PHP sozlov faylidagi `url_fopen_wrappers` direktivasida nazoratga olinadi va odatda bu yoqilgan bo'ladi. Lekin windows uchun PHP ning 4.3.0 versiyasigacha bu imkoniyat mavjud bo'lmagan.

`include()` bu maxsus konstruksiya, shuning uchun ichki shartli bloklar qo'llanilishida uni figurali qavs ichiga olib yoziladi.

```
<? if ($value=true) { include("file1.inc"); }  
Else { include("file2.inc"); } ?>
```

`include()` operatorini qo'llashda ikki xil turdagi xatolik yuzaga kelishi mumkin — yuklash xatoligi (masalan, yuklash buyrug'ini o'zi noto'g'ri yozilgan bo'lsa, ko'rsatilgan faylni topish mumkin bo'lmasa va h.k.) yoki bajarilish xatoligi (agar yuklanayotgan faylda xato mavjud bo'lsa).

require() operatori

`require()` va `include()` operatorlari juda o'xshash bo'lib, ishlash prinsipiga ko'ra bir-biridan farq qiladi. `require()` operatori taxminan C++ dagi `# include` kabi ishlaydi. `include()` uchun aytilganlarning barchasi `require` uchun ham o'rindidir. `require` shuningdek dasturga bog'lanishga va biror-bir faylni amalga oshirishga imkon yaratadi. `require` va `include` ning asosiy farqi uning xatolarga qanday reaksiya

berishidir. include ogohlantiradi va script ishini davom ettiraveradi. require da esa xato skript ichida fatal xatoni chiqaradi va uning bajarilishini to'xtatadi. Savol tug'ilishi mumkin: u holda require() operatori nima uchun qo'llaniladi, uning include() operatoridan qanday farqi bor? Masala shundaki, require() operatori include() operatoridan birmuncha tezroq ishlaydi.

require() operatori shartli operator ichida qo'llanilsa, yaxshi samara bermaydi. Agar hech bo'lmaganda ushbu operator qo'llanilgan qator bajarilmasa, unda yuklangan fayldan bitta qator ham bajarilmaydi. Shuningdek, sikllar ham require() ga ta'sir qilmaydi. Hatto yuklangan fayl sikl obyekti bo'lsa ham, yuklash o'z-o'zidan bir marotabagina amalga oshiriladi.

PHP ning 4.0.2 versiyalarigacha require() ning qo'llanilishi, interpretator albatta yuklanayotgan faylni o'qishga harakat qilishini anglatgan.

require() ni ham include() kabi shartli bloklar ichida qo'llaganda figurali qavs ichiga olib yozish kerak.

```
<? If ($value=true) { require("file1.inc"); }  
Else { require("file2.inc"); } ?>
```

require() operatori shartli bloklar ichida qo'llanilganda, har safar sahifa bajarilganda, barcha fayllar yuklab olinadi, lekin ulardan faqatgina bittasi \$value o'zgaruvchisi qiymatiga qarab bajariladi.

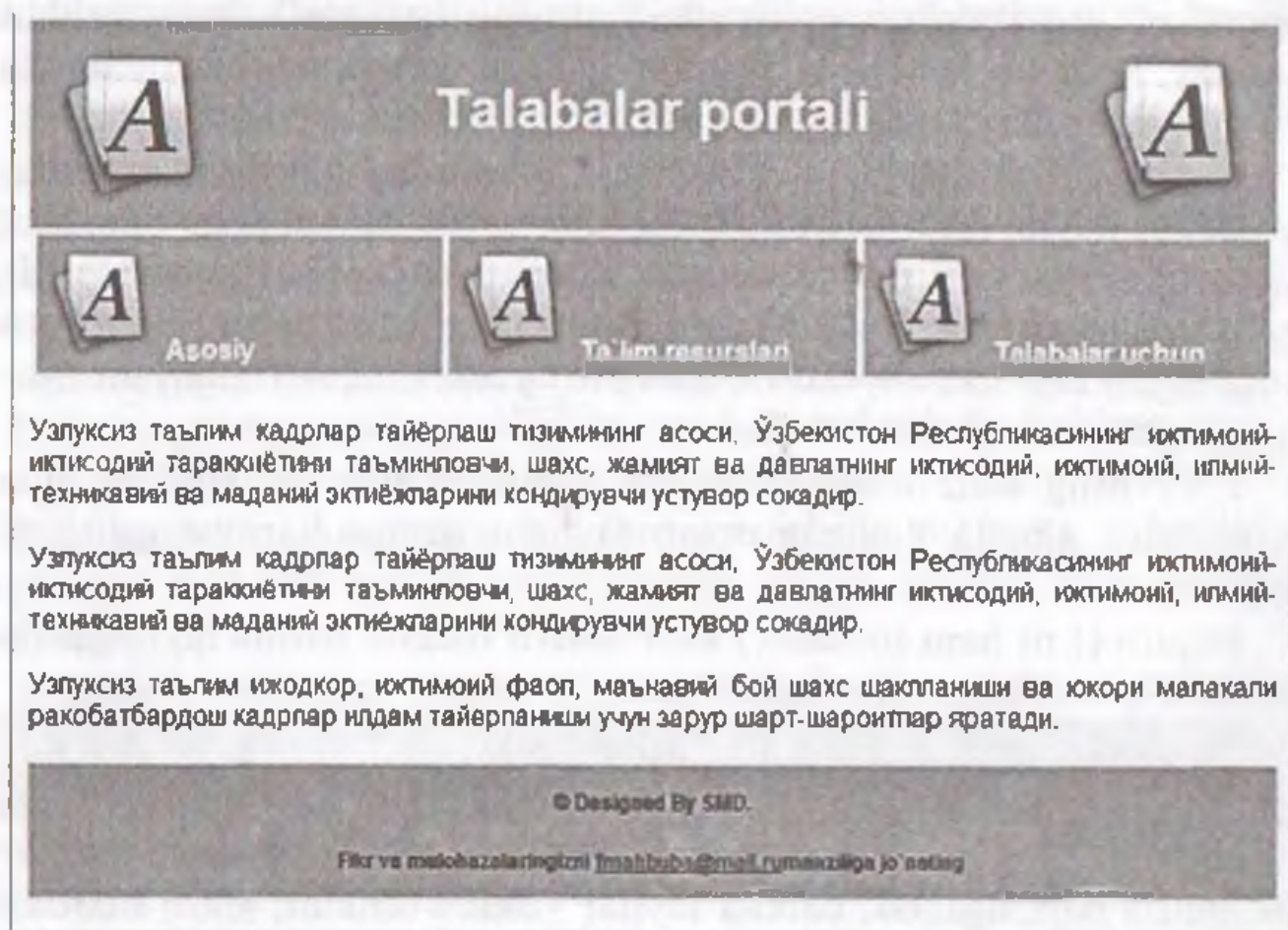
Agar require() operatorining o'rnida include() operatori qo'llanilsa, u holda \$value o'zgaruvchisi qiymatiga qarab faqat bitta fayl yuklab olinadi va bajariladi.

require() operatorining Web sayt shablonlari uchun qo'llanilishi

Web sayt sahifalari bir xil ko'rinishdan iborat bo'lsa, u holda PHP dan foydalanib require() operatori yordamida sahifaga shablon va standart elementlarni qo'shish mumkin.

Masalan, Web sayt 5.2-rasmdagi ko'rinishdagi kabi bo'lsin. Sahifalarda quyidagicha holatlar yuzaga kelishi mumkin: Web sayt ancha vaqtdan beri ishlab kelmoqda va u bir necha o'n, yuz va minglab sahifadan iborat va ularning hammasi bir xil ko'rinishda bo'lishi mumkin. Uning standart ko'rinishiga kichkina o'zgartirish kiritish lozim, masalan, uning quyi qismidagi elektron pochta manzilini o'zgartirish yoki sahifaning menyular satriga yangi menyu qo'shish lozim bo'lsin. U holda o'n, yuz, minglab sahifani o'zgartirish kimga

yoqadi? Shuning uchun sahifani HTML kodini maqbul yo'l bilan qismlarga ajratiladi. 5.2-rasmdagi sahifaning kodi 5.1-listingda berilgan.



5.2-rasm. Talabalar portalining Web sahifasi

5.1-listing. task_form.html – Talabalar portalining asosiy sahifasining HTML kodi.

```
<html> <head> <title>Talabalar portali</title>
<style> <!-- h1{color: white; font-size :24p; text-align : center ; font-
family : arial , sans-serif}
.menu{color: white; font-size: 12pt; text-align : center ; font-family : arial
, sans-serif; font-weight : bold}
td {background: silver}
p {color :darkblue; font-size: 12pt; text-align: justify;font-family: arial ,
sans-serif}
p.foot {color: darkblue; font-size :9pt; text-align: center; font-family : arial
, sans-serif ; font-weight : bold}
a: link , a : visited, a : active {color : white }--> </style> </head>
```

```

<body <!-- page header -->
<table width="100%" cellpadding = 12 cellspacing =0 border = 0>
<tr bgcolor = silver>
    <td align = left><img src = "logo.png "></td>
<td> <h1>Talabalar portali</h1> </td>
    <td align = right><img src = "logo.png "></td>
</tr> </table>
<!-- menu -->
<table width = "100%" bgcolor = white cellpadding = 4 cellspacing =
4>
<tr > <td width = "25%">
<img src = "s-logo.png "> <span class=menu>Asosiy</span></td>
<td width = "25%">
<img src = "s-logo.png "> <span class=menu>Ta'lim resurslari</span></
td>
<td width = "25%"><imgsrc="s-logo.png"> <span class=menu>Talabalar
uchun</span></td> </tr> </table>
<!-- page content -->
<p>Uzluksiz ta'lim kadrlar tayyorlash tizimining asosi, O'zbekiston Res-
publikasining ijtimoiy-iqtisodiy taraqqiyotini ta'minlovchi, shaxs, jamiyat
va davlatning iqtisodiy, ijtimoiy, ilmiy-texnikaviy va madaniy ehtiyojlarini
kondiruvchi ustuvor sohadir. </p>

```

```

<!-- page footer-->
<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr> <td>
<p class=foot>&copy; Designed By SMD.</p>
<p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@
mail.ru">fmahbuba@mail.ru</a>manziliga jo'nating </p>
</td> </tr>
</table> </body> </html>

```

5.1-listingdan ko'rinib turibdiki, bu faylda kodlar alohida satrlarga ajratilgan.

“page header” (“sahifaning yuqori kolontituli”) bilan e’lon qilingan qismda portalning nomi va logotipi, “menu”(“menyu”) qismida menyular paneli, “page content” (“sahifaning tarkibi”) qismida berilgan sahifa uchun matn va so’ngida esa sahifaning quyi kolontituli joylashadi. Bu faylni ajratish va uning qismlarini mos ravishda header.inc, home.php va footer.inc kabi nomlash kerak. header.inc va footer.inc fayllarida boshqa sahifalarda takrorlanuvchi kodlar ishlatiladi.

home.php fayli esa task_form.html fayli o'rniga almashtiriladi va u sahifaning unikal tarkibi va ikkita require() operatoridan tashkil topadi (5.2.1-listingda ko'rsatilgani kabi).

5.2.1-listing. home.php – talabalar portalining asosiy sahifasini yaratuvchi PHP kod.

```
<? require("header.inc"); ?>
<!-- page content -->
<p>Uzluksiz ta'lim kadrlar tayyorlash tizimining asosi, O'zbekiston Res-
publikasining ijtimoiy-iqtisodiy taraqqiyotini ta'minlovchi, shaxs, jamiyat
va davlatning iqtisodiy, ijtimoiy, ilmiy-texnikaviy va madaniy ehtiyojlarini
qondiruvchi ustuvor soxadir. </p>
<? require("footer.inc"); ?>
```

home.php faylida require() operatori header.inc va footer.inc fayllarini yuklaydi.

header.inc fayli ushbu sahifada ishlatiluvchi CSS – aniqllovchi (Cascading Style Sheet, CSS), portal nomi va menyular satri tasvir- lanadigan jadvaldan tashkil topgan.

5.2.1-listing. header.inc – talabalar portalining barcha Web sahi- falari uchun takrorlanib ishlatiluvchi yuqori kolontituli.

```
<html> <head> <title>Talabalar portali</title>
<style> <!--
h1 {color: white; font-size :24p; text-align : center ; font-family : arial ,
sans-serif}
.menu {color: white; font-size: 12pt; text-align : center ; font-family : arial
, sans-serif; font-weight : bold}
td {background: silver}
p {color :darkblue; font-size: 12pt; text-align: justify;font-family: arial ,
sans-serif}
p.foot {color: darkblue; font-size :9pt; text-align: center; font-family : arial
, sans-serif ; font-weight : bold}
p {color :darkblue; font-size: 12pt; text-align: justify;font-family: arial ,
sans-serif}
p.foot {color: darkblue; font-size :9pt; text-align: center; font-family : arial
, sans-serif ; font-weight : bold}
a: link , a : visited, a : active {color : white }
-->
</style>
```



```

</head> <body>
  <!-- page header -->
  <table width="100%" cellpadding = 12 cellspacing =0 border = 0>
  <tr bgcolor = silver>
    <td align = left><img src = "logo.png "></td>
    <td> <h1>Talabalar portali</h1> </td>
    <td align = right><img src = "logo.png "></td>
  </tr>
</table>
  <!-- menu -->
  <table width = "100%" bgcolor = white cellpadding = 4 cellspacing =
4>
    <tr>
      <td width = "25%">
        <img src = "s-logo.png "> <span class=menu>Asosiy</span></td>
      <td width = "25%">

```

```

        <img src = "s-logo.png "> <span class=menu>Ta'lim resurslari</span></
td>
      <td width = "25%">
        <img src = "s-logo.png "> <span class=menu>Talabalar uchun</span></
td>
    </tr>
  </table>

```

footer.inc fayli har bir sahifaning quyi qismida ishlatiluvchi quyi kolontitul tasvirlanadigan jadvaldan tashkil topgan.

5.2.1-listing. footer.inc – talabalar portalining barcha Web sahifalari uchun takrorlanib ishlatiluvchi quyi kolontituli.

```

  <table width = "100%" bgcolor = silver cellpadding = 12 border =0>
  <tr> <td>
    <p class=foot>&copy; Designed By SMD.</p>
    <p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@
mail.ru">fmahbuba@mail.ru</a>manziliga jo'nating </p>
  </td> </tr>
</table>
</body>
</html>

```


VI bob. FUNKSIYALAR

Bu bobda funksiya tushunchasi, funksiyalar, foydalanuvchi tomonidan aniqlangan funksiyalar, funksiya parametrlari, parametrlarni qiymati va jo'natmasi bo'yicha uzatish, qaytariladigan qiymatlar va jo'natmalar (return() funksiyalar) bilan tanishiladi.

Funksiya ko'pgina dasturlash tillarida ishlatiladi. Funksiya dasturda batafsil aniqlangan topshiriqni alohida bajarish uchun yozilgan buyruqlar to'plami (kodlar)ni belgilash uchun ishlatiladi. Funksiya har gal topshiriqni bajarish uchun buyruqlar to'plamini o'qish va qayta ishlatish imkoniyatini yaratadi. Funksiyalar ikki xil bo'ladi: standart funksiyalar va foydalanuvchi tomonidan aniqlangan funksiyalar.

PHP ning standart funksiyalari fayllar bilan ishlash, ma'lumotlar bazasini ishlatish, grafik tasvirlar yaratish va boshqa serverlarga bog'lanish uchun imkon beradi.

Ba'zi hollarda dastur tuzish vaqtida standart funksiyalar imkoniyatlaridan tashqari biror-bir narsani bajarishga to'g'ri keladi. Bu hollarda dasturchilarning imkoniyatlari chegaralanmagan, ya'ni standart funksiyalardan tashqari ular o'zlari funksiya yaratishlari mumkin va bu funksiyalar foydalanuvchi tomonidan aniqlangan funksiyalar deyiladi. Foydalanuvchi tomonidan aniqlangan funksiya o'zida nimani mujassamlashtirganini ko'rib o'tsak. Funksiya dasturlashda ham matematikaga o'xshab bir nechta parametrlarni ularning qiymatiga aks etishini ifodalaydi, ya'ni funksiya ishining natijasi bo'lgan har bir berilgan parametrqa qandaydir qiymat qaytaradi. Topshiriqni bajarish uchun yaratilgan kod biror-bir topshiriqni bajaruvchi maxsus, mantiqiy va mavjud funksiyalar va amallar kombinatsiyasidan iborat. Bu kod funksiya e'lon qilinganda berilib, sahifaning bir nechta qismida yoki bir nechta sahifalarida funksiya murojaat etilganda bajariladi.

6.1. Foydalanuvchi tomonidan aniqlangan funksiyalar

Foydalanuvchi tomonidan aniqlangan funksiyaning e'lon qilish yangi funksiyaning yaratilishi. Funksiya keyinchalik foydalanuvchi tomonidan aniqlangan funksiya ekanligi haqida foydalanuvchiga xabar berish va dasturda PHP sintaksisini tahlil qilish uchun function kalit so'zi yordamida e'lon qilinadi. Funksiya function kalit so'zi bilan

boshlanib, funksiya nomini, talab qilinayotgan parametrlar va funktsiyaga har gal murojaat etilganda bajariladigan kodlarni o'z ichiga oladi.

Umumiy holda foydalanuvchi tomonidan aniqlangan funksiyalarni e'lon qilish quyidagi ko'rinishda bo'ladi.

Sintaksisi:

```
function funksiya_nomi(parametr1, parametr2, ..., parametrN)
{
    harakatlar_maydoni
    return qaytariladigan_funksiya_qiymati;
}
```

Bu yerda

funksiya_nomi – funksiya nomi;

harakatlar_maydoni – funksiya tanasi, ya'ni biror-bir topshiriqni bajaruvchi kodlar;

parametr1, ..., parametrN – lar esa almashuvchi ma'lumotlar.

Funksiya quyidagi talablarga javob berishi kerak:

funksiya nomi va parametr nomlari PHP ning nomlash qoidalariga mos tushishi kerak (krillcha belgilarni qo'llamaslik maqsadga muvofiq). Funksiya nomlari registrga bog'liq emas;

funksiya parametrlari o'zgaruvchilar hisoblanadi. Shuning uchun ularning har biri oldidan \$ belgisi ishlatilishi kerak. Parametrlar ro'yxatida ...(uch nuqta) ishlatish mumkin emas;

harakatlar_maydoni so'zi o'rnida biror-bir vazifani bajaruvchi PHP kod turishi kerak (parametrlarga bog'liq bo'lishi shart emas).

PHP ning standart funksiyalari barcha sahifalarda ishlay oladi. Foydalanuvchi tomonidan aniqlangan funksiyalar esa faqatgina ular e'lon qilingan sahifalarda ishlaydi. Bundan kelib chiqadiki, tez-tez ishlatiladigan foydalanuvchi tomonidan aniqlangan funksiyalarni alohida faylda saqlash kerak. Va bu funksiyalarga qaysi sahifada murojaat etilsa, shu sahifani boshida require() yoki include() operatorlari yordamida funksiyalar joylashgan fayl yuklab olinadi.

Funksiyaning ichki kodi figurali qavs ichida berilib, talab qilingan topshiriqni bajaradi.

Agar funksiya ichida PHP dan chiqib, HTML matnga o'tish kerak bo'lsa, sahifaning ixtiyoriy joyida PHP deskriptori quyidagicha yopilib, ochiladi:


```
<? function my_funksiya() { ?>
```

Mening funksiyam e'lon qilindi

```
<? } ?>
```

6.2. Registr va funksiya nomi

Funksiya nomini berilishi o'zgaruvchi nomini berishdan farq qiladi. O'zgaruvchi nomi registrga sezgir, shuning uchun \$Nomi va \$nomi – turli xil o'zgaruvchilar, lekin Nomi() va nomi() - bitta funksiya. Funksiya nomi registrga ahamiyatsiz, shuning uchun funksiya_nomi(), Funksiya_Nomi() yoki FUNKSIYA_NOMI() murojaatlari bitta natijani beradi. Funksiya nomlarida harflarni ixtiyoriy (katta, kichik) holda ishlatish mumkin.

Funksiya nomi qisqa va tushunarli bo'lishi kerak. Agar funksiya sahifaning yuqori qismi uchun yaratilayotgan bo'lsa, u holda page_header() yoki page_header() kabi nomlanishi mumkin.

Funksiya nomiga quyidagicha chegaralar qo'yiladi:

- funksiya PHP ning standart funksiyalari nomi bilan nomlana olmaydi;
- funksiya nomi faqatgina harflardan, raqamlardan va chiziqli belgisidan iborat bo'la oladi;
- funksiya nomi raqam bilan boshlana olmaydi.

Ko'pgina dasturlash tillarida funksiya nomi qaytarilishiga ruxsat bor. Bu funksiyaning qaytarilishi(peregruzka) deb ataladi. Faqatgina PHP qaytarilgan funksiyaning qo'llamaydi, shuning uchun funksiya nomi PHP ning standart funksiyalari nomlari va avval foydalanuvchi tomonidan aniqlangan funksiya nomlari bilan bir xil bo'lmasligi kerak.

Avval aytib o'tganimizdek, PHP ning standart funksiyalari barcha PHP sahifalarda ishlashi mumkin, foydalanuvchi tomonidan aniqlangan funksiya esa faqatgina e'lon qilingan sahifalardagina ishlay oladi. Bu shuni anglatadiki, e'lon qilinadigan funksiya nomlari boshqa fayllarda takroran ishlatilishi mumkin, lekin bu chalg'ishga (adashishga) olib keladi. Shuning uchun bu holatni qo'llamagan ma'qul.

Funksiya nomlari quyidagicha bo'lishi mumkin:

Name();

Name2();

Name_tree();

_namefour;

Funksiya nomlari quyidagicha bo'lishi mumkin emas:

\$name()

Name_six;

Fopen()

(Fopen()) funksiya nomi PHP ning standart funksiyasi)

6.3. Funksiyaga murojaat etish

Funksiyaga qanday murojaat etiladi? Avval funksiya nomi beriladi va qavslar ichida uning parametrlari qiymatlari ro'yxati kiritiladi.

Sintaksisi:

funksiya_nomi();

bu kod biror-bir parametрни talab qilmaydigan funksiya_nomi nomli funksiyaga murojaat etadi. Funksiyaga bu tarzda murojaat etish funksiya qaytaradigan barcha qiymatlarni inkor etadi. Ko'pgina funksiyalar shu tarzda beriladi.

Misol uchun, PHP ning standart funksiyalaridan biri phpinfo() funksiyasi. Bu funksiya o'rnatilgan PHP versiyasi va u haqida ma'lumot olish, PHP va serverning turli o'zgaruvchilari qiymatlarini ko'rsatish uchun ishlatiladi. Bu funksiya hech qanday parametr qabul qilmaydi va uning qaytaradigan qiymati ham inkor etiladi. Shuning uchun phpinfo() ning berilishi quyidagicha:

phpinfo();

Foydalanuvchi tomonidan aniqlangan funksiyaga murojaat etish PHP ning standart funksiyalariga murojaat etish kabi bo'ladi.

6.1-misol. Oddiy funksiyani e'lon qilamiz:

```
<? function my_function() // my_function() funksiyasi e'lon qilindi  
{ Echo "Mening funksiyam e'lon qilindi"; }  
my_function(); // my_function() funksiyasiga murojaat etildi  
>
```

Natija:Mening funksiyam e'lon qilindi

Funksiyaga qachon murojaat etiladi?

PHP ning 4 versiyasigacha funksiyaga u aniqlangandan so'ng murojaat etiladi, ya'ni dasturlash qatorida

function funksiya_nomi() {...}

blokidan so'ng.

Lekin PHP 4 da esa bunday talab yo'q. Hammasi interpretator olingan kodni qanday qayta ishlashiga bog'liq. Faqat shartli aniqlanadigan funksiyalar (shartli operator va boshqa funksiyalar ishida) bundan xoli.

6.2-misol. Shartli o'tish operatori ichida funksiyaning ishlatilishi

```
<? $make = true;
/* bu yerda Make_event()ga murojaat etish mumkin emas, chunki u bu
yerda mavjud emas, lekin Save_info()ga murojaat etish mumkin */
Save_info("Vohid", "Fozilov", "Men Informatika fanini tanladim");
if ($make)
{ function Make_event() { // Make_event() funksiyasining aniqlanishi
    echo "<p> Tarix fanini o'rganishni xohlayman"; } }
// endi Make_event()ga murojaat etish mumkin
Make_event();
// Save_info funksiyasining aniqlanishi
function Save_info($first, $last, $message)
{ echo "<br>$message<br>";
    echo "Ismi-sharifi: ". $first . " ". $last . "<br>"; }
// Save_info ga bu yerda ham murojaat etish mumkin
Save_info("Laziz", "Abdullayev", "Men Falsafa fanini tanladim"); ?>
```

Natija:

Men Informatika fanini tanladim

Ismi-sharifi: Vohid Fozilov

Tarix fanini o'rganishni xohlayman

Men Falsafa fanini tanladim

Ismi-sharifi: Laziz Abdullayev

Agar funksiya dasturda bir marta aniqlangan bo'lsa, uni ikkinchi marotaba qayta aniqlash yoki o'chirib tashlash mumkin emas. Funksiyalar registriga bog'liq emasligiga qaramasdan ularga qanday nom bilan aniqlangan bo'lsa, o'sha nom bilan murojaat etish lozim.

6.4. Parametrlar (yoki Argumentlar)

Ko'pgina funksiyalarda o'z topshiriqlarini bajarish uchun bir yoki undan ortiq parametrlarning uzatilishi talab etiladi. Parametrlar ma'lumotlarni, qiymatlarni funksiyaga yetkazish imkonini beradi.

Har bir parametr o'zida o'zgaruvchini va konstantani mujassamlashtiradi. Ular () qavs ichida funksiya nomidan so'ng beriladi.

funksiya_nomi("parametr1 uchun qiymat", "parametr2 uchun qiymat",...);

Bu yerda "parametr1 uchun qiymat", "parametr2 uchun qiymat",... lar o'rnida parametrlarning qiymatlari beriladi. Masalan,

funksiya_nomi(2);

funksiya_nomi(7.993);

funksiya_nomi(\$variable);

Oxirgi berilgan funksiyada \$variable o'zgaruvchisi ixtiyoriy tipda, hatto massiv ham bo'lishi mumkin.

Parametr qiymatlari yoki qiymatlarni o'zida jamlagan o'zgaruvchi nomlari turli yo'l bilan uzatiladi. Funksiyaga ma'lumotlarni funksiya parametrlari yordamida uch xil yo'l bilan uzatish mumkin:

1) qiymati bo'yicha (odatga ko'ra ishlatiladi);

2) jo'natmasi (ssilka) bo'yicha;

3) ko'rinmasligi bo'yicha.

Funksiya parametrlarini oddiy yo'l bilan uzatish qiymati bo'yicha uzatish deb ataladi. Parametrlarni uzatishda uzatilayotgan qiymatni o'zida jamlagan yangi o'zgaruvchi yaratiladi. U mavjud (ichki) o'zgaruvchini nusxasi. Bu qiymat ixtiyoriy tarzda o'zgarishi mumkin, lekin mavjud (ichki) o'zgaruvchining qiymati funksiya tashqarisida o'zgarmay qoladi. Parametr funksiya qiymati bo'yicha uzatilayotganda parametrning funksiya ichida o'zgarishi uning funksiya tashqarisidagi qiymatiga ta'sir qilmaydi.

Qiymatni ketma-ket oshirib boruvchi funktsiyani yaratamiz.

6.3-misol. Parametrlarni qiymati bo'yicha uzatish

```
<? Function increment($value, $a=1)
```

```
{ $value=$value+$a; }
```

```
$value=15;
```

```
increment($value);
```

```
Echo $value; ?>
```

```
Natija:15
```

Natija 15 chiqadi, ya'ni \$value o'zgaruvchining qiymati o'zgarmaydi.

Demak, bu kod biz talab etgan vazifani bajarmadi. Bu esa harakatlar_maydoni bilan bog'liq. Kod vazifasiga ko'ra \$value o'zgaruvchisini

yaratadi, so'ng dastur increment() funksiyasiga murojaat etadi. Funksiya ichida \$value o'zgaruvchisi unga murojaat etishiga ko'ra yaratiladi, so'ng uning qiymatiga 1 qo'shiladi va funksiya ichida \$value qiymati 16 ga teng bo'ladi, lekin natijani ekranga chiqarishga buyruq berilmaganligi va funksiya tashqarisida \$value o'zgaruvchisi global maydonda aniqlangan boshqa o'zgaruvchi bo'lganligi uchun o'zgarmay qoladi, shuning uchun natija 15 ga teng bo'ladi.

Bu muammoni hal qilish yo'llaridan biri bu — \$value o'zgaruvchisini funksiyada global sifatida e'lon qilish kerak, lekin bu shuni anglatadiki, bu funktsiyani ishlatishda uning qiymati oshirilishi kerak bo'lgan o'zgaruvchisini \$value deb nomlash talab etiladi. Bundan ham maqbul yo'l sifatida jo'natma bo'yicha uzatishni aytishimiz mumkin. Funksiya parametrlarini o'zgartirishga ruxsat berish uchun ularni jo'natma (ssilka) bo'yicha uzatish kerak.

Jo'natma bo'yicha parametrlar uzatilayotganda yangi qiymat yaratish bilan birga funksiya jo'natmadan ichki o'zgaruvchilarni qabul qiladi. Bu jo'natma \$ dollar belgisi bilan boshlanuvchi o'zgaruvchi nomi yoki biror-bir ixtiyoriy o'zgaruvchi. Farqi shundaki, parametr olayotgan qiymatni ichki o'zgaruvchiga uzatadi.

Parametrlarni jo'natma yordamida uzatishda funksiya e'lon qilinayotgan vaqtda parametr oldidan (&) ampersand belgisi qo'yiladi. Funksiyaga murojaat etishda esa hech qanday o'zgartirish kiritish shart emas. 6.2-misol kamchiligini to'g'rilaymiz.

6.4-misol. Parametrlarni jo'natma bo'yicha uzatish.

```
<? Function increment (&$value, $a=1)
```

```
{ $value=$value+$a; }
```

```
$b=15;
```

```
Echo $b.<br>;
```

```
increment($b);
```

```
Echo $b; ?>
```

```
Natija:
```

```
15
```

```
16
```

Funksiya oddiy yo'l bilan uzatilgan bo'lsa, unda parametrlar qiymatini aniqlash mumkin. Oddiy yo'l bo'yicha uzatilgan qiymat o'zgaruvchi va klass vakili yoki boshqa funksiyadan chaqirilgan bo'lmasdan konstanta bo'lishi kerak.

Parametrlarni ko‘rinmasligi bo‘yicha uzatish.

Bizda berilgan parametrga bog‘liq holda o‘zgaradigan informat-sion xabar yaratuvchi funksiya mavjud. Agar funksiya parametri qiymati berilmagan bo‘lsa, u holda qiymat o‘rnida “Dekanat” yozuvi qo‘llaniladi.

6.5-misol. Parametrlarni ko‘rinmasligi bo‘yicha uzatish

```
<?php
function xabar($sign="Dekanat"){
// bu yerda sign "Dekanat" mazmuniga ega
echo "Keyingi majlis ertaga bo'ladi.";
echo "$sign .<br>"; }
xabar();//
funksiyani paramsiz chaqiramiz. Bu holda yozuv –Dekanat
xabar("Hurmat bilan Afzalov"); // natija "Hurmat bilan Afzalov" bo'ladi
?>
```

Natija:
Keyingi majlis ertaga bo'ladi.Dekanat .
Keyingi majlis ertaga bo'ladi.Hurmat bilan Afzalov .

Bir nechta parametrlar bilan berilgan funksiyaning oddiy yo‘l bilan e‘lon qilinishi, qiymati bilan berilgan parametrlar hammasi parametrlardan keyin yozilishi kerak. Aks holda, funksiya murojaat etilayotganda bu parametrlar tashlab yuboriladi va natija xato chiqadi.

6.6-misol. Talaba tomonidan biror-bir maqola katalogga qo‘shilishi kerak. Maqolani katalogga qo‘shish uchun sahifada maqolaning nomi, muallifi va qisqa izohi kabi tavsiflari kiritiladi. Agar talaba maqolaning muallifini kiritmagan bo‘lsa, u holda maqola muallifini Komilov Islom deb dasturning o‘zi kiritsin.

```
<? function Add_maqola($nomi, $izohi, $muallifi="Komilov Islom"){
echo "Katalogga $nomi maqolasini qo'shamiz. ";
echo "<br>Muallif: $muallifi";
echo "<br>Qisqa izoh: ";
echo "$izohi<br>"; }
Add_maqola("Informatika asoslari", "Bu maqola informatika bo'yicha ...", "Yuldashev Umid");
Add_maqola("Ma'lumotlar bazasi", "Bu maqola ma'lumotlar bazasi haqida ..."); ?>
```


Natija: Katalogga Informatika asoslari maqolasini qo'shamiz.

Muallif: Yuldashev Umid

Qisqa izoh: Bu maqola informatika bo'yicha ...

Katalogga Ma'lumotlar bazasi maqolasini qo'shamiz.

Muallif: Komilov Islom

Qisqa izoh: Bu maqola ma'lumotlar bazasi haqida ...

Agar biz quyidagicha yozsak, xatolik yuzaga keladi, chunki qiymatga ega parametr oxirida emas, balki parametrlar ro'yxatining boshida yozilgan:

```
<? function Add_maqola($muallifi="Komilov Islom", $nomi, $izohi)
{ echo "Katalogga $nomi maqolasini qo'shamiz. ";
  echo "<br>Muallif: $muallifi";
  echo "<br>Qisqa izoh: ";
  echo "$izohi<br>"; }
Add_maqola("Ma'lumotlar bazasi", "Bu maqola ma'lumotlar bazasi ha-
qida ..."); ?>
```

O'zgaruvchi parametrlar ro'yxati

PHP4 da funktsiyaga nechta parametr bo'yicha murojaat etishni oldindan bilmasdan turib funktsiyani yaratish mumkin. Bunday funktsiyani yozish uchun hech qanday maxsus sintaksislar talab etilmaydi. Hammasi PHP ning standart funktsiyalari yordamida bajariladi `func_num_args()`, `func_get_arg()`, `func_get_args()`.

`func_num_args()` funktsiyasi foydalanuvchi tomonidan aniqlangan funktsiyada beriladigan parametrlar sonini qaytaradi. Bu funktsiya faqat foydalanuvchi tomonidan aniqlangan funktsiya ichida qo'llanilishi mumkin. Agar `func_num_args()` funktsiyasi foydalanuvchi tomonidan aniqlangan funktsiyaning tashqarisida qo'llanilsa, unda interpreter bu haqida xabar beradi.

6.7-misol. `func_num_args()` funktsiyasining ishlatilishi

```
<? function DataCheck(){
  $n = func_num_args();
  echo "Funktsiya parametrlari soni $n ta<br>"; }
DataCheck(); // "Funktsiya parametrlari soni 0 ta " qatorini chiqaradi
DataCheck(1,2,3,6);
// "Funktsiya parametrlari soni 3 ta " qatorini chiqaradi
?>
```

`func_get_arg(butun parametr_nomeri)` funktsiyasi foydalanuvchi tomonidan aniqlangan funktsiya parametrlari ro'yxatidan raqamlar ketma-ketligi bo'yicha parametr_nomeriga mos berilgan parametрни

qaytaradi. Foydalanuvchi tomonidan aniqlangan funksiya parametri noldan boshlab sanaladi. `func_num_args()` dagidek bu funksiya ham faqat foydalanuvchi tomonidan aniqlangan funksiya ichida qo'llanilishi mumkin.

`parametr_nomeri` foydalanuvchi tomonidan aniqlangan funksiya da berilgan parametrlar sonidan katta bo'lishi mumkin emas. Aks holda `func_get_arg()` funksiyasi false ni qaytaradi va mos xabarni ishlab chiqadi.

6.8-misol. Parametrlarda berilgan ma'lumotlar tipini tekshirish uchun funksiya yarataylik.

```
<? Function DataCheck(){ $check =true;
$n = func_num_args(); // funksiya ga berilgan parametrlar soni
/* birinchi berilgan parametr butun son ekanligini tekshiramiz */
if ($n>=1) if (!is_int(func_get_arg(0))) $check = false;
/* ikkinchi berilgan parametr satr ekanligini tekshiramiz*/
if ($n>=2) if (!is_string(func_get_arg(1))) $check = false;
return $check; }
if (DataCheck(123,"text"))
echo "Berilganlar shartlarni qoniqtiradi<br>";
else echo "Berilganlar shartlarni qoniqtirmaydi<br>";
if (DataCheck(324)) echo " Berilganlar shartlarni qoniqtiradi<br>";
else echo "Berilganlar shartlarni qoniqtirmaydi"; ?>
Natija:
Berilganlar shartlarni qoniqtirmaydi
Berilganlar shartlarni qoniqtiradi
```

`func_get_args()` funksiyasi foydalanuvchi tomonidan aniqlangan funksiya da berilgan parametrlar ro'yxatidan tuzilgan massivni qaytaradi. Massivning har bir elementi funksiya da berilgan parametrga mos tushadi. Agar funksiya foydalanuvchi tomonidan aniqlangan funksiya da qo'llanilmayotgan bo'lsa, u holda xabar ishlab chiqiladi.

6.9-misolga ushbu funksiyaning qo'llab o'zgartirish kiritamiz. Funksiya ga berilayotgan har bir juft parametr butun son ekanligini tekshiramiz.

```
<? Function DataCheck(){
$check =true;
$n = func_num_args(); // funksiya ga berilgan parametrlar soni
$args = func_get_args(); // funksiya parametri massivi
for ($i=1;$i<=$n;$i++){ $v = $args[$i];
if ($i % 2 == 0){ if (!is_int($v)) $check = false;
```



```
// juft parametr butun sonligini tekshiramiz } }  
return $check; }  
if (DataCheck("text", 324)) echo "Berilgan juft parametr juft son<br>";  
else echo "Berilganlar shartlarni qoniqtirmaydi"; ?>
```

Ko'rib turganimizdek, `func_num_args()`, `func_get_arg()`, `func_get_args()` funksiyalar kombinatsiyasi foydalanuvchi tomonidan aniqlangan funksiyalar o'zgaruvchi parametrlar ro'yxatiga ega bo'lishi uchun qo'llaniladi. Bu funksiyalar faqat PHP 4 da berilgan. PHP 3 da shunga o'xshash natijaga erishish uchun massiv funksiyasi qo'llaniladi.

6.5. Harakatlar maydoni

Oldingi mavzularda o'zgaruvchilarni biror-bir fayl ichida e'lon qilib, sahifaga `require()` yoki `include()` operatorlari yordamida yuklab olish haqida gapirib o'tgan edik, funksiyada esa o'zgaruvchilarni funksiyani o'zida e'lon qilinadi.

O'zgaruvchilarning harakat diapazoni o'zgaruvchilarni ko'rinarli va belgilanganligiga qarab boshqaradi. Turli dasturlash tillarida o'zgaruvchilarni harakat diapazoni o'rnatilgan turlicha qoidalarga bo'ysunadi.

PHP da esa quyidagicha qoida mavjud:

- funksiya ichida aniqlangan o'zgaruvchilar ular e'lon qilingan operatoridan boshlab funksiyani yakunlovchi qavsgacha bo'lgan maydonda harakatlanadi. Bu maydon funksiya maydoni, o'zgaruvchilar esa – lokal o'zgaruvchilar deb ataladi;

- funksiya tashqarisida aniqlangan o'zgaruvchilar ular e'lon qilingan operatoridan boshlab, funksiya ichida emas, balki fayl oxirigacha bo'lgan maydonda harakatlanadi. Bu maydon global maydon, o'zgaruvchilar esa – global o'zgaruvchilar deyiladi;

- `require()` va `include()` operatorlari yordamida yuklab olingan fayllarda e'lon qilingan o'zgaruvchilar funksiya harakat maydonida qabul qilinmaydi. Agar operatorlar funksiya ichida ishlatilsa, funksiya maydonida qabul qilinadi. Agar operatorlar funksiya tashqarisida ishlatilsa, global maydonda qabul qilinadi;

- funksiya ichida aniqlangan yoki ishlatilayotgan o'zgaruvchilarni global maydonda harakatlana olishi uchun global kalit so'zi yordamida belgilash (ko'rsatish) mumkin, ya'ni o'zgaruvchilarni funk-

siya ichida qo'llash uchun bu o'zgaruvchilar global o'zgaruvchi sifatida e'lon qilinishi kerak. Buning uchun funksiya tanasida global kalit so'zidan keyin ularning nomlari keltirilishi kerak;

global \$var1, \$var2, \$var3, ...,

— o'zgaruvchilar unset(\$variable_name) funksiyasi yordamida o'chiriladi. Agar o'zgaruvchi o'chirilgan bo'lsa, harakat maydonida boshqa ishlatilmaydi.

6.6. Funksiya ichida o'zgaruvchilarning ishlatilishi

fn() funksiyasi ichida \$var o'zgaruvchisi e'lon qilindi. Bu o'zgaruvchi funksiya ichida e'lon qilingani uchun e'lon qilingan joydan boshlab funksiya yakunigacha funksiya harakatlanish maydonida ishlay oladi. Funksiyadan tashqarida esa \$var o'zgaruvchisiga murojaat etildi, ya'ni yangi \$var o'zgaruvchisi yaratildi. Bu yangi o'zgaruvchi global harakatlanish maydonida fayl oxirigacha ishlay oladi. Agar bu o'zgaruvchini echo operatori yordamida ekranga chiqazsak, hech qanday qiymat chiqmaydi. Chunki bu o'zgaruvchi funksiya tashqarisida qiymatga ega emas.

```
<? Function fn()  
{ $var="mazmun"; }  
Echo $var; ?>
```

Quyidagi misol bunga teskari. O'zgaruvchini funksiya tashqarisida e'lon qilamiz, so'ngra funksiya ichida uni ishlatamiz.

6.10-misol. O'zgaruvchilarning funksiya ichida va tashqarisida ishlatilishi

```
<? Function fn() { Echo "funksiya ichida, \ $var=". $var."<br>";  
$var="mazmun2"; Echo "funksiya ichida, \ $var=". $var."<br>"; }  
$var="mazmun1";  
Fn(); Echo "funksiya tashqarisida, \ $var=". $var."<br>"; ?>  
Natija: funksiya ichida, $var=  
funksiya ichida, $var=mazmun2  
funksiya tashqarisida, $var=mazmun1
```

funksiya bu holda ham bajarilmadi, chunki o'zgaruvchi funksiya ichiga chaqirilmadi, birinchi bajarilishi lozim bo'lgan operator \$var="mazmun1";

bu yerda global maydonda harakatlana oladigan va qiymati mazmun1 ga teng \$var o'zgaruvchisi yaratiladi.

Keyingi bajarilayotgan operator — funksiyaga murojaat etish. Funksiya ichidagi satrlar ketma-ket bajariladi. Funksiyani birinchi satri \$var o'zgaruvchisiga murojaat etadi. Bu satr bajarilganda yaratilgan \$var o'zgaruvchisini ko'ra olmaydi, shuning uchun funksiya maydonida yangi o'zgaruvchi yaratadi va uni ekranga chiqazadi. Natijada 1 satr ekranga chiqadi.

Funksiya ichidagi keyingi satr \$var o'zgaruvchisini "mazmun2" ga tenglashtiradi. Harakat funksiya ichida bo'layotganligi uchun bu satr global emas, balki lokal o'zgaruvchi \$var qiymatini o'zgartiradi. Ikkinchi satr esa bu o'zgarishni ekranga chiqazadi.

Shundan so'ng funksiya ishini tugatadi va sahifani oxirgi satri bajariladi. Echo operatori yordamida global o'zgaruvchining o'zgarmaganligini ko'rish mumkin.

Global o'zgaruvchilar

Agar funksiya ichida yaratilgan o'zgaruvchi global bo'lishi kerak bo'lsa, uning oldidan global kalit so'zi ishlatiladi.

6.11-misol. Global o'zgaruvchilarni ishlatilishi

```
<? Function fn() { global $var; $var="contents";  
Echo "funksiya ichida, \ $var=". $var."<br>"; }  
Fn(); Echo "funksiya tashqarisida, \ $var=". $var."<br>"; ?>  
Natija: funksiya ichida, $var=contents  
funksiya tashqarisida, $var=contents
```

Bu misolda, \$var global o'zgaruvchi sifatida e'lon qilingan bo'lib, o'zgaruvchi maydonda global \$var satri bajarilgan joydan boshlab aniqlangan. Shuning uchun o'zgaruvchi funksiya tashqarisida ham ishlatilishi mumkin.

Funksiyani unga murojaat etilayotgan joydan yuqorida yoki pastda ham e'lon qilish mumkin. (E'tibor bering, funksiya harakatlanish maydoni o'zgaruvchi harakatlanish maydonidan tubdan farq qiladi!)

Statik o'zgaruvchilar

O'zgaruvchilarni funksiya ichida, funksiyadan chiqqandan keyin ham ular qiymatini saqlab qolinadigan qilib qo'llash uchun ularni statik o'zgaruvchilar sifatida e'lon qilish kerak. Agar dastur bajarilishi funksiya qiymati chegaralaridan chiqsa, u holda statik o'zgaruvchilar faqat funksiya ichida ko'rinadi. Bunday o'zgaruvchilarni e'lon qilish uchun static kalit so'zi qo'llaniladi:

```
static $var1, $var2;
```

Statik o'zgaruvchiga istalgan qiymat biriktirilishi mumkin, lekin jo'natmani unga biriktirib bo'lmaydi.

6.12-misol. Statik o'zgaruvchilarni ishlatilishi

```
<? function Test_s(){  
    static $a = 1; // Ifoda va jo'natmani biriktirish mumkin emas  
    $a = $a*2; echo $a; }  
Test_s(); // 2 ni chiqaradi  
echo $a;  
/* $a faqatgina funksiya ichida qiymatga ega bo'lganligi uchun hech nar-  
sani chiqarmaydi Test_s(); funksiya ichida $a=2, shuning uchun funksiya ishi  
natijasi 4 soni bo'ladi */ ?>  
Natija:24
```

6.7. Qaytariluvchi mazmunlar

Yuqorida misol tariqasida keltirilgan funksiyalarning hammasi qandaydir harakatlarni bajaradi va natijani ekranga chiqazadi. Ba'zi hollarda istalgan funksiya ishining natijasi sifatida qandaydir qiymatni ekranga chiqazish emas, balki faqatgina natijani qaytarish kerak bo'ladi. Bu return buyrug'i yordamida amalga oshiriladi. Qaytariladigan qiymat har xil toifada bo'lishi mumkin, shu qatorda ro'yxat ham, obyekt ham. Interpretator funksiya tanasida return buyrug'ini ko'rgach, uni bajarilishini darrov to'xtatadi va qaysi qatorda funksiya murojaat etilgan bo'lsa o'sha qatorga o'tadi.

6.13-misol. Qaytariluvchi mazmunlar

```
<? function kvadrat($a) { $nat=$a*$a; return $nat; }  
$y=kvadrat($a); $c=$y/10+25;  
echo $c; ?>  
Natija:25
```

6.8. Qaytariluvchi jo'natmalar

O'z ishining natijasi sifatida funksiya qandaydir o'zgaruvchi(global, statik yoki parametrlardan biriga)ga jo'natmani ham qaytarishi mumkin. Qaytariluvchi jo'natmalar funksiyaning qaysi o'zgaruvchiga jo'natma biriktirilganini aniqlash kerak bo'lgan hollarda juda ham qo'l keladi. Funksiyadan jo'natmani olish uchun uni e'lon qilinishida va har safar funksiya murojaat etilayotganda nomi oldidan ampersand belgisi (&) ni qo'yish kerak. Odatda funksiya jo'natma bilan berilgan bo'lsa, jo'natmani qandaydir global o'zgaruvchiga

(yoki uning qismi – jo‘natmani global massiv elementiga), statik o‘zgaruvchiga (yoki uning qismiga) yoki parametrlardan biriga qaytaradi.

6.14-misol. Qaytariluvchi jo‘natmalar

```
<? $a = 'juft son'; $b = 'toq son';  
function &ref($par){ global $a, $b;  
    if ($par % 2 == 0) return $a; else return $b; }  
$c=4; var =&ref($c);  
echo $c. " - ". $var. "<br>"; // 4 - juft son chiqaradi  
$c = 11; $var =&ref($c);  
echo $c. " - ". $var. "<br>"; // 11 - toq son chiqaradi  
?>
```

Natija:

4 - juft son

11 - toq son

6.9. O‘zgaruvchili funksiyalar

PHP o‘zgaruvchili funksiyalar konsepsiyasini qo‘llaydi. Bu, agar o‘zgaruvchi qavslar bilan tugagan bo‘lsa, u holda PHP xuddi shunday nomli funksiyani qidirishini va uni bajarishga harakat qilishini bildiradi.

6.15-misol. O‘zgaruvchili funksiyalarning ishlatilishi

```
<? /* ikki oddiy funksiyani yaratamiz: Add_sign – qatorga yozuv qo‘shadi  
va Show_matn – matn qatorini chiqaradi */
```

```
function Add_sign($string, $sign="Hurmat bilan Komil")
```

```
{ echo $string." ".$sign; }
```

```
function Show_matn() { echo "Xabarni pochta orqali jo‘natish<br>"; }
```

```
$func = "Show_matn";
```

```
// Show_matn funksiyasi nomiga teng qiymatli o‘zgaruvchini yaratamiz
```

```
$func(); // Bu Show_matn funksiyasiga murojaat etadi
```

```
$func = "Add_sign";
```

```
// Add_sign funksiyasi nomiga teng qiymatli o‘zgaruvchini yaratamiz
```

```
$func("Hammaga salom <br>");
```

```
//“Hammaga salom” parametrlil Add_sign funksiyasiga murojaat etadi
```

```
?>
```

Natija:

Xabarni pochta orqali jo‘natish

Hammaga salom
Hurmat bilan Komil

Bu misolda Show_matn funksiyasi faqat satrni chiqarish uchun xizmat qiladi. Maxsus echo() funksiyasi mavjudligi tufayli nima uchun alohida funksiya yaratdik, degan savol tug'iladi. Gap shundaki echo(), print(), unset(), include() kabi funksiyalarni o'zgaruvchi sifatida qo'llab bo'lmaydi. Shuning uchun biz echo "TEXT"; ni quyidagidek yozsak, u holda interpretator quyidagi xatoni chiqazadi

```
<? $func = "echo";  
$func("TEXT"); ?>
```

Natija:

Fatal error: Call to undefined function echo() in D:\xampplite\htdocs\learn_php\learn.php on line 3

Shuning uchun yuqoridagi sanab o'tilgan funksiyalarni o'zgaruvchi sifatida qo'llash uchun yuqoridagi misoldagidek yangi funksiya yaratib olish kerak.

6.10. Standart funksiyalar

PHP — Web sahifalarni yaratishda ishlatiladigan kengaytirilgan funksiyalar kutubxonasiga ega texnologiya. PHP ning 100 dan ortiq standart funksiyalari mavjud. Bu funksiyalarning ko'pchiligi qandaydir ma'lumotlarni qabul qilib va natija sifatida biror-bir ma'lumot chiqazadi.

Funksiyani e'lon qilishni oddiy formasi sifatida PHP ning standart funksiyalaridan biri date() funksiyasini ko'ramiz.

Sintaksisi:

date(parametr);

E'tibor qarating, funksiyaga berilayotgan ma'lumot qavs ichiga olib yozilmoqda. Bu qiymat funksiya argumenti yoki parametri deb ataladi. Parametrlar kiruvchi qiymatlar hisoblanib, funksiya ekranga parametrlarga bog'liq aniq bir natijani chiqazadi.

Bu yerda parametr o'rnida quyidagilar ko'rsatilishi mumkin:

- a — "am" yoki "pm"
- A — "AM" yoki "PM"
- d — oy kuni, raqamli, 2 ta raqamdan (birinchi o'rinda nol turadi)
- D — hafta kuni, matnli, 3 harfdan iborat; "Fri" va h.

- F — oy, matnli, to'liq; "January" va h.
- h — soat, raqamli, 12 soatli formatda
- H — soat, raqamli, 24 soatli formatda
- i — minut, raqamli
- j — oy kuni, raqamli, boshida keluvchi nolsiz
- l ('L' katta harfli) — hafta kuni, matnli, to'liq; "Friday" va h.
- m — oy, raqamli
- M — oy, matnli, 3 harfdan iborat; "Jan" va h.
- s — sekund, raqamli
- S — inglizcha tartibdagi suffiks, matnli, 2 belgidan iborat; "th", "nd" va h.
- U — sekundda, asr boshidan boshlab
- Y — yil, raqamli, 4 ta raqamdan iborat
- w — hafta kuni, raqamli, 0 yakshanbani bildiradi
- y — yil, raqamli, 2 ta raqamdan iborat
- z — yil kuni, raqamli; "299" va h.

6.16-misol. date() funksiyasining ishlatilishi

```
<? $d=date(d);
$m=date(m);
$y=date(y);
Echo "Bugun: $d.$m.$y"; ?>
Natija:Bugun: 27.01.11
```

Isset() funksiyasi o'zgaruvchi mavjud yoki mavjud emasligini tekshiradi, agar o'zgaruvchi mavjud bo'lsa, funksiya true(1) qiymatini, aks holda false(0) ni qaytaradi.

Sintaksisi:

isset (o'zgaruvchi);

6.17-misol. Isset() funksiyasining ishlatilishi

```
<? $c=34;
If(isset($c))
{ Echo "O'zgaruvchi mavjud va uning qiymati $c ga teng"; }
Else { Echo "O'zgaruvchi mavjud emas"; } ?>
Natija:O'zgaruvchi mavjud va uning qiymati 34 ga teng
```

Unset() funksiyasi o'zgaruvchilarni o'chirish uchun ishlatiladi.

Sintaksisi:

unset (o'zgaruvchi);

6.18-misol. Unset() funksiyasining ishlatilishi

```
<? $k=34;  
Echo "funksiyaga murojaat etishdan avval k=". $k."<br>";  
unset($k);  
Echo "funksiyaga murojaat etgandan so'ng k=". $k."<br>"; ?>  
Natija:  
funksiyaga murojaat etishdan avval k=34  
funksiyaga murojaat etgandan so'ng k=
```

mail() funksiyasi xabarni elektron pochta orqali berilgan manzilga jo'natadi.

Sintaksisi:

boolean mail (string oluvchi, string mavzu, string xabar [, string qo'shimcha_sarlavha])

"mavzu" parametrida xabar mavzusi beriladi. "xabar" parametri esa xabar matnidani tarkib topadi, shart bo'lmagan parametr - "qo'shimcha_sarlavha" da xabar bilan jo'natiladigan qo'shimcha ma'lumot belgilanadi (masalan, HTML formatlash atributlari). Misol uchun,

```
<? $email = "yourname@yourserver.com";  
$mavzu = "Bu mavzu";  
$xabar = "Bu xabar";  
$sarlavha = "From: somebody@somesite.com";  
mail ($email, $mavzu, $xabar, $sarlavha); ?>
```

Biz PHP ning standart funksiyalarini ba'zilar bilan, ya'ni echo(), print(), include(), require(), date(), isset(), unset(), mail() kabilar bilan tanishdik. Aslida yuqorida sanab o'tilgan funksiyalar ning ichidan date() dan tashqari hammasi til konstruksiyalari hisoblanadi. Ular PHPning yadrosi hisoblanadi va hech qanday sozlash va modullarni talab etmaydi. Lekin shunday funksiyalar borki, ular bilan ishlash uchun har xil kutubxonalarni o'rnatish va mos modulni ulash kerak. Masalan, ma'lumotlar bazasi bilan ishlovchi MySQL funksiyasi bilan ishlash uchun PHP ni shu kengaytmada komplekslash kerak. Oxirgi vaqtlarda tarqalgan kengaytmalar va mos ravishda ularning funksiyalari ular bilan ishlashda interpretatorning hech qanday qo'shimcha sozlashlarisiz ishlashda PHPning boshlang'ich tarkibiga kiritilgan.

Shunga o'xshash PHP ning massivlar bilan, fayllar bilan, ma'lumotlar bazasi bilan ishlovchi standart funksiyalari mavjud. Biz bu funksiyalar bilan keyingi bo'limlarda tanishamiz.

6.1-masala.

Html formasini yaratish imkonini beruvchi interfeys yozamiz. Bu interfeys foydalanuvchi ko'rsatgan elementlar va ularni nechta sonda yaratishni, ularga nom berishni tanlaganda kerakli formani o'zi qayta ishlashi kerak.

Masalani bir nechta qismga bo'lamiz: 1) yaratiladigan elementlarni tanlash va ularning sonini kiritish; 2) yaratilgan elementlar nomini kiritish va olingan ma'lumotlarni qayta ishlash hamda 3) formani uzluksiz qayta ishlash. Birinchi masala oddiy: ya'ni yaratiladigan elementlarni tanlash va ularning sonini kiritish uchun formani yozamiz.

6.19-listing. task_form.html

```
<form action="ask_names.php">
  "Matn kiritish qatori" elementini yaratish: <input type=checkbox
name=types[] value=string><br>
  Elementlar soni: <input type=text name=numbers[string] size=3><br>
  <br>
  "Matn sohasi" elementini yaratish: <input type=checkbox name=types[]
value=text><br>
  Elementlar soni: <input type=text name=numbers[text] size=3><br>
  <input type=submit value="Yaratish">
</form>
```

Biz element formasi nomiga, masalan, types[] deb yozadigan bo'lsak, bu uning nomi – massivning keyingi elementi types ekanligini bildiradi. Ya'ni bizda birinchi element nomi types[0] (Matn kiritish satri), ikkinchisi types[1] (Matn sohasi) bo'ladi.

Bunday forma ma'lumoti jo'natilgandan keyin, qanday elementlar va ularning har biridan nechta yaratish kerakligi haqida ma'lumotga ega bo'lish mumkin. Keyingi skriptda bu elementlar uchun nom beriladi:

6.20-listing. ask_names.php

```
<? $file = "task.php";
/* shu script bilan formani qayta ishlovchi fayl*/
function Ask_names(){
global $file; //bu o'zgaruvchini berilgan funksiya tashqarisida
//qo'llashimiz uchun global sifatida e'lon qilamiz
```

```

if (isset($_GET["types"])){
    $st = '<form action="'. $file.' ">';
    foreach ($_GET["types"] as $k => $type){
        /* yaratilishi kerak bo'lgan hamma elementlar tipini olamiz */
        $num = $_GET["numbers"][$type];
        // har bir tipdan nechta element kerak
        for ($i=1;$i<=$num;$i++){
            $st.= "$type tipidagi $i-element nomini kiriting: ";

            $st.= "<input type=text name=names[$type][]><br>"; }
            $st.= "<input type=hidden name=types[] value=$type>";
            $st.= "<input type=hidden name=numbers[] value=$num><br>"; }
            $st.= "<input type=submit name=send value=Jo'natish ></form>";
            return $st;
            // $st o'zgaruvchisida so'rov nomi uchun forma kodi saqlanadi
        } else echo "Select type"; }
        echo Ask_names();
        // funktsiyani chaqiramiz va ishning natijasini chiqaramiz
    ?>

```

6.19-listingdagi task.form.html formada foydalanuvchi matn kiritish qatori tipiga 2 ta element va matn sohasi tipiga 1 ta element yaratish kerakligini ko'rsatgan bo'lsin. U holda ask_names.php skripti uni qayta ishlagach, biz quyidagiga ega bo'lamiz (6.1-rasm):

string tipidagi 1-element nomini kiriting: Nomi

string tipidagi 2-element nomini kiriting: Muallifi

text tipidagi 1-element nomini kiriting: Qisqacha mazmuni

Jo'natish

6.1-rasm. Yaratilgan elementlar nomini kiritish uchun forma

Bu formada qatorlarga "Nomi","Muallifi" va "Qisqacha mazmuni" kabi nomlarni kiritaylik. Bu ma'lumotlarni task.php skripti qayta ishlaydi.

6.21-listing. task.php

```
<?
$show_file = "task_show.php";
/* Ushbu fayl formasi yaratgan ma'lumotlarni qayta ishlovchi fayl */
function Create_element($type,$name){
    // funksiya kiritilga tipi va nomiga ko'ra element yaratadi
    switch($type){
        case"string":
            $str .= "$name: <input type=text name=string[]><br>";
            break;
        case "text":
            $str .= "$name: <textarea name=text[]></textarea><br>"; break; }
    return $str; }
function Create_form(){
    // funksiya kerakli elementlar bilan forma yaratadi
    global $show_file;
    $str = '<form action="'. $show_file. "'>';
    foreach ($_GET["types"] as $k => $type){
        // elementlar tipini qayta tanlaymiz
        $num = $_GET["numbers"][$k]; // bu elementlar tipining soni
        for ($i=1;$i<=$num;$i++){
            $arr = $_GET["names"][$type][$i-1];
            // yaratilayotgan elementning nomi
            $str .= Create_element($type,$arr);
            // elementni yaratish uchun funksiyani chaqiramiz } }
        $str .= "<input type=submit value=send></form>";
    echo $str; }
    $crt = "Create_form";
    $crt(); // Create_form formasini yaratish
    // funksiyasini chaqiramiz
?>
```

VII bob. FAYLLAR BILAN ISHLASH

Bu bobda fayllar yaratish, fayldan ma'lumotlarni o'qish, fayllarni o'chirish, serverdagi fayllar mavjudligini tekshirish kabi savollarga javob beriladi (fopen, fwrite, fclose, file, fget, unlink, file_exists funksiyalari).

7.1. Ma'lumotlarni saqlash

Ma'lumotlarni saqlashning ikkita asosiy usuli mavjud: ikki o'lchamli faylda va ma'lumotlar bazasida.

Ikki o'lchamli fayl turli formatlarda bo'lishi mumkin, biroq ikki o'lchamli (flat) fayl deganda oddiy matnli faylni tushunamiz. Quyida biz talabalar haqidagi ma'lumotlarni matnli faylga yozishni ko'rib o'tamiz. Bunda har bir talabaning ma'lumotlari bitta satrga yoziladi.

Bu usul oddiy va chegaralangan. Agar katta hajmdagi ma'lumotlar bilan ishlashga to'g'ri kelganda, ma'lumotlar bazasi bilan ishlagan ma'qul. Biroq, ikki o'lchamli fayl keng ko'lamda qo'llaniladi, shuning uchun fayllar bilan ishlash texnologiyalarini o'rganish zarur.

Ma'lumotlarni faylga yozish uch bosqichda amalga oshiriladi:

1. Faylni ochish. Agar fayl mavjud bo'lmasa, uni yaratish talab etiladi.

2. Ma'lumotlarni faylga yozish.

3. Faylni yopish.

Xuddi shunga o'xshash, ma'lumotlarni fayldan o'qish ham shu uch bosqichga bog'liq holda bajariladi:

1. Faylni ochish. Agar faylni ochish mumkin bo'lmasa (masalan, fayl mavjud bo'lmasa), bu holat tekshiriladi va bu holatdan to'g'ri chiqib ketish yo'li qaraladi.

2. Fayldan ma'lumotlarni o'qish.

3. Faylni yopish.

Zarur bo'lganda, fayldan ma'lumotlarni bir o'qishda faylning qancha qismini o'qish kerakligini tanlash mumkin. Keyinroq barcha mavjud imkoniyatlarini ko'rib o'tamiz.

7.2. Faylni ochish va faylga bog'lanish

PHP da faylni ochish uchun `fopen()` funksiyasi ishlatiladi. Faylni ochishda fayl ustida qanday amal olib borish mo'ljallanayotganligi ko'rsatiladi. Bu ruxsat tipi deyiladi.

Server operatsion tizimi ochilgan fayl bilan qanday amal bajarish kerakligini bilishi kerak. Undan quyidagilarni bilishi talab etiladi: balki fayl ochiq bo'lishi va ayni vaqtda agar sahifa egasining ushbu sahifa bilan ishlashga huquqi yetarli bo'lsa fayl ochiq va boshqa bir sahifa bilan qayta ishlanayotgan bo'lishi mumkin. Mantiqan olganda, ruxsat tiplari operatsion tizimga boshqa foydalanuvchilar yoki ssenariylardan tushayotgan so'rovlarni qayta ishlash usulini aniqlash mexanizmini, shuningdek, biror fayl bilan ishlash uchun ruxsat va huquqlarning mavjudligini tekshirish imkoniyatini beradi.

Faylni ochish quyidagi uchta qoidaga asoslanadi:

1. Faylni faqatgina o'qish uchun, faqatgina yozish uchun yoki o'qish va yozish uchun ochiladi.

2. Faylga yozish vaqtida ixtiyoriy fayl tarkibi qayta yoziladi yoki yangi ma'lumot faylning oxiriga yoziladi.

3. Tizimdagi faylga yozish vaqtida ikkilik va matnli fayllar birbiridan farqlanadi, shuning uchun fayl tipi ko'rsatilishi talab etiladi.

Umuman olganda, PHP ning fayllar yaratish uchun belgilangan funksiyasi mavjud emas. Ko'pgina funksiyalari serverdagi fayllar sistemasidagi mavjud fayllar bilan ishlaydi. Lekin, vaqtinchalik fayllarni yaratish uchun bir nechta funksiyalari mavjud.

Eng oddiy faylni yaratish uchun lokal va masofaviy fayllarni ochuvchi funksiyadan foydalaniladi. Bu funksiya `fopen()` deb nomlanadi. `fopen()` berilgan faylni dasturni boshqaruvchi potok bilan bog'laydi.

`fopen()` funksiyasi

Sintaksisi:

```
resource fopen ( fayl_nomi, ruxsat_tipi  
[, use_include_path(yo'lni_qo'shish)])
```

Bu funksiya natijasi sifatida ochilgan faylga ko'rsatkichni (resurs tipi) qaytaradi. Funksiya parametrlari sifatida ochilishi kerak bo'lgan `fayl_nomi`, faylga `ruxsat_tipi` (bu yerda fayl bilan qanday amal bajarilmoqchiligi ko'rsatiladi) va `include_path` (ko'rsatilgan faylni qayerdan qidirish ko'rsatiladi) beriladi. Bu uchta parametr haqida batafsilroq ma'lumot beramiz.

`fayl_nomi` parametri faylning to'g'ri lokal nomi yoki faylning tarmoqdagi URL-manzilidan tarkib topgan satrdan iborat

bo'ladi. Agar fayl_nomi kirish protokolini ko'rsatish bilan boshlansa (masalan, `http://...` Yoki `ftp://...`), u holda interpretator buni URL manzili nomi deb hisoblaydi va qayta ishlovchini ko'rsatilgan URL protokolidan qidiradi. Agar qayta ishlovchi topilgan bo'lsa, u holda PHP xuddi oddiy fayl (`allow_url_fopen` direktivasi) kabi URL obyekti bilan ishlashga ruxsat berilganligini tekshiradi. Agar `allow_url_fopen=off` bo'lsa, u holda `fopen()` xatolikni chaqiradi va ogohlantirishni chiqazadi. Agar fayl_nomi protokol bilan boshlanmasa, u holda fayl_nomi lokal nom sifatida ko'rsatilgan bo'ladi. Lokal faylni ochish uchun, PHP bu faylga mos ruxsat huquqiga ega bo'lishi kerak.

fayl_nomi bir necha xil ko'rinishlarda berilishi mumkin:

agar parametr lokal fayl nomidan iborat bo'lsa, `fopen()` funksiyasi bu faylni ochib, ko'rsatkichni qaytaradi;

agar parametr `php://stdin`, `php://stdout` yoki `php://stderr` ko'rinishida berilgan bo'lsa, mos kirish/chiqish standart potoklari ochiladi;

agar parametr `http://` old qo'shimchasi bilan boshlansa, u holda funksiya serverga HTTP bog'lanishni ochadi va ko'rsatkichni ko'rsatilgan faylga qaytaradi;

agar parametr `ftp://` old qo'shimchasi bilan boshlansa, u holda funksiya serverga FTP bog'lanishni ochadi va ko'rsatkichni (manipulyatorni) ko'rsatilgan faylga qaytaradi. Bu holatda asosiy e'tiborni ushbu ikkita holga qaratish lozim: agar server FTP ning passiv rejimini qo'llamasa, u holda `fopen()` ga murojaat etish muvaffaqiyatsiz yakunlanadi. Xususan, FTP fayllari yoki o'qish uchun yoki yozish uchun ochiladi.

ruxsat_tipi parametri quyidagi qiymatlardan birortasini qabul qilishi mumkin (7.1-jadval).

7.1-jadval

Faylning ruxsat tiplari

Ruxsat tipi	Nomlanishi
r	O'qish rejimi — faylni faqat o'qish uchun ochadi; ko'rsatkich faylga uning boshidan boshlab o'rnatiladi.
r+	O'qish rejimi — fayl o'qish va yozish uchun ochiladi; ko'rsatkich faylga uning boshidan boshlab o'rnatiladi.

w	Yozish rejimi – fayl faqat yozish uchun ochiladi; ko'rsatkich faylni boshidan boshlab o'rnatiladi. Agar fayl mavjud bo'lsa, fayl nol uzunlikka qisqartiriladi, ya'ni tarkibidagilarning barchasi o'chiriladi. Agar fayl mavjud bo'lmasa, uni ochish vaqtida faylni yaratadi.
w+	Yozish rejimi – fayl o'qish va yozish uchun ochiladi; ko'rsatkich faylni boshidan boshlab o'rnatiladi. Agar fayl mavjud bo'lsa, tarkibidagilarning barchasi o'chiriladi. Agar fayl mavjud bo'lmasa, uni ochishga harakat qiladi va natijada faylni yaratadi.
a	Qo'shish rejimi – fayl faqat qo'shish (yozish) uchun ochiladi; ko'rsatkich fayl tarkibining oxiriga o'rnatiladi. Agar fayl mavjud bo'lmasa, uni ochishga harakat qiladi va natijada faylni yaratadi.
a+	Qo'shish rejimi – faylni yozish va o'qish uchun ochadi; ko'rsatkich fayl tarkibini oxiriga o'rnatiladi. Agar fayl mavjud bo'lmasa, uni ochishga harakat qiladi va natijada faylni yaratadi.
x	Faqatgina yozish uchun fayl yaratadi va ochadi; ko'rsatkichni faylning boshiga o'rnatadi. Agar fayl mavjud bo'lsa, u holda fopen() false ni chiqazadi va ogohlantirish beradi. Agar fayl mavjud bo'lmasa, u holda uni yaratishga urinib ko'radi. Ruxsatning bu tipi PHP 4.3.2 versiyasidan boshlab qo'llaniladi va faqatgina lokal fayllar bilan ishlaydi.
x+	Yozish va o'qish uchun fayl yaratadi va ochadi; ko'rsatkichni faylni boshiga o'rnatadi. Agar fayl mavjud bo'lsa, u holda fopen() funksiyasi false ni qaytaradi va ogohlantirish beradi. Agar fayl mavjud bo'lmasa, u holda uni yaratishga urinib ko'radi. Ruxsatning bu tipi PHP 4.3.2 versiyasidan boshlab qo'llaniladi va faqatgina lokal fayllar bilan ishlaydi.

Quyida faylni ochish funksiyasi fopen() ga misol keltirilgan. Fayl ochilmagan holatda, xatolik haqida xabar chiqazish uchun die() chaqiruvi fopen() funksiyasi bilan birgalikda ishlatiladi.

```
<? $file = "user.txt"; // birorta fayl
    $fh = fopen($file, "a+") or die("File ($file) doesnot exist!"); ?>
```

Quyidagi fragment PHP saytiga ulanishni ochadi (<http://www.php.net>):

```
<? $site = "http://www.php.net";
    $sh = fopen($site, "r"); ?>
```

use_include_path parametrida 1 yoki TRUE qiymatlari o'rnatilgan bo'lsa, interpretator fopen() da ko'rsatilgan faylni include_pathdan

qidiradi. `include_path` – bu PHP ni sozlash faylida fayllarni yoqish uchun joylashtirilgan, direktoriyalar ro'yxatida berilgan direktiva. `fopen()` funksiyasidan tashqari bu direktiva `include()` va `require()` funksiyalarida ham ishlatiladi.

7.1-misol. `fopen()` funksiyasining qo'llanilishi

```
<?php
$h = fopen("my_file.html","w");
/* my_file.html fayliga ma'lumot yozish uchun agar u mavjud bo'lsa, uni
ochadi yoki mavjud bo'lmasa shunday nomli bo'sh faylni yaratadi */
$h = fopen("dir/another_file.txt","w+");
/* dir katalogida yozish va o'qish uchun another_file.txt faylini ochadi
yoki yaratadi */
$h = fopen( "http://www.server.ru/dir/file.php","r");
/* ko'rsatilgan manzildagi faylni o'qish uchun ochadi */ ?>
```

Fayl yaratishdan oldin qaysi operatsion tizimda ishlayapsiz va bu faylni qaysi operatsion tizim(OT)da o'qish mumkin, shuni bilish kerak. Gap shundaki, turli operatsion tizimlarda satr tugallanishi turlicha belgilanadi. Unix OTida satr tugallanishi `\n`, Windows tipidagi OTlarda esa satr tugallanishi `\r\n` bilan belgilanadi. Satrlarni Unix tipidagi tizimlardan Windows tizimiga o'tkazish uchun satr tugallanishi belgisini maxsus bayroq `t` sifatida belgilash taklif etiladi. Buning aksini bajaruvchi `b` bayrog'i ham mavjud bo'lib, binar fayllarda tez-tez ishlatiladi. Bu bayroqlarni fayl uchun tanlangan ruxsat tipi belgisi oxirisida yozib ishlatish mumkin. Masalan, o'qish uchun ochilayotgan faylda, satr oxiridagi barcha belgilarni `\r\n` ga qayta kodlashtirish uchun, `r` o'rniga `rt` ni ishlatish kerak. Agar binar fayllarni ochishda `b` bayrog'i ishlatilmasa, u holda mavjud fayllarni o'zgartirish bilan bog'lik xatolar yuzaga keladi. Dasturni turli platformalarga ko'chirish masalalarida, `fopen()` yordamida fayllarni ochishda doimo `b` bayrog'ini ishlatish tavsiya etiladi.

Agar `fopen()` yordamida faylni ochish yoki yaratish amalga oshmasa, u holda PHP da xatolik yuzaga keladi, `fopen` funksiyasi esa o'zining ishi natijasi sifatida `false` qiymatni qaytaradi. Bunday ko'rinishdagi xatolikni `@` belgisi yordamida yashirish mumkin.

Masalan, quyidagi satr, agar faylni ochish amalga oshmasa ham, xatolik haqidagi ogohlantirishni chiqazmaydi:

```
$h = @fopen("dir/another_file.txt","w+");
```


Bu holda `fopen()` funksiyasi faqatgina bo'sh faylni yaratadi va unga yozish uchun ruxsatni ta'minlaydi. Bu faylga ma'lumotlarni qanday yozish mumkin? Mavjud fayldan ma'lumotlarni qanday o'qish mumkin?

Bu savollarga javob berishdan avval `fopen()` yordamida ochilgan bog'lanishni yopishni ko'rib o'tamiz. Fayl bilan ishlab bo'lgandan so'ng har doim `fclose()` funksiyasi yordamida bog'lanishni yopish kerak.

7.3. Fayl bilan bog'lanishni yopish

Fayldagi ma'lumotni o'qish, yozish yoki boshqa biror-bir zarur amallarni bajarish uchun bu fayllar bilan `fopen()` yordamida bog'lanish o'rnatiladi. Bu amallar bajarib bo'lingandan so'ng, albatta bog'lanishni yopish zarur. Buning uchun `fclose()` funksiyasi ishlatiladi.

`fclose()` funksiyasi

Sintaksisi:

`fclose (faylga ko'rsatkich)`

Bu funksiya natijasi sifatida bog'lanishni yopish muvaffaqiyatli amalga oshirilgan bo'lsa, `TRUE` qiymatini, aks holda `FALSE` ni qaytaradi. `fclose()` funksiyasi `fopen()` yoki `fsockopen()` funksiyalari yordamida ochilgan fayllarni muvaffaqiyatli yopadi. Bu funksiya-ning parametri sifatida `fopen()` funksiyasi yordamida ochilgan fayl ko'rsatkichini ko'rsatish kerak.

7.2-misol. `fclose()` funksiyasining ishlatilishi

```
<?php
$h = fopen("my_file.html","w");
fclose($h); ?>
```

Agar fayl bilan bog'lanish yopilmasa, skriptni bajarishda hech qanday xatolik yuzaga kelmaydi. Lekin bu serverda jiddiy oqibat-larga olib kelishi mumkin. Masalan, xakker ochiq bog'lanishdan foydalangan holda faylga virus yozishi mumkin, serverda bu ortiq-cha yo'qotishni yuzaga keltiradi. Shuning uchun fayl bilan zarur amallarni bajargandan so'ng bog'lanishni yopishni maslahat be-ramiz.

Ochiq fayllar bilan ikkita asosiy amal bajarish mumkin — o'qish va yozish. Undan avval esa fayl mavjudligini tekshirib olish zarur.

7.4. Fayl mavjudligini tekshirish

Biz amallar bajarmoqchi bo'lgan fayllar mavjud bo'lmasa, yoki o'qish va yozish uchun ruxsat berilmagan bo'lsa, u holda amallar bajaruvchi funksiyalarning hech biri ishlamaydi va PHP xatolik haqida xabar chiqazadi. Shuning uchun fayl mavjudligini quyidagi funksiyalar yordamida tekshirib olish kerak: `file_exists()`, `is_writable()`, `is_readable()`.

Fayl mavjudligini tekshirish: `file_exists()` funksiyasi

Sintaksisi:

`bool file_exists (fayl yoki katalog nomi)`

`file_exists()` funksiyasi argument sifatida berilgan nomdagi fayl yoki katalog mavjudligini tekshiradi. Agar katalog yoki fayl server fayllar sistemasida mavjud bo'lsa `TRUE`, aks holda— `FALSE` ni qaytaradi. Bu funksiyalar ishining natijasi keshlanadi (kompyuter xotirasidagi jildda saqlab qo'yadi). Ushbu keshlarni tozalash tavsiya etiladi, buning uchun dastur kodiga `clearstatcache()` buyrug'ini qo'shib, muvozanat holatidagi kesh tozalanadi. Lokal bo'lmagan fayllar uchun `file_exists()` funksiyasini ishlatib bo'lmaydi.

7.3-misol. `fclose()` funksiyasining ishlatilishi

```
<? $file = "user.txt";  
if (file_exists($file)){ $fh = fopen($file, "r");  
// bu yerda fayllar bilan amallar bajariladi  
fclose($fh);} else {print "File $file does not exist!";}?>
```

7.4-misol. `file_exists()` funksiyasining qo'llanilishi

```
<?php  
$filename = 'c:/users/files/my_file.html';  
if (file_exists($filename)) { print " <b>$filename</b> fayli mavjud"; }  
else { print " <b>$filename</b> fayli mavjud emas "; } ?>  
Natija: c:/users/files/my_file.html fayli mavjud emas
```

Faylga yozish mumkinligini tekshirish: `is_writable()` funksiyasi
`bool is_writable (string fayl)`

Fayl mavjudligini tekshirishdan tashqari, bu faylga ma'lumot yozish mumkinligini ham tekshirish mumkin, buning uchun `is_writable()` funksiyasi yoki uning psevdonimi — `is_writable()` funksiyasidan foydalaniladi. Yozish imkoniyatining mavjudligi fayl uchun qanday tekshirilsa, katalog uchun ham shunday tekshiriladi.

Sintaksisi:

bool is_writable (fayl yoki katalog nomi)

Bu funksiya agar fayl mavjud bo'lsa va unga yozish mumkin bo'lsa TRUE qiymatini qaytaradi. Faylga ruxsat serverda ishlayotgan foydalanuvchining (ko'pincha bu foydalanuvchi nobody yoki www) foydalanuvchi nomi orqali amalga oshiriladi. is_writable() funksiyasi ishining natijasi keshlanadi.

Faylni o'qish mumkinligini tekshirish: is_readable() funksiyasi

Fayldagi ma'lumotni o'qish mumkinligini tekshirish uchun is_readable() funksiyasidan foydalaniladi. O'qish imkoniyatining mavjudligi fayl uchun qanday tekshirilsa, katalog uchun ham shunday tekshiriladi.

Sintaksisi:

bool is_readable (fayl yoki katalog nomi)

7.5-misol. my_file.txt fayli mavjudligi va uni o'qish uchun ruxsat borligini is_readable() funksiyasi yordamida tekshirish.

```
<?php
$filename = 'c:/users/files/my_file.txt';
if (is_readable($filename)) {
    print "<b>$filename</b> fayl mavjud va o'qish mumkin"; }
else { print "<b>$filename</b> fayl mavjud emas yoki o'qish mumkin
emas"; } ?>
```

Yoki

```
<?
$faylnomi="my_file.txt";
if(is_readable($faylnomi))
// agar fayl mavjud va unga yozish mumkin bo'lsa
{ $fh = fopen($faylnomi,"r"); }
// faylni ochib ko'rsatkichni joriy pozitsiyasini fayl oxiriga o'rnatish
else { print "$faylnomi is not readable!";}
?>
```

7.5. Faylga ma'lumot yozish

fwrite() funksiyasi

fopen() funksiyasi yordamida ruxsat ochilgan faylga ma'lumot yozish uchun fwrite() funksiyasi ishlatiladi.

Sintaksisi:

int fwrite (faylga ko'rsatkich, o'zgaruvchi [, uzunligi])

Bu funksiya o'zgaruvchi tarkibidagi satrlarni faylga ko'rsatkich ko'rsatgan faylga yozadi. Agar qo'shimcha parametr — uzunligi berilgan bo'lsa, u holda yozuv yozilayotgan belgilar soni ushbu parametr qiymatiga teng bo'lganda yoki satr oxiriga yetganda tugatiladi.

fwrite() funksiyasi o'zining ishi natijasi sifatida yozuv bayti sonini yoki xatolik yuzaga kelsa false ni chiqazadi.

7.6-misol. fwrite() funksiyasining qo'llanilishi

```
<?
// foydalanuvchi sayti trafigi haqidagi ma'lumot
$data = "09:15:00|12:30:12|208.277.186.187|Win98";
$filename = "somefile.txt";
// faylni ochib ko'rsatkichni fayl oxiriga o'rnatish
$fh = fopen($filename, "a+");
// agar fayl mavjud va unga yozish mumkinligini tekshirish
if ( is_writable($filename) ) : // $data tarkibini faylga yozish
    $success = fwrite($fh, $data); // faylni yopish
    fclose($fh); else :
    print "Could not open $filename for writing";
endif; ?>
```

7.7-misol. fwrite() funksiyasining qo'llanilishi.

Bizning ishchi katalogimizda my_file.txt fayli mavjud emas. Uni yaratamiz va unga matn satrlarini yozamiz:

```
<? $faylnomi="my_file.txt";
$h = fopen($faylnomi,"w");
$text = "Bu matnni faylga yozamiz.";
if (fwrite($h,$text))
    echo "Yozuv muvaffaqiyatli yakunlandi";
else echo "Ma'lumotlarni yozishda xatolik yuzaga keldi";
fclose($h); ?>
Natija:Yozuv muvaffaqiyatli yakunlandi
```



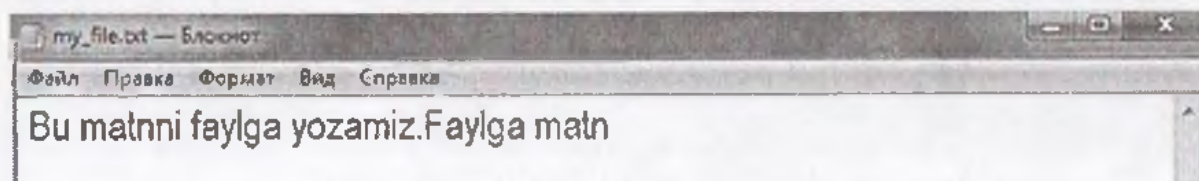
7.1-rasm. Dastur kodi ishga tushgandan so'ng hosil bo'lgan my_file.txt fayli

Bu skript ishi natijasi sifatida agar yozish muvaffaqiyatli amalga oshirilsa, brauzerda my_file.txt faylini ochganimizda “Bu matnni faylga yozamiz.” xabarini ko’rishimiz mumkin. Agar ushbu fayl yuqoridagi skriptni bajarguncha mavjud bo’lgan bo’lsa, u holda undagi barcha ma’lumotlar o’chiriladi va ma’lumot qayta yoziladi.

Agar biz quyidagicha skript yozsak:

```
<?php
$faylnomi="my_file.txt";
$h = fopen("$faylnomi ","a");
$add_text = "Faylga matn qo'shamiz.";
if(fwrite($h,$add_text,1))
echo "Matn qo'shish muvaffaqiyatli amalga oshirildi <br>";
else echo "Ma'lumotlarni qo'shishda xatolik yuzaga keldi <br>";
fclose($h);
?>
```

Natija:Matn qo'shish muvaffaqiyatli amalga oshirildi



7.2-rasm. my_file.txt faylining keyingi ko’rinishi

U holda, avvaldan mavjud bo’lgan my_file.txt faylidagi satr davomiga \$add_text o’zgaruvchisi tarkibidagi satrning 11 ta belgisi qo’shiladi, ya’ni “Faylga matn”.

fwrite()(file — write faylga yozish) funksiyasining u kabi ishlaydigan fputs()(file put string — faylga satrlarni yozish) o’xshashi mavjud. Bu funktsiyani fwrite() funktsiyasi o’rnida ishlatish mumkin va uning sintaksisi fwrite() kabidir. fputs() funktsiyasining sintaksisi:

int fputs (faylga ko’rsatkich, o’zgaruvchi [, uzunligi])

7.1-masala. 7.2-jadvalda berilgan talabalar ma’lumotlarini qayta ishlovchi forma yaratamiz. Bunda har bir talabaning ma’lumotlari bitta faylga yoziladi. Keyin ushbu yozilgan ma’lumotlarni ko’rish uchun web-interfeys yaratamiz.

Ma’lumotlarni faylga yozishda jadvalning har bir qatori faylning alohida satrlariga yoziladi. Satrlarni bir-biridan ajratish uchun yangi satr belgisi (“\n”) ishlatiladi.

Talabalar haqidagi ma'lumotlar

stud_id	surname	name	stipend	kurs	city	birthday	univ_nomi
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	TDPU
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid	150	4	Namangan	7/06/1989	NamDU
10	Kari-mov	Botir	100	2	Jizzax	8/12/1991	TDPU

Ma'lumotlar maydonlari jadvalda berilgan tartibda yoziladi, bu maydonlarni ajratish uchun esa tabulyatsiya belgisi ("\\t") ishlatiladi. Maydonlarni ajratish uchun o'qish oson bo'lgan ixtiyoriy belgini ishlatish mumkin.

Jadvaldagi ma'lumotlarni 7.8 va 7.9-listingda berilgan formalar yordamida faylga yozamiz.

7.8-listing. insert_form.php. Talabalar ma'lumotlarini kiritish uchun forma.

```

<form action="insert.php" method=post>
<h2>Talaba ma'lumotlarini kiriting</h2>
<table border=0 bgcolor=#cccccc>
<tr> <td>Talaba id</td>
<td><input type="text" name="stud_id" size=5 maxlength=5></td>
</tr>
<tr> <td>Familiyasi</td>
<td><input type="text" name="surname" size=30 maxlength=30></td>
</tr>
<tr> <td>Ismi</td>
<td><input type="text" name="name" size=30 maxlength=30></td>
</tr>
<tr> <td>Stipendiyasi</td>

```

```

<td><input type="text" name="stipend" size=5 maxlength=5></td>
</tr>
<tr> <td>Kursi</td>
<td><input type="text" name="kurs" size=5 maxlength=5></td>
</tr>
<tr> <td>Manzili</td>

```



```

        <td><input type="text" name="city" size=30 maxlength=30></td>
    </tr>
    <tr> <td>Tug'ilgan sanasi</td>
    <td><input type="text" name="birthday" size=10 maxlength=10></
td>
    </tr>
    <tr> <td>Universiteti</td>
    <td><input type="text" name="univ" size=30 maxlength=30></td>
    </tr>
    <tr><td colspan=2 align=center><input type="submit" value="Kiriting"></
td>
</tr> </table>

```

The screenshot shows a web browser window with the address bar displaying 'http://localhost/learn_php/insert_form.php'. The browser's menu bar includes 'Файл', 'Правка', 'Вид', 'Избранное', 'Сервис', and 'Справка'. The address bar also shows 'Избранное' and the same URL. The main content area has the title 'Talaba ma'lumotlarini kiriting'. Below the title is a form with the following fields and labels: 'Talaba id', 'Familiyasi', 'Ismi', 'Stipendiyasi', 'Kursi', 'Manzili', 'Tug'ilgan sanasi', and 'Universiteti'. Each field is represented by a text input box. At the bottom of the form is a button labeled 'Kiriting'.

7.3-rasm. Talaba ma'lumotlarini kiritish uchun forma

7.9-listing. Rasmda tasvirlangan formadagi ma'lumotlarni faylga yozuvchi insert.php fayli.

```

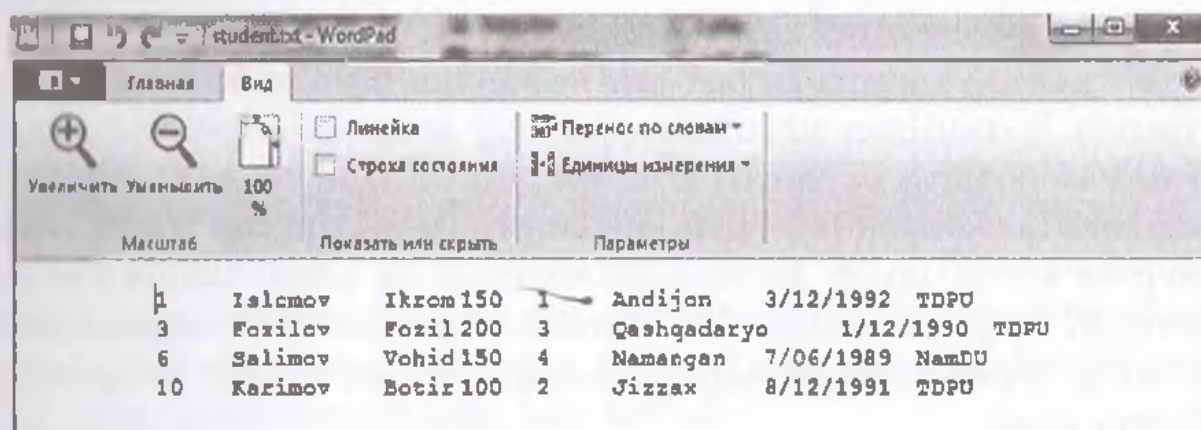
<? $faylnomi="student.txt";
$fh = fopen($faylnomi,"a");
$text = $stud_id."\t".$surname."\t".$name."\t".$stipend."\t".$kurs."\t"
.$city."\t".$birthday."\t".$univ."\n";

```

```

if (fwrite($h,$text)
echo "Yozuv muvaffaqiyatli yakunlandi";
else echo "Ma'lumotlarni yozishda xatolik yuzaga keldi";
fclose($h); ?>

```



7.4-rasm. Talaba ma'lumotlarini kiritilgan student.txt fayli

7.6. Fayldan ma'lumotlarni o'qish

Fayldagi ma'lumotlarni o'qish fayllar bilan ishlashning eng asosiy amallaridan biridir. Quyida fayldan o'qishni amalga oshiruvchi bir qancha funksiyalar berilgan.

Agar mavjud fayldagi ma'lumotni o'qimoqchi bo'lsak, ma'lumot yozishdagi kabi fopen() funksiyasini ishlatish yetarli emas. Bu ochiq faylga ko'rsatkichni qaytaradi xolos, lekin fayldagi birorta satrni o'qimaydi. Shuning uchun fayldan ma'lumotlarni o'qish uchun maxsus funksiyalar (file, readfile, file_get_contents, fread, fgets va h.) dan birortasidan foydalaniladi.

Fayl oxirini aniqlash: feof() funksiyasi

PHP da fayl pozitsiyasi ko'rsatkichi fayl oxiriga kelganligini ko'rishni tekshiruvchi maxsus funksiya mavjud. Bu bulev funksiya — feof(). feof() funksiyasi bitta parametr faylga ko'rsatkichni qabul qiladi.

Sintaksisi:

string feof (faylga ko'rsatkich)

Bu funksiya nomini eslab qolish oson, uning to'liq nomi File End Of File (Fayl: fayl tugadi).

Satrlarni berilgan uzunligiga ko'ra o'qish: fread() funksiyasi

Bu funksiyadan fayldagi ma'lumotlarni o'qishda foydalaniladi. Uni binar fayldagi ma'lumotlarni o'qish uchun ham ishlatish mumkin.

Sintaksisi:

`string fread (faylga ko'rsatkich, uzunligi)`

Bu funksiya faylga ko'rsatkich ko'rsatgan fayldan uzunligi parametrda ko'rsatilgan uzunlikka (baytga) teng ma'lumotni o'qiydi. faylga ko'rsatkich parametri aniq mavjud bo'lgan, masalan, `fopen()` yordamida ochilgan fayl bilan bog'langan resurs tipidagi o'zgaruvchi bo'lishi kerak. faylga ko'rsatkich ochilgan va o'qish uchun ruxsat berilgan faylga yuboriladi. Ma'lumotlarni o'qish fayl oxirigacha yoki uzunligi parametrda ko'rsatilgan baytga teng satrni o'qib bo'lguncha davom etadi.

`fread()` funksiyasining natijasi sifatida fayldagi satrdan o'qilgan ma'lumotlar chiqaziladi.

Bu funksiyada uzunligi parametri albatta ko'rsatilishi kerak. Shunday ekan, agar biz fayldagi barcha satrlarni o'qimoqchi bo'lsak, u holda uning uzunligini bilishimiz kerak. PHP ko'rsatilgan fayl uzunligini hisoblay olish imkoniyatiga ega. Buning uchun `filesize()` funksiyasidan foydalaniladi.

Fayl o'lchamini (uzunligini) aniqlash: filesize() funksiyasi

Fayl o'lchami `filesize()` funksiyasi yordamida aniqlanadi. Bu funksiya fayl o'lchamini baytlarda chiqazadi.

Sintaksisi:

`filesize(fayl nomi)`

Bu funksiya `fread()` funksiyasi bilan birgalikda faylni bir vaqtda barchasini (yoki uning aniq bir qismini) o'qish uchun ishlatiladi.

Xatolik yuzaga kelsa, bu funksiya `false` ni qaytaradi. Afsuski, uni faqatgina lokal fayllarning hajmi (uzunligi)ni olishgagina ishlatish mumkin.

7.10-misol. `fread()` funksiyasining qo'llanilishi.

`my_file.txt` fayli tarkibini o'qiymiz

```
<?php
$fh = fopen("my_file.txt","r+") or die("Can't openfile!");
// faylni yozish va o'qish uchun ochamiz
$content = fread($fh,
```

```

filesize("my_file.txt")); // fayl tarkibidagi satrlarni o'qiydiz
echo filesize("my_file.txt").'<br>'; // faylning uzunligini chiqazamiz
fclose($h); // fayl bilan bog'lanishni yopamiz
echo $content; // brauzer ekraniga fayl tarkibini chiqazish
?>

```

Windows kabi tizimlarda mavjud binar fayl (masalan, rasm)ni o'qish uchun fayl tarkibining oxirida rb yoki uni o'xshashi b bayrog'i yordamida ochish tavsiya etiladi.

filesize() funksiyasi o'z ishining natijasini keshlab qo'yadi (kompyuter xotirasidagi jildga saqlab qo'yadi). Agar my_file.txt fayli tarkibi o'zgartirilib, so'ng yuqoridagi skript qayta ishga tushirilsa, uning ishini natijasi o'zgarmaydi. Skriptni bu fayldagi ma'lumotlarni o'qish uchun boshqa funksiya (masalan, fgetss)dan foydalanib yozsak ham, natija fread() dagi kabi bo'ladi.

Satrma-satr o'qish: fgets(), fgetss(),fgetcsv() funksiyalari

fgets() funksiyasi

fgets() funksiyasi yordamida fayldagi matn satri o'qiladi. Bu funksiyani sintaksisi fread() kabi bo'lib, undan farqli tarafi o'qiladigan satr uzunligini ko'rsatish shart emas:

string fgets (faylga ko'rsatkich [, uzunligi])

fgets() funksiyasi o'z ishi natijasi sifatida faylga ko'rsatkich ko'rsatgan fayldan satr uzunligi (uzunligi-1) baytga teng satrni chiqazadi.

7.11-misol. fgets() funksiyasining qo'llanilishi

```

<? $h = fopen("my_file.txt","r+");
$content = fgets($h,2);
// my_file.txt faylining birinchi satridan birinchi belgini o'qiydi
fclose($h);
echo $content; ?>

```

fread() va fgets() funksiyalarning ikkalasi ham ko'rsatkich agar fayl oxiriga kelsa, fayldan ma'lumotlarni o'qishni to'xtatadi.

Masalan, my_file.txt faylidagi barcha satrlarni quyidagicha o'qish mumkin:

```

<?$h = fopen("my_file.txt","r");
while (!feof ($h)) {
    $content = fgets($h);
    echo $content."<br>"; }
fclose($h); ?>

```


Esga olamiz, PHP da bitta belgi — bu bir bayt. Agar o‘qiladigan satr uzunligi ko‘rsatilmasa (berilayotgan imkoniyat PHP 4.2.0 versiyasidan boshlab kiritilgan), u holda 1 Kbait (1024 bayt) matn yoki 1024 belgi o‘qiladi.

PHP 4.3 versiyasidan boshlab, agar uzunligi parametri berilmasa, satrni hammasi o‘qiladi.

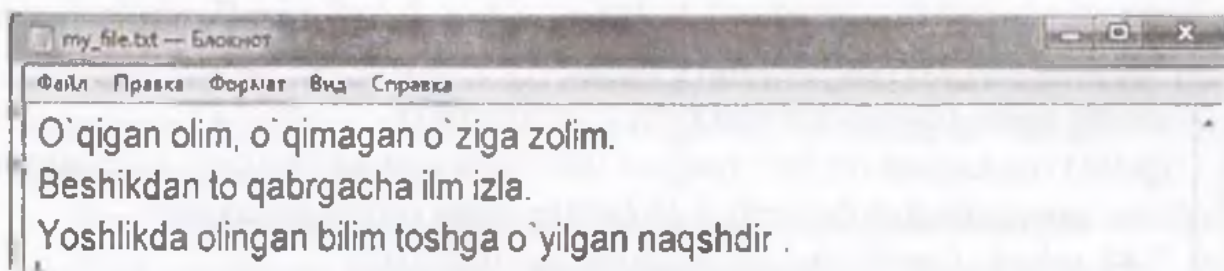
Xatolik yuzaga kelsa `fgets()` funksiyasi `false` ni qaytaradi. PHP ning 4.3 versiyasidan boshlab ikkilik fayllar uchun bu funksiya xavfsiz hisoblanadi.

Quyidagi shartlardan birortasi bajarilsa, o‘qish to‘xtatiladi:

- fayldan (uzunligi-1) bayt yoki belgi o‘qilgan bo‘lsa;
- fayldan keyingi (yangi) satrga o‘tish belgisi o‘qilsa;
- fayldan fayl tugaganligi belgisi (EOF) o‘qilsa.

Agar fayldan har bir satrni o‘qish talab etilsa, u holda ikkinchi parametr sifatida satrdagi baytlar soniga teng qiymatni berish kerak. Fayldan satrma-satr o‘qish va chiqazish uchun misol keltiramiz:

7.12-misol. `my_file.txt` fayli tarkibi quyidagicha bo‘lsin.



7.5-rasm. `my_file.txt` faylining ko‘rinishi

7.5-rasmdagi matnli faylni o‘qib brauzerga chiqazuvchi dastur kodi quyidagicha beriladi:

```
<? $fh = fopen("my_file.txt", "r");  
while (!feof($fh)){ $line = fgets($fh, 4096); print $line. "<br>"; }  
fclose($fh); ?>
```

fgetss() funksiyasi

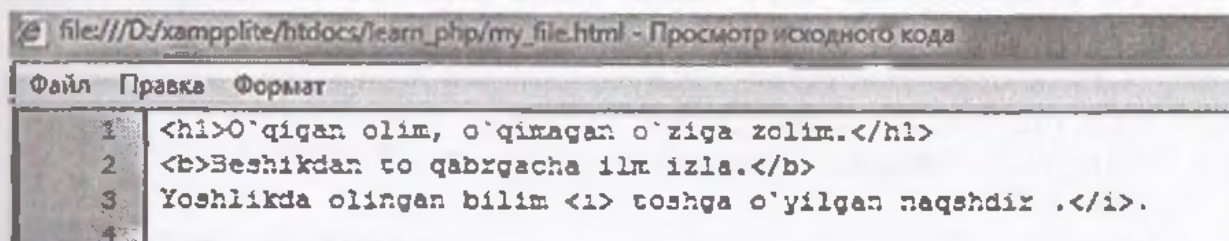
`fgets()` funksiyasining yana bir o‘xshashi bu — `fgetss()` funksiya-sidir. Bu funksiya ham ko‘rsatilgan fayldan satrlarni o‘qish uchun imkon yaratadi va shu bilan birga faylda uchraydigan barcha html va PHP — teglarni o‘chiradi.

Sintaksisi:

string fgetss (faylga ko'rsatkich, uzunligi [, ruxsat berilgan teglar])

E'tibor bering, bu yerda uzunligi parametri albatta bo'lishi kerak.

7.13-misol. my_file.html fayli tarkibi quyidagicha bo'lsin.



```
file:///D:/xampplite/htdocs/learn_php/my_file.html - Просмотр исходного кода
Файл Правка Формат
1 <h1>O'qigan olim, o'qimagan o'ziga zolim.</h1>
2 <b>Beshikdan to qabrgacha ilm izla.</b>
3 Yoshlikda olingan bilim <i> toshqa o'yilgan naqshdir .</i>.
```

7.6-rasm. my_file.html faylining ko'rinishi

Ekranga my_file.html faylining barcha satrlarini va <i> teglaridan boshqa barcha teglarini o'chirib chiqazamiz:

```
<?php
$h = fopen("my_file.html","r");
while (!feof ($h)) {
    $content = fgetss($h,1024,'<b><i>');
    echo $content,"<br>"; }
fclose($h); ?>
```

fgetcsv() funksiyasi

fgetcsv() — funksiyasi fgets() funksiyasining yana bir varianti. U quyidagicha sintaksisga ega:

array fgetcsv (int faylga ko'rsatkich, int uzunligi, string [ajratuvchi]);

Bu funksiya fayldagi satrlarni ajratuvchi sifatida berilgan belgi yordamida ajratish uchun ishlatiladi. Ajratuvchi sifatida ishlatiluvchi belgilar elektron jadvallar va b. larda qo'llanilgan tabulyatsiya belgisi, vergul va h.k.lar bo'lishi mumkin.

Bu funksiyaga fgets() funksiyasiga murojaat etgan kabi murojaat etiladi, faqatgina unga maydonlarni ajratish uchun ishlatilgan ajratuvchi belgi beriladi. Uzunligi parametri fayldan o'qilayotgan eng uzun satrdan katta uzunlikka ega bo'lishi kerak.

Agar berilgan 7.1-masalamizdagi talabaning har bir ma'lumotini matnli satr sifatida emas, balki alohida o'zgaruvchiga tiklash talab

etilsa, `fgetcsv()` funksiyasi ishlatiladi. Ushbu funksiya fayldan satrlarni olib, har bir uchragan tabulyatsiya belgisida satrni ajratadi. Hosil bo'lgan ma'lumotlar massivga joylashtiriladi (ushbu misolda - \$talaba ga). Massivlar bilan ishlashni keyingi bo'limda ko'ramiz.

7.14-listing. 7.1-masalada talaba ma'lumotlari kiritilgan student.txt faylidagi ma'lumotlarni massiv shaklida chiqazamiz.view.php fayli.

```
<? $h = fopen("student.txt","r");
while (!feof($h)) { $talaba = fgetcsv($h,100,"\\t");
    print_r($talaba);
    echo "<br>"; }
fclose($h); ?>
```

Natija:

```
Array ( [0] => 1 [1] => Islomov [2] => Ikrom [3] => 150 [4] => 1 [5] =>
Andijon [6] => 3/12/1992 [7] => TDPU )
```

```
Array ( [0] => 3 [1] => Fozilov [2] => Fozil [3] => 200 [4] => 3 [5]
=> Qashqadaryo [6] => 1/12/1990 [7] => TDPU )
```

```
Array ( [0] => 6 [1] => Salimov [2] => Vohid [3] => 150 [4] => 4 [5]
=> Namangan [6] => 7/06/1989 [7] => NamDU ) Array ( [0] => 10 [1] =>
Karimov [2] => Botir [3] => 100 [4] => 2 [5] => Jizzax [6] => 8/12/1991 [7]
=> TDPU )
```

Belgini o'qish: `fgetc()` funksiyasi

Agar fayldan ma'lumotlarni satrma-satr o'qish mumkin bo'lsa, u holda belgima-belgi o'qish ham mumkin. Buning uchun `fgetc()` funksiyasidan foydalaniladi.

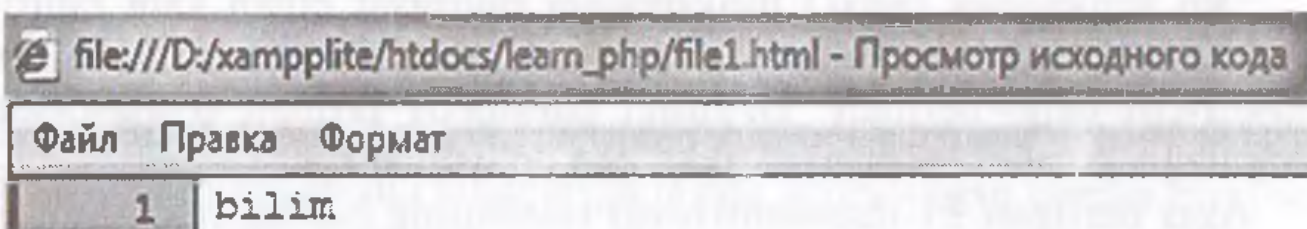
Sintaksisi:

string `fgetc` (faylga ko'rsatkich)

Bu funksiya faylga ko'rsatkichda ko'rsatilgan fayldagi belgilarni chiqazadi va agar satr oxiri uchrasa `FALSE` qiymatini qaytaradi.

7.15-misol. Bu misolda fayl belgilarini `fgetc()` funksiyasi yordamida bittalab o'qish ko'rsatilgan.

file1.html fayli tarkibi quyidagicha bo'lsin:



7.7-rasm. file1.html faylining ko'rinishi

```
<? $h = fopen("file1.html","r");
while (!feof($h)) { $content = fgetc($h);
echo $content. "<br>"; }
fclose($h); ?>
```

Fayl tarkibini o'qish uchun `fopen()` funksiyasi yordamida faylni ochib, bog'lanish o'rnatish shart emas. PHP da buni bitta fayl nomini ishlatib bajara oladigan funksiyalari mavjud. Bu funksiyalar `readfile()`, `file()` va `file_get_contents()`.

7.7. Fayllar bilan ishlovchi funksiyalar

Readfile() funksiyasi

Sintaksisi:

```
int readfile ( fayl nomi [, use_include_path])
```

`readfile()` funksiyasi fayl nomi parametrida berilgan faylni o'qiydi va uning tarkibini, belgilar baytini ekranga chiqazadi. Agar qo'shimcha parametr `use_include_path` `TRUE` qiymatiga teng bo'lsa, u holda berilgan nomdagi faylni `include_path` ga kiruvchi kataloglardan qidiradi. Dasturda bu funksiya faylning o'qilgan belgilari va baytini chiqazadi, aks holda xatolik yuzaga kelganda `FALSE` ni qaytaradi. Xatolik haqidagi xabarni bu funksiyada `@` amali yordamida yashirish mumkin.

7.16-misol. `readfile()` funksiyasining qo'llanilishi.

Quyidagi skript, fayl.txt fayli mavjud bo'lsa, uning tarkibini va hajmini chiqazadi. Aks holda xatolik yuzaga kelganda "Error in readfile" satrini chiqazadi.

```
<?php
$n = @readfile ("somefile.txt");
/* fayl tarkibini ekranga chiqazadi va uning hajmini $n o'zgaruvchisiga
yozib qo'yadi*/
if (!$n) echo "Error in readfile";
/* agar readfile() funksiyasi xatolik bilan bajarilsa,
u holda $n=false va xato haqida xabarni chiqazamiz */
else echo $n;
// agar xato bo'lmasa, o'qilgan belgilar sonini chiqazamiz
?>
```

Natija: Fayllar bilan ishlashni o'rganamiz. 36

readfile() funksiyasi yordamida fayl nomi sifatida berilgan, URL-manzili ko'rsatilgan, uzoq masofadagi fayl tarkibini o'qish mumkin (agar bu opsiya serverni sozlashda o'chirilmagan bo'lsa).

Fayl tarkibini darhol ekranga chiqazish har doim ham qulay emas. Gohida, keyinchalik bu fayl bilan biror-bir amalni bajarish imkoni bo'lishligi uchun fayldagi ma'lumotni o'zgaruvchiga yozish kerak bo'ladi. Buning uchun file() yoki file_get_contents() funksiyalarini ishlatish mumkin.

Fayldagi ma'lumotlarni massivga o'tkazish: file() funksiyasi

file() funksiyasi fayldagi ma'lumotlarni massiv tipidagi o'zgaruvchiga o'tkazib o'qiydi. Bu yerda massivning har bir elementi faylning bitta satriga mos tushadi. Bu funktsiyaning sintaksisi readfile() funksiyasi kabi bo'lib, farqli tarafi u natijasi sifatida massivni chiqazadi.

Sintaksisi:

```
array file ( fayl nomi [, use_include_path])
```

Bu funktsiya qanday massivni chiqazadi? Berilgan massivning har bir elementi biz o'qiyotgan ma'lumotlar fayli(uning nomi fayl nomi parametrida beriladi)ning satri hisoblanadi. file() funksiyasida xatolik yuzaga kelsa, boshqalari kabi false ni qaytaradi. Qo'shimcha parametr use_include_path berilgan fayl include_path katalogida bor yoki yo'qligini aniqlash uchun ishlatiladi. Agar server tomonidan taqiqlanmagan bo'lsa, bu funktsiya yordamida uzoq masofadagi fayllarni ham ochish mumkin. PHP 4.3 versiyasidan boshlab bu funktsiya orqali binar fayllar bilan ishlash xavfsiz hisoblanadi.

7.17-misol. file() funksiyasining qo'llanilishi

7.6-rasmda berilgan my_file.html fayli tarkibini file() funksiyasi yordamida o'qiymiz:

```
<? $file_array = file("my_file.html");  
while (list( $line_num, $line) = each($file_array))  
{print "<b>Satr $line_num:</b> ". htmlspecialchars($line). "<br>\n";  
} ?>
```

Natija:

Satr 0: <h1>O'qigan olim, o'qimagan o'ziga zolim.</h1>

Satr 1: Beshikdan to qabrgacha ilm izla.

Satr 2: Yoshlikda olingan bilim <i> toshga o'yilgan naqshdir.</i>

Satr 3:

Fayl ko'rsatkichi o'rnini o'zgartirish: rewind(), fseek() va ftell()

Fayl ichida fayl ko'rsatkichining o'rnini (pozitsiyasini) aniqlash va uni rewind(), fseek() va ftell() funksiyalari yordamida o'zgartirish.

rewind() funksiyasi fayl ko'rsatkichini faylning boshlanishiga o'rnatadi.

Sintaksisi:

int rewind(faylga ko'rsatkich);

ftell() funksiyasi esa ko'rsatkich o'rnini fayl boshidan boshlab baytlarda aniqlaydi.

Sintaksisi:

int ftell(faylga ko'rsatkich);

7.18-misol. my_file.txt fayli tarkibi quyidagicha bo'lsin.

7.5-rasmdagi faylning tarkibini o'qiymiz:

```
<? $fh = fopen("my_file.txt", "r");  
echo fread( $fh, filesize("my_file.txt" )) ;  
echo "Fayl ko'rsatkichi o'rni ushbu nuqtada - ". (ftell ($fh)); echo  
"<br>";  
rewind($fh);  
echo "rewind dan so'ng, ko'rsatkich o'rni - ". (ftell($fh)); echo "<br>";  
fclose($fh); ?>
```

Natija:

O'qigan olim, o'qimagan o'ziga zolim.

Beshikdan to qabrgacha ilm izla.

Yoshlikda olingan bilim toshga o'yilgan naqshdir.

Fayl ko'rsatkichi o'rni ushbu nuqtada – 124

rewind dan so'ng, ko'rsatkich o'rni – 0

Bu misolda faylni o'qib bo'lgandan so'ng, ko'rsatkich faylni oxirini ko'rsatadi, ya'ni ko'rsatkich satr boshidan 124 bayt siljiydi. rewind() funksiyasiga murojaat etgandan so'ng, ko'rsatkich yana 0 o'ringa – fayl boshiga o'rnatiladi.

fseek() funksiyasi fayl ko'rsatkichini fayl ichidagi ba'zi nuqtalarga o'rnatish uchun ishlatiladi.

Sintaksisi:

int fseek (int faylga ko'rsatkich, int offset);

fseek() funksiyasiga murojaat etgandan so'ng, faylga ko'rsatkich fayl boshidan boshlab offset bayt siljigan nuqtaga o'rnatiladi.

`rewind()` funksiyasi `fseek()` funksiyasiga nol miqdorda siljish uchun murojaat etilganda (`offset=0` da) ekvivalent bo'ladi.

Masalan, `fseek()` funksiyasini fayldagi yozuvlarni o'rtasini topishda yoki binar qidiruvlarni bajarishda ishlatish mumkin.

Faylni satrlarga o'qib olish: `file_get_contents()` funksiyasi

PHP ning 4.3 versiyasidan boshlab fayl tarkibini satrlarga o'qib olish imkoniyati paydo bo'ldi. Bu `file_get_contents()` funksiyasi yordamida bajariladi. Bu funksiya ham parametr sifatida fayl nomini oladi va `use_include_path` ko'rsatkichiga ko'ra uni `include_path` katalogidan qidiradi.

Sintaksisi:

`string file_get_contents (fayl nomi [, use_include_path])`

Bu funksiya `file()` funksiyasiga juda o'xshash, faqatgina u fayl tarkibini satr sifatida chiqazadi. Bundan tashqari u binar ma'lumotlarga ishlov berish uchun xavfsiz va agar server sozlashlarida taqiqlanmagan bo'lsa, masofada joylashgan fayldagi ma'lumotlarni o'qiy oladi.

Faylni o'chirish: `unlink()` funksiyasi

Fayl ustida bajariladigan amallardan biri — bu faylni o'chirish amali. PHP da faylni o'chirish uchun `unlink()` funksiyasidan foydalaniladi.

Sintaksisi:

`bool unlink ( fayl nomi)`

Bu funksiya fayl nomida ko'rsatilgan nomga ega bo'lgan faylni o'chiradi va agar funksiya ishi muvaffaqiyatli amalga oshirilgan bo'lsa `TRUE` ni, aks holda xatolik yuzaga kelsa `FALSE` ni qaytaradi. Faylni o'chirish uchun ham, unga mos ruxsat huquqiga ega bo'lish kerak (masalan, o'qish uchun berilgan ruxsat, faylni o'chirishga ruxsat bermaydi).

7.19-misol. `Unlink()` funksiyasining qo'llanilishi

```
<? $filename = 'd:/xampplite/htdocs/learn_php/my_file.html';  
if (unlink($filename)) {echo $filename.' fayli o'chirildi';}  
else {echo $filename.' fayli o'chirilmadi';}  
// d:/xampplite/htdocs/learn_php/my_file.html nomli faylni o'chirish  
?>
```

Tashqi dasturlarni yurgizish: `yexec()` funksiyasi

PHP ssenariylari serverda joylashgan dasturlarni ham bajarishi mumkin. Bunday imkoniyatlar tizimni web-brauzer orqali boshqa-

rishda tez-tez ishlatiladi hamda tizim haqida ma'lumot olish uchun qulay.

yexes() funksiyasi berilgan dasturni ishga tushiradi va u chiqargan ma'lumotlarni oxirgi satrini qaytaradi.

Sintaksisi:

string exec (string buyruq [, string massiv [, int qaytarish]])

yexes() funksiyasi ishi natijasi sifatida hech narsa chiqarmagan holda, faqatgina buyruqni bajaradi. Buyruq natijasida olingan barcha ma'lumotlar shart bo'lmagan parametr — massivda saqlanadi. Bundan tashqari, agar berilgan massiv parametrida qaytarish o'zgaruvchisi ham berilsa, buyruq bajarilishi so'ngida qaytarish kodlari ushbu o'zgaruvchiga tenglanadi.

7.20-misol. yexes() funksiyasining ishlatilishi

Server bilan aloqani tekshirish

```
<? exec("ping -n 5 www.php.net", $ping);  
for ($i=0; $i< count($ping);$i++){  
print " <br>$ping[$i] "; } ?>
```

7.21-misol. yexes() funksiyasining ishlatilishi

Server bilan aloqani tekshirish

```
<? exec(" ping -n 5 192.168.75.253 ", $ping);  
for ($i=0; $i< count($ping);$i++){  
print " <br>$ping[$i] "; } ?>
```

Fayllar tizimi bilan ishlash: basename() funksiyasi

Serverdagi fayllarni ko'rish va turli amallarni bajarish uchun PHP ning maxsus funksiyasi mavjud. Server fayllari atributlari (turgan joyi, egasi va vakolatlari) haqidagi ma'lumotlar tez-tez kerak bo'ladi. basename() funksiyasi berilgan to'liq nomdan fayl nomini ajratib oladi.

Sintaksisi:

string basename (string to'liq nomi)

faylning tayanch nomini to'liq nomdan ajratib olish uchun quyidagicha yoziladi:

```
<? $path = "/usr/local/phppower/htdocs/index.php";  
$file = basename ($path); // $file = "index.php" ?>
```

Bu funksiya to'liq nomdan unga ko'rsatilgan yo'lni o'chiradi va fayl nomini o'zini qoldiradi.

7.8. Faylni muhofazalash (blokirovkalash)

Faraz qilaylik, bir vaqtning o'zida ikkita foydalanuvchi web-forma yordamida ma'lumotlarni kiritmoqchi. Birinchi foydalanuvchi `fopen()` funksiyasiga murojaat etib, faylga yozishni boshladi, ikkinchi foydalanuvchi ham `fopen()` funksiyasiga murojaat etdi va faylga yozish uchun urinmoqda. Natijada fayl tarkibi qanday bo'ladi? Avval birinchi foydalanuvchi ma'lumoti yoziladi, keyin ikkinchisini-kimi yoki aksincha? Yoki bu ikki ma'lumot ixtiyoriy tarzda galma-gal yoziladimi?

Bu savollarga javob berish ishlatilayotgan operatsion tizimga bog'liq va ularga aniq javob berib bo'lmaydi. Shunga o'xshash muam-molardan qochish uchun fayllarni muhofazalash ishlatiladi. PHP da muhofazalash `flock()` funksiyasi yordamida amalga oshiriladi. Bu funksiyaga faylni ochgandan so'ng, lekin faylga ma'lumotlarni yo-zish va ma'lumotlarni o'qishdan oldin murojaat etiladi.

Sintaksisi:

`bool flock (int faylga ko'rsatkich, int blok_amali);`

Bu funksiyaga parametr sifatida faylga ko'rsatkich va son(`blok_amali`) beriladi. Bu son talab qilinayotgan muhofazalash amalining turi. Agar muhofazalash muvaffaqiyatli amalga ohsa, funksiya `true` qiymatini, aks holda `false` ni qaytaradi.

`blok_amali` parametrining qiymatlari (muhofazalash amalining turi) 9.3-jadvalda berilgan.

7.3-jadval

`flock()` funksiyasining `blok_amali` parametrining qiymatlari

blok_amali parametrining qiymati	Vazifasi
\	O'qishni muhofazalash. Bu shuni anglatadiki, fayl boshqa o'quvchi ilovalar bilan birga (sovmestno) ishlatilishi mumkin.
2	Yozishni muhofazalash. Bu yakka (monopoliya) rejimi. Fayl birga ishlash uchun ruxsatga ega emas.
3	Mavjud muhofazadan chiqarish.
+4	<code>blok_amali</code> parametrining joriy qiymatiga 4 qo'shilsa, joriy muhofazalash bajarilayotgan vaqtda boshqa muhofazalashga urinishlarni bartaraf etadi.

Agar flock() funksiyasini ishlatishga qaror qilingan bo'lsa, uni berilgan fayl ishlatilayotgan tegishli barcha sahifalarga kiritish kerak, aks holda u o'z mazmunini yo'qotadi.

7.22-misol. Muhofazalash uchun misol ko'ramiz, learn.php faylini quyidagicha o'zgartiramiz:

```
<? $qoshish="Yuqorida maqollar berilgan.";  
$fh = fopen("my_file.txt", "a", 1);  
flock($fh, 2); // yozish uchun faylni muhofazalash  
fwrite($fh, $qoshish);  
flock($fh, 3); // yozish uchun muhofazalashni olib tashlash  
fclose($fh); ?>
```

Hamda insert.php sahifasida ham faylni blokirovkalashni qo'shamiz:

```
<? $fh = fopen("my_file.txt", "r");  
flock($fh, 1); // fayldan o'qish uchun faylni muhofazalash flock($fh, 3);  
// muhofazalashni olib tashlash fclose($fh); ?>
```

Endi kod ishonchli, biroq hali ham mukammal emas. Agar ikkita sahifa bir vaqtning o'zida bitta faylni muhofazalashga urinsa, nima bo'ladi? Jarayonlar muhofazalashni o'rnatishda bahslashib qolib, konflikt yuzaga kelishi mumkin va bunga qaysidir biri muvaffaq bo'ladi, biroq o'z navbatida yana yangi muammo tug'iladi. Shuning uchun bunday masalalarni yechishda MBBT ni ishlatiladi, bu kabi masalalar MBBTda nisbatan muvaffaqiyatli yechiladi.

7.9. Serverga faylni yuklash

Endi birmuncha qiyinroq, lekin amalda tez-tez qo'llaniladigan—serverga faylni yuklash masalasini ko'rib o'tamiz. Avvalroq ko'rib o'tgan 9.1-masalamizga o'zgartirish kiritamiz, ya'ni ushbu forma yordamida faqatgina talaba ma'lumotlari emas, balki talabaning rasmi ham kiritilsin. Buning uchun serverga fayl (rasm) ni yuklab oluvchi forma yaratamiz, to'g'rirog'i insert_form.php sahifasiga o'zgartirish kiritamiz. Bu forma yordamida faylni yuklab olish uchun form tegida enctype atributini multipart/form-data qiymati bilan hamda file tipidagi input elementini ishlatamiz.

7.23-misol. Serverga faylni yuklash uchun forma

```
<form enctype="multipart/form-data" action="insert.php" method=post>  
<h2>Talaba ma'lumotlarini kiriting</h2>
```



```

<table border=0 bgcolor=#cccccc>
<tr> <td>Talaba id</td>
<td><input type="text" name="stud_id" size=5 maxlength=5></td>
</tr>
<tr> <td>Familiyasi</td>
<td><input type="text" name="surname" size=30 maxlength=30></td> </
tr>
<tr> <td>Ismi</td>
<td><input type="text" name="name" size=30 maxlength=30></td>
</tr>

<tr> <td>Stipendiyasi</td>
<td><input type="text" name="stipend" size=5 maxlength=5></td>
</tr>
<tr> <td>Kursi</td>
<td><input type="text" name="kurs" size=5 maxlength=5></td>
</tr>
<tr> <td>Manzili</td>
<td><input type="text" name="city" size=30 maxlength=30></td>
</tr> <tr> <td>Tug'ilgan sanasi</td>
<td><input type="text" name="birthday" size=10 maxlength=10></td> </
tr>
<tr> <td>Universiteti</td>
<td><input type="text" name="univ" size=30 maxlength=30></td>
</tr>
<!--faylni yuklash--> <tr> <td>Rasmni yuklang: </td>
<td> <input type="hidden" name="MAX_FILE_SIZE" size="50"
value="30000" /> <input type="file" name="myfile" /></td> </tr>
<tr><td colspan=2 align=center>
<input type=submit value="Kiriting"/></td> </tr> </table>

```

Bu formada yashirin maydon ham ishtirok etadi. Ushbu yashirin maydonda fayl (rasm)ga ruxsat etilgan maksimal hajm baytda ko'rsatiladi. Agar faylni yuklash jarayonida fayl hajmi bu maydonda ko'rsatilgan hajmdan katta bo'lsa, u holda xatolik haqida xabar aks ettiriladi. Brauzerda yaratilgan forma tarkibi matn kiritiladigan satr, lokal diskdan faylni tanlash uchun tugma va faylni jo'natish uchun tugmalardan tashkil topgan.

Endi olingan faylni qayta ishlash uchun skript yozamiz.

Serverga yuklangan fayl haqidagi barcha ma'lumotlar global massiv `$_FILES` tarkibida bo'ladi. Bu massiv PHP 4.1.0 versiyasidan boshlab kiritilgan. Agar `register_globals` direktivasi yoqilgan bo'lsa,

u holda uzatilgan o'zgaruvchilarning qiymati ularning nomi bilan ishlatilishi mumkin.

Yuqoridagi formada Obzor... tugmasini bosib biror-bir fayl (masalan, 16.JPG) yuklab olinib, Faylni jo'nating tugmasi bosiladi. Agar mijoz kompyuteridan 16.JPG nomli hajmi 7759 baytga teng fayl yuklab olingan bo'lsa, u holda `print_r($_FILES);` buyrug'i yordamida quyidagilarni ekranga chiqazish mumkin:

```
Array ( [myfile] => Array ( [name] => 16.JPG  
[type] => image/jpeg [tmp_name] => d:\xampplite\tmp\  
phpEDDA.tmp [error] => 0 [size] => 7759 ) )
```

Umuman olganda, `$_FILES` massivi har doim quyidagi elementlarga ega:

`$_FILES['myfile']['name']` – faylning mijoz mashinasidagi nomi.

`$_FILES['myfile']['type']` – agar brauzer bu ma'lumotni bergan bo'lsa, jo'natilgan faylning mime-tipi. Bizning misolda bu `image/jpeg`.

`$_FILES['myfile']['size']` – yuklangan faylning hajmi (baytda).

`$_FILES['myfile']['tmp_name']` – serverda saqlangan faylning vaqtinchalik nomi.

`$_FILES['myfile']['error']` – yuklash jarayonida yuzaga kelgan xatolar kodi.

Bu yerda 'myfile' – serverga faylni yuklashni amalga oshirishga yordam beruvchi forma elementi nomi. Agar forma elementi nomi boshqa nom bilan nomlansa, bu nom ham boshqacha bo'lishi mumkin. Lekin boshqa kalitlar (name, type va h.) ixtiyoriy boshqa forma uchun o'zgarib qoladi.

Agar `register_globals=On` bo'lsa, u holda `$_FILES['myfile']['name']` ga ekvivalent `$myfile_name` kabi qo'shimcha o'zgaruvchilarni ham ishlatish mumkin bo'ladi.

PHP da faylni yuklashda beshta tipdagi xatolik vujudga keladi va mos ravishda `$_FILES['myfile']['error']` beshta qiymatga ega bo'ladi:

0 – xatolik yuzaga kelmadi va fayl muvaffaqiyatli yuklandi;

1 – yuklanayotgan fayl hajmi `php.ini` sozlash faylidagi `upload_max_filesize` direktivasida o'rnatilgan hajmdan oshib ketgan;

2 – yuklanayotgan fayl hajmi html formadagi `MAX_FILE_SIZE` elementida ko'rsatilgan hajmdan oshib ketgan;

3 — fayl qisman yuklangan;

4 — fayl yuklanmadi.

Agar php.ini sozlash faylining upload_tmp_dir opsiyasi yordamida biror-bir boshqa papka ko'rsatilmasa, fayl odatga ko'ra serverning vaqtinchalik papkasiga yuklanadi. Yuklangan faylni kerakli papkaga o'tkazish uchun move_uploaded_file() funksiyasidan foydalanish mumkin.

Sintaksisi:

bool move_uploaded_file (faylning_vaqtinchalik_nomi, belgilangan_joyi)

Bu funksiya faylning_vaqtinchalik_nomi bilan belgilangan fayl haqiqatda HTTP mexanizmining POST metodi bilan yuklanganligini tekshiradi. Agar shunday bo'lsa, fayl belgilangan_joyi (bu parametr tarkibida yo'l sifatida saqlash uchun yangi papka hamda yangi fayl nomi joylashgan) parametrida berilgan faylga o'tkaziladi.

Agar faylning_vaqtinchalik_nomi da noto'g'ri yuklangan fayl berilsa, u holda hech qanday amal bajarilmaydi va move_uploaded_file() funksiyasi FALSE ni qaytaradi. Agar fayl biror-bir sabab bilan belgilangan_joyi ga o'tkazilmasa ham funksiya yuqoridagi natijani qaytaradi. Bu hollarda interpretator mos ogohlantirishni chiqazadi. Agar fayl berilgan belgilangan_joyi parametrida mavjud bo'lsa, u holda move_uploaded_file() funksiyasi uni qayta yozadi. Demak, yuqoridagi forma yordamida fayl yuklab olindi, vaqtinchalik yuklangan joyidan kerakli katalogga faylni o'tkazamiz. Buni uchun quyidagi skiptni yozamiz va bu sahifani insert.php deb nomlangan, talaba ma'lumotlarini faylga yozuvchi sahifamizga ham o'zgartirish kiritamiz.

7.24-misol. Faylni serverga yuklash dasturi

```
<? //ma'lumotlarni faylga yozish
$faylnomi="student.txt";
$h = fopen($faylnomi,"a");
$text = $stud_id."\t".$surname."\t".$name."\t".$stipend."\t".$kurs."\t".$city."\t".$birthday."\t".$univ."\n";
if (fwrite($h,$text)) echo "Yozuv muvaffaqiyatli yakunlandi";
else echo "Ma'lumotlarni yozishda xatolik yuzaga keldi";
fclose($h); //rasm(fayl)ni yuklash
/* PHP ning 4.1.0 dan avvalgi versiyalarida $_FILES massivi bilan birga
$_HTTP_POST_FILES massivini ham ishlatish kerak*/
```

```
$uploadaddir = 'D:/xampplite/htdocs/learn_php/upload/';
// ushbu papkaga yuklangan rasm(fayl)ni saqlaymiz
$destination = $uploadaddir . $_FILES['myfile']['name'];
$destination = $uploadaddir . $_FILES['myfile']['name'];
// rasm(fayl) nomini o'zgartirmaymiz
print "<pre>";
```

```
if (move_uploaded_file($_FILES['myfile']['tmp_name'],$destination))
{ /* rasmni(fayl)ni vaqtinchalik papkadan saqlash uchun tanlangan papkaga
o'tkazamiz*/
print "Fayl muvaffaqiyatli yuklandi <br>"; }
else { echo "Fayl ni yuklashda xatolik yuzaga keldi. Ba'zi bir
ma'lumotlar:<br>";
print_r($_FILES); }
print "</pre>"; }?>
Natija:
Yozuv muvaffaqiyatli yakunlandi
Fayl muvaffaqiyatli yuklandi
```

Natijada fayl vaqtinchalik yuklangan joyidan ko'rsatilgan papkaga o'tkazildi. Ya'ni fayl ni yuklab olish muvaffaqiyatli bajarildi.

7.10. Qayta ishlashning ratsional usuli. Ma'lumotlar bazasini boshqarish tizimi

Barcha ko'rib o'tilgan misollarda ikki o'lchamli fayllar ishlatildi. Keyingi bo'limda MySQL — relyatsion ma'lumotlar bazasini boshqarish tizimini ko'rib o'tamiz. Savol tug'ilishi mumkin: "Bu nima uchun kerak?"

Ikkilik fayllar (dvumernix faylov)da ishlash bilan bog'liq muammolar:

- Ikki o'lchamli fayl kattalashgan sari u bilan ishlash sekinlashadi.
- Aniq bir yozuvni yoki yozuvlar guruhini fayldan qidirish qiyin. Agar yozuvlar tartiblangan bo'lsa, u holda qidirish uchun kalitli maydonda yozuvning fiksirlangan uzunligini qo'llash bilan birga bina qidiruv turining biror-biri ishlatiladi. Agar biror-bir shablona mos ma'lumotni qidirish kerak bo'lsa (masalan, Andijonda yashovchi barcha talabalarni), u holda har bir yozuvni alohida o'qish va tekshirish kerak bo'ladi.
- Raqobatli ruxsat (berilgan ruxsat bo'yicha bir vaqtning o'zida ikki yoki undan ortiq foydalanuvchi bitta amalni bajarishi) muam-

molar keltirib chiqarishi mumkin. Xuddi avval ko'rib o'tilgan, fayl-ni muhofazalashda yuzaga kelgan konflikt hodisalar kabi. Bundan tashqari, tarmoqdagi "nozik nuqtalarni" paydo bo'lishiga sabab bo'ladi. Saytdagi ko'p miqdordagi foydalanuvchilardan yetarli darajada shiddat bilan ishlayotgan ma'lumotlar oqimiga o'zlarining ma'lumotlarini kiritish uchun kutish talab etiladi (chunki bu vaqtda faylning muhofazasini olib tashlash ishi bajarilayotgan bo'ladi). Agar kutish uzoq vaqt davom etsa, foydalanuvchilar boshqa saytlardan foydalanishni ma'qul ko'rishadi.

- Qaralayotgan fayllarni qayta ishlash ketma-ketlikda amalga oshirilishi kerak — o'qish faylning boshidan boshlab, uning oxirigacha bajariladi. Agar yozuv qo'shish yoki yozuvni faylning o'rtasidan o'chirish zarur bo'lsa, u holda avval faylning barchasini xotiraga o'qib olish, o'zgartirishni bajarish va faylga qaytadan yozishga to'g'ri keladi, bu oxir-oqibat mushkul ishga aylanadi. Katta hajmdagi ma'lumotli fayllar bilan ishlashda bu jarayon sezilarli darajada tizimning ortiqcha yuklanishiga sabab bo'ladi.

- Fayllar bilan ishlayotganda chegaralanishdan tashqari, ma'lumotlarga turli darajadagi foydalanish huquqlarini ta'minlashning hech qanday usuli yo'q.

Bu muammolar RMBBTda qanday yechiladi?

Relyatsion ma'lumotlar bazasini boshqarish tizimi (RMBBT) ushbu barcha muammolarni hal etadi:

- RMBBT ma'lumotlardan foydalanishni ikkilik faylga nisbatan tezroq amalga oshiradi. MySQL ma'lumotlar bazasini boshqarish tizimi esa barcha RMBBT ichida eng ko'p foydalanilayotgan tizimlardan biridir.

- RMBBT da aniqlangan mezonlar asosida ma'lumotlar to'plamini olish uchun so'rovlar oson jo'natiladi.

- RMBBT odatda ko'p sonli so'rovlar bilan ishlash mexanizmi-ga ega bo'ladi, bu esa RMBBT dan foydalanuvchi dasturchini ko'p sonli so'rovlar haqida o'ylashdan ozod etadi.

- RMBBT ma'lumotlardan ixtiyoriy tarzda foydalanishni ta'minlaydi.

- RMBBTda foydalanuvchi huquqlarini ta'minlash tizimi mavjud bo'lib, MySQL esa bu borada alohida imkoniyatlarga ega.

Ko'rinib turibdiki, ma'lumotlarni saqlash tizimini tashkil etishda RMBBT dan foydalanish funksional imkoniyatlarni beradi.

VIII bob. MASSIVLAR

PHP ning massiv elementlari bilan ishlovchi bir qancha funksiyalari mavjud va ular: massiv elementini qidirish, massiv elementini saralash; bir yoki bir nechta, birinchi yoki oxirgi elementlarni qo'shish va o'chirish; massiv hajmini kattalashtirish; standart elementlarni qo'shish kabi imkoniyatlarga ega. Bu funksiyalar massivni saralash bilan bog'liq tez-tez uchrab turadigan masalalarni yechishda yordam beradi. Bu bobda bunday funksiyalarning bir nechtasini ko'ramiz va ular yordamida bir qancha amaliy masalalarni yechamiz. Massiv elementini qidirish, massiv elementini saralash.

8.1. Massivlar

Oldingi ma'ruzalarda massiv tipidagi ma'lumotlar haqida gapirib o'tganmiz. Massivni ikki xil usul bilan yaratish mumkin:

`Array()`

`array()` funksiyasi yordamida

`$array_name = array ([value1, value2,...]);`

Assotsiativ massivlarda

`$array_name = array("key1"=>"value1", "key2"=>"value2");`

2. to'g'ridan to'g'ri qiymat berish yo'li bilan

`$array_name []=value1;`

Assotsiativ massivlarda

`$array_name["key1"] = value1;`

Bizdan ma'lumotlar bazasidan o'chirilishi kerak bo'lgan fanlar ro'yxatini saqlash talab etilgan bo'lsin. Ularni massiv ko'rinishida saqlash mumkin: bu yerda kalit — fan identifikatori (uni unikal nomeri), qiymat esa — fan nomi. Bu massivni quyidagicha yaratish mumkin:

```
<? $del_items = array("10"=>"Fizika", "12"=>"Informatika");  
$del_items["13"] = "Php da dasturlash"; // massivga element qo'shamiz  
>
```

8.1-masala. Avvalgi bo'limda ko'rib o'tgan talabalar haqidagi ma'lumotlar joylashgan jadval ham massiv tipidagi ma'lumotlarga misol bo'la oladi. Massivlar bilan ishlashga doir misollarda ushbu jadvaldan foydalanamiz.

Talabalar haqidagi ma'lumotlar

stud_id	surname	name	stipend	kurs	city	birthday	univ_nomi
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	TDPU
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid	150	4	Namangan	7/06/1989	NamDU
10	Karimov	Botir	100	2	Jizzax	8/12/1991	TDPU

Ushbu jadvalda berilgan talabalarning ismlaridan tashkil topgan massivni yozamiz:

```
<? $talabalar=array("Ikrom", "Fozil", "Vohid", "Botir");?>
```

Bu massiv tarkibini ko'rish uchun o'zgaruvchi nomi va kalit (indeks) kerak bo'ladi. Ushbu massiv tarkibini chiqazamiz.

```
echo "$talabalar[0] $talabalar[1] $talabalar[2] $talabalar[3]";
```

Agar massiv ketma-ket sonlar bilan indekslangan bo'lsa, uning tarkibini chiqazish uchun sikldan foydalanish mumkin:

```
<? $talabalar=array("Ikrom", "Fozil", "Vohid", "Botir");
for ($i=0; $i<4; $i++)
{ echo "$talabalar[$i] "; } ?>
Natija:Ikrom Fozil Vohid Botir
```

Ko'p o'lchovli massivlar

Endi jadvaldagi ma'lumotlarni barchasini massiv shaklida aniqlaymiz va quyidagicha PHP kodini yozamiz:

```
<? $talabalar=array(
array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992",
"TDPU"),
array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990",
"TDPU"),
array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989", "Nam-
DU"),
array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", TDPU));
?>
```

Bundan ko'rinib turibdiki, talabalar massivi to'rtta massivdan tashkil topgan, ya'ni ko'p o'lchovli massiv. Bir o'lchovli massivga murojaat etishda massiv nomi va uning indeksidan foydalangan edik. Ikki o'lchovli massivda esa, har bir elementning ikkita indksi

bo'ladi — satri va ustuni. Bu massivning tarkibini ko'rish quyidagicha amalga oshiriladi:

```
echo      "|"$. $talabalar[0][0]."|"$. $talabalar[0][1]."|"$. $talabalar[0][2]."|"$. $talabalar[0][3]."|"$. $talabalar[0][4]."|"$. $talabalar[0][5]."|"$. $talabalar[0][6]."|"$. $talabalar[0][7]."|<br>;
echo      "|"$. $talabalar[1][0]."|"$. $talabalar[1][1]."|"$. $talabalar[1][2]."|"$. $talabalar[1][3]."|"$. $talabalar[1][4]."|"$. $talabalar[1][5]."|"$. $talabalar[1][6]."|"$. $talabalar[1][7]."|<br>;
echo      "|"$. $talabalar[2][0]."|"$. $talabalar[2][1]."|"$. $talabalar[2][2]."|"$. $talabalar[2][3]."|"$. $talabalar[2][4]."|"$. $talabalar[2][5]."|"$. $talabalar[2][6]."|"$. $talabalar[2][7]."|<br>;
```

Yoki ushbu natijani olish uchun sikldan foydalanish ham mumkin:

```
<? $talabalar=array(
    array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992",
    "TDPU"),
    array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990",
    "TDPU"), array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989",
    "NamDU"),
    array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU")
);
for($row=0;$row<4;$row++)
{ for($column=0; $column<8; $column++)
{ echo "|"$. $talabalar[$row][$column]; }
echo "|<br>; } ?>
Natija:
|1|Islomov|Ikrom|150|1|Andijon|3/12/1992|TDPU|
|3|Fozilov|Fozil|200|3|Qashqadaryo|1/12/1990|TDPU|
|6|Salimov|Vohid|150|4|Namangan|7/06/1989|NamDU|
|10|Karimov|Botir|100|2|Jizzax|8/12/1991|TDPU|
```

Jadvaldagi ma'lumotlarning massivini ustun nomlarini ishlatib ham hosil qilish mumkin. Bunday massiv assotsiativ massiv bo'ladi. Bunday massivning tarkibini ko'rish uchun ham sikllardan foydalanish mumkin:

```
<? $talabalar=array(
    array(stud_id=>"1",surname=>"Islomov", name=>"Ikrom", sti-
pend=>150, kurs=>1, city=>"Andijon", birthday=>"3/12/1992", univ_
id=>"TDPU"),
    array(stud_id=>"3",surname=>"Fozilov", name=>"Fozil", sti-
pend=>200, kurs=>3, city=>"Qashqadaryo", irthday=>"1/12/1990", univ_
id=>"TDPU"),
```



```

    array(stud_id=>"6",surname=>"Salimov",           name=>"Vohid",
    stipend=>150, kurs=>4, city=>"Namangan", birthday=>"7/06/1989", univ_
    id=>"NamDU"),
    array(stud_id=>"10",surname=>"Karimov",           name=>"Botir",
    stipend=>100, kurs=>2, city=>"Jizzax", birthday=>"8/12/1991", univ_
    id=>"TDPU"));
    for($row=0;$row<4;$row++)
    {echo "|".$talabalar[$row]["stud_id"]."|".$talabalar[$row]
    ["surname"]."|".$talabalar[$row]["name"]."|".$talabalar[$row]
    ["stipend"]."|".$talabalar[$row]["kurs"]."|".$talabalar[$row]
    ["city"]."|".$talabalar[$row]["birthday"]."|".$talabalar[$row]["univ_
    id"]."<br>"; }?> Natija:
    |1|Islomov|Ikrom|150|1|Andijon|3/12/1992|TDPU|
    |3|Fozilov|Fozil|200|3|Qashqadaryo|1/12/1990|TDPU|
    |6|Salimov|Vohid|150|4|Namangan|7/06/1989|NamDU|
    |10|Karimov|Botir|100|2|Jizzax|8/12/1991|TDPU|

```

8.2. Massivlar ustida amallar

Massiv – bu ma’lumot tipi, ma’lumotlarning bu tipi ustida amallar bajarish mumkin. Massivlarni qo’shish va taqqoslash mumkin.

Massivlarni qo’shish uchun “+” operatoridan foydalanamiz. Umuman olganda, bu amal massivlarni birlashtirish amali. Bizda ikkita \$a va \$b massivlari mavjud, ularni qo’shish (birlashtirish) natijasi \$a va \$b massivlardan tashkil topgan \$c massivga teng bo’ladi. Agar mos kalitlar uchrashsa, u holda natijada birinchi massiv (ya’ni, \$a) ning elementlari olinadi.

8.1-misol. Massivlarni qo’shish

```

<? $a = array("i"=>"Informatika", "m"=>"Matematika");
$b = array("i"=>"Ingliz tili", "m"=>"Manbashunoslik", "f"=>"Fizika");
$c = $a + $b;
$d = $b + $a;
print_r($c).<br>';
/* Array([i]=>Informatika [m]=>Matematika [f]=>Fizika) */
print_r($d);
/* Array([i]=>Ingliz tili [m]=>Manbashunoslik [f]=>Fizika) */ ?>

```

Massivlarni teng yoki teng emas, ekvivalent yoki ekvivalent emasligini tekshirish yordamida taqqoslash mumkin.

Barcha massiv elementlarining kalit/qiymat juftliklari mos tushsa, bu massivlar teng deyiladi.

Solishtirilayotgan massivlarning kalit/qiymat juftliklari bir xil tartibda yozilsa va teng bo'lsa, u holda bu massivlar ekvivalent massivlar deyiladi.

PHP da teng qiymati - "==" belgisi bilan, ekvivalent esa - "===" belgisi bilan belgilanadi.

10.2-misol. Massivlarni taqqoslash

```
<? $a = array("i"=>"Informatika", "m"=>"Matematika");  
$b = array("m"=>"Matematika", "i"=>"Informatika");  
if ($a == $b) echo "Massivlar teng va";  
else echo " Massivlar teng emas va";  
if ($a === $b) echo " ekvivalent";  
else echo " ekvivalent emas";  
// " Massivlar teng va ekvivalent emas "natijasini olamiz  
?>
```

8.3. Massivlar bilan ishlovchi funksiyalar

each() funksiyasi

Bu funksiya massivning joriy elementini qaytaradi (chiqazadi) va keyingi elementini joriy deb belgilaydi.

Sintaksisi:

each(massiv);

each() funksiyasiga while sikli ichida murojaat etilganda, u navbatma-navbat massivning har bir elementini qaytaradi va massiv oxiriga yetgach o'z ishini to'xtatadi.

8.2-misol.

```
<? $talabalar=array(stud_id=>"1",surname=>"Islomov",name=>"Ikrom",  
stipend=>150, kurs=>1, city=>"Andijon", birthday=>"3/12/1992", univ_  
id=>"TDPU");  
while( $element = each( $talabalar ) )  
{echo $element[ "key" ]; echo " - ";  
echo $element[ "value" ]; echo "<br>"; } ?>
```

Natija: stud_id - 1

surname - Islomov


```
name - Ikrom
stipend - 150
kurs - 1
city - Andijon
birthday - 3/12/1992
univ_id - TDPU
```

Bu misolda \$element o'zgaruvchisi ham massiv elementlarini qabul qilib olayotganligi uchun massiv bo'ladi. each() funksiyasiga murojaat etganda, u sakkizta qiymatdan va sakkizta indeksdan iborat kataklardan tashkil topgan massivni beradi. key va 0 katagi joriy elementning kaliti, value va 1 katagi — joriy elementning qiymatlaridan tarkib topgan. Kataklardan qaysi birini tanlash ahamiyatsiz, ushbu misolda raqamlangan kataklarni emas, nomli kataklarni ishlatdik, ya'ni assotsiativ massivdan foydalandik.

List() funksiyasi

List() funksiyasi array() ga o'xshash bo'lib, uning asosiy vazifasi — bir vaqtning o'zida massiv ichidagi bir nechta qiymatni bir nechta o'zgaruvchiga tenglashtiradi. Bu funksiya berilgan massivni bir nechta qiymatlarga ajratadi.

Sintaksisi:

```
void list(o'zgaruvchi1 [,o'zgaruvchi2 , ...])
```

list() — ma'lumotlar bazasidan yoki fayldan ma'lumotlarni o'zgaruvchiga o'qib olish uchun juda qulay funksiya.

Bu yerda \$talaba1= Lola, \$talaba2=Karim, \$talaba3= Vasila, \$talaba4= Akmal, \$talaba5= Botir

8.3-misol. list() funksiyasini qo'llash. 10.1-jadvaldan foydalanamiz.

```
<? $talabalar=array("1", "Islomov", "Ikrom", "150", "1", "Andijon",
"3/12/1992", "TDPU");
list($stud_id, $surname, $name, $stipend, $kurs, $city, $birthday, $univ_id)=$talabalar;
echo $stud_id.<br>; echo $surname.<br>;
echo $name.<br>; echo $stipend.<br>;
echo $kurs.<br>; echo $city.<br>;
echo $birthday.<br>; echo $univ_id;?>
```

8.4-misol. List() funksiyasini each() funksiyasi bilan qo'llaymiz.

```

<?
$stalabalar=array(stud_id=>"1", surname=>"Islomov", name=>"Ikrom",
stipend=>150, kurs=>1, city=>"Andijon", birthday=>"3/12/1992", univ_
id=>"TDPU");
while( list($key, $value) = each( $stalabalar ) )
{echo "$key - $value <br>"; } ?>
Natija: stud_id - 1
surname - Islomov
name - Ikrom
stipend - 150
kurs - 1
city - Andijon
birthday - 3/12/1992
univ_id - TDPU

```

Bu misolda each() funksiyasi \$stalabalar massividan joriy elementni olib, uni massiv sifatida qaytaradi va keyingi elementni joriy etadi. List() funksiyasi esa each() funksiyasi qaytargan massivning 0 - va 1 - elementini ikkita yangi o'zgaruvchiga (\$key va \$value) tenglaydi.

range() funksiyasi

range() funksiyasi quyi va yuqori chegarasi aniq bo'lgan butun sonlar intervalini oson va tez yaratish imkonini beradi. range() funksiyasi ko'rsatilgan intervaldagi barcha butun sonlardan tashkil topgan massivni chiqazadi.

Sintaksisi:

array range(int quyi_chegara, int yuqori_chegara).

8.5-misol. range() funksiyasini qo'llash

```

<? $slot= range(0,9);
// $slot=array(0,1,2,3,4,5,6,7,8,9). ?>

```

range() funksiyasida 0 dan 9 gacha bo'lgan interval ko'rsatilganligi uchun \$slot massivi ko'rsatilgan intervaldagi butun sonlardan iborat bo'ladi.

Massiv elementlarini hisoblash: sizeof(),count(),array_count_values()

sizeof() funksiyasi

Massivlarni ustida yana bir muhim amalni bajarish mumkin – massiv elementlari sonini aniqlash. sizeof() funksiyasi – massiv elementlari sonini hisoblaydi. Yana bir shunga o'xshash funksiya sizeof() ning kengaytirilgan formasi – bu count() funksiyasidir.

count() funksiyasi

count() funksiyasi sizeof() ga o'xshash bo'lib, massivdagi qiymatlarning sonini hisoblaydi. Umuman olganda, bu funksiya o'zgaruvchining elementlari sonini hisoblaydi.

Agar bu funksiyaning massiv o'zgaruvchidan boshqa biror-bir o'zgaruvchi uchun qo'llanilsa, natija 1 chiqadi. NULL tipidagi o'zgaruvchi esa bundan mustasno, ya'ni count(NULL) 0 ga teng.

Bu funksiya ko'p o'lchovli massiv elementlari sonini hisoblash uchun qo'llanilganda, qo'shimcha parametr COUNT_RECURSIVE ishlatiladi.

Sintaksisi:

count(o'zgaruvchi, [COUNT_RECURSIVE]);

8.6-misol. count() funksiyasini qo'llash

```
<? $del_items = array("langs" => array( "10"=>"Python", "12"=>"Lisp"),  
"other"=>"Infromatika");  
echo count($del_items) . "va"; // 2 chiqadi  
echo count($del_items,COUNT_RECURSIVE); // 4 chiqadi ?>  
Natija: 2 va 4
```

array_count_values() funksiyasi

Sintaksisi:

array_count_values (o'zgaruvchi);

array_count_values(\$ar) funksiyasi \$ar massivida unikal qiymat necha marta uchraganligi sonini chiqazadi.

8.7-misol. array_count_values() funksiyasini qo'llash

```
<? $ar=array(4,5,1,2,3,1,2,2,1,2,1,1);  
$nat=array_count_values($ar); echo '<table>';  
echo '<tr><td>qiyamatlar</td><td>soni</td></tr>';  
for($i=1;$i<=5;$i++){  
echo '<tr><td>'. $i. '</td><td>'. $nat[$i]. '</td></tr>'; }  
echo '</table>'; ?>
```

Natija shuni ko'rsatadiki, \$ar massivida 1 - 5 marta, 2 - 4 marta, 3,4,5 qiymatlari 1 marta uchragan.

Massiv elementlari bilan ishlash: In_array(), array_search(), array_keys(), array_values(), array_unique()

In_array() funksiyasi

in_array() — massiv elementlarini qidiruvchi funksiya. Berilgan massivda qidirilayotgan qiymat mavjudligini tekshiradi.

Sintaksisi:

```
in_array("qidirilayotgan_qiymat", "massiv", ["tip chegarasi"]);
```

Agar uchinchi parametr true sifatida berilgan bo'lsa, u holda massivdan faqatgina qiymati bo'yicha emas, balki tipi bo'yicha ham mos elementni qidiradi. Agar qidirilayotgan qiymat – satr bo'lsa, u holda taqqoslanayotganda registrga ahamiyat beradi.

8.8-misol. Berilgan massiv elementlari fanlar ro'yxatidan tashkil topgan. Berilgan massivda Informatika fani mavjudmi yoki yo'q qidiramiz.

```
<?php
$dfan = array("Matematika","Fizika","Falsafa","Informatika","Tarix");
if (in_array("Informatika",$dfan)) echo "Informatika fanidan imtihon
topshiraman<br>";
// " Informatika fanidan imtihon topshiraman" xabarini chiqazadi
if (in_array("informatika",$dfan)) echo " informatika fanidan imtihon
topshiraman<br>";
/* hech narsa chiqmaydi, chunki " Informatika " satri mavjud, lekin "
informatika " mavjud emas. Ya'ni bu funksiya registrga sezgir */ ?>
Natija:Informatika fanidan imtihon topshiraman
```

In_array() funksiyasida qidirilayotgan qiymat massiv shaklida ham bo'lishi mumkin. Faqatgina bu imkoniyat PHP 4.2.0. versiyasidan boshlab qo'shilgan.

8.9-misol.

```
<?php
$dfan = array("Matematika","Fizika",array("Falsafa","Informatika"),"T
arix");
if (in_array(array("Informatika","Falsafa"),$dfan)) echo "Informatika va
Falsafa fanlaridan imtihon topshiraman<br>"; ?>
Natija:Informatika va Falsafa fanlaridan imtihon topshiraman
```

array_search() funksiyasi

Ushbu funksiya ham in_array() funksiyasi kabi massiv elementlarini qidiradi. array_search() funksiyasining in_array() funksiyasidan farqli tarafi agar element topilgan bo'lsa, kalitini chiqazadi, aks holda false ni qaytaradi.

Sintaksisi:

```
array_search("qidirilayotgan_qiymat", "massiv", ["tip chegarasi"]);
```

Satrlarni solishtirish registrga sezgir, agar uchinchi parametr ko'rsatilgan bo'lsa, u holda qiymat tipini ham solishtiradi. PHP ning

4.2.0 versiyasigacha, agar qidirilayotgan qiymat topilmasa, funksiya xato yoki bo'sh NULL qiymatni qaytaradi.

8.10-misol. Endi yuqoridagi misolning teskarisini ko'rib o'tamiz, imtihon topshirilgan fanlar massivi bor. Massivda imtihon topshirilgan fanlar tartibi massiv elementi kaliti bilan mos ravishda berilgan.

```
<?php
$dfan = array("Matematika","Fizika","Falsafa", "Informatika","Tarix");
if (!array_search("Informatika",$dfan))
    echo "Informatika fanidan imtihon topshirilmagan.<br>";
else { $k = array_search("Informatika",$dfan);
    $n=$k+1;
    echo "Informatika imtihon topshirgan fanlarimning $n - si"; } ?>
Natija:Informatika imtihon topshirgan fanlarimning 4 - si
```

Bu funksiya `in_array()` funksiyasiga nisbatan funksionalroq, chunki bu funksiya yordamida qidirilayotgan element nafaqat massivda mavjudligini, balki u massivning qayerida joylashgani haqida ham bilishimiz mumkin. Agar qidirilayotgan element massivda bir nechta bo'lsa, nima qilamiz? Bu holatda `array_search()` funksiyasi birinchi topilgan element kalitini chiqazadi. Barcha element kalitlarini olish uchun esa `array_keys()` funksiyasidan foydalaniladi.

array_keys funksiyasi

`array_keys()` funksiyasi massivning barcha kalitlarini chiqazadi. Qo'shimcha parametr sifatida berilgan aniq qiymat yordamida element kalitlari ro'yxatini olish mumkin.

Sintaksisi:

```
array_keys("massiv", ["qidirish uchun qiymat"])
```

`array_keys()` funksiyasi massivning sonli kalitlari kabi satrli kalitlari qiymatlarini ham chiqazadi. Bunda qiymatlardan hosil bo'lgan yangi massiv sonli indeksli massiv ko'rinishida bo'ladi.

8.11-misol. `array_keys()` funksiyasining qo'llanilishi:

Fanlar massivi mavjud. Ushbu massiv juda ko'p elementlardan tashkil topgan shuning uchun ro'yxatdagi ba'zi fanlar tasodifan takrorlanishi mumkin. Shu fanlardan biri — Fizika fani takrorlangani haqida shubha tug'ildi. Takrorlanganmi yoki yo'qligini tekshirib ko'ramiz.

```
<? $dfan = array("Matematika","Fizika",
  Falsafa", "Informatika","Tarix","Fizika");
$fan_keys = array_keys($dfan,"Fizika");
echo "Fizika massivda ". count($fan_keys) ." marta yozilgan.<br>";
foreach ($fan_keys as $val){
  echo " $val – o'rinda <br>"; } ?>
```

Natija:

Fizika massivda 2 marta yozilgan.

1 – o'rinda

5 – o'rinda

array_keys(), funksiyasi avvalgi ikkita funksiya kabi registrga sezgir, masalan, FIZIKA elementini massivdan topmaydi. array_keys() funksiyasi PHP4 versiyasidan boshlab qo'shilgan.

Massiv elementlarining kalitlarini olish uchun ishlatiladigan funksiyalarni ko'rib chiqdik, endi massiv elementlarining qiymatlarini olish uchun ishlatiladigan funksiyalar bilan tanishamiz.

array_values() funksiyasi

Sintaksisi:

```
array_values(massiv);
```

Bu funksiya massivda berilgan barcha element qiymatlarini yangi massivga butun sonlar bilan indekslab yozadi, shuning uchun massiv elementlarining barcha kalitlari yo'qoladi va faqatgina qiymatlari qoladi.

array_unique funksiyasi

Sintaksisi:

```
array_unique(massiv);
```

Bu funksiya massivdan takrorlanuvchi qiymatlarni o'chirib, yangi massiv hosil qiladi. Natijada, bir nechta bir xil qiymat va ularning kalitlari o'rniga bitta qiymat qoladi. Uni kaliti qanday bo'ladi? Yangi massivga kalit bir xil elementlarning bir nechta kalitlari ichidan qanday tanlab olinadi? Massivning barcha elementlari satrlarga akslantiriladi va saralanadi. Shundan so'ng saralovchi har bir qiymatning birinchi kalitini eslab qoladi, boshqa kalitlar esa inkor etiladi.

8.11-misolga qaytamiz. Biz bilamizki, Fizika fani bizning massivda tasodifan ikki marta qatnashib qoldi. Rostdan ham bitta fan ikki marta o'qilmaydi, ("o'qidi, lekin esdan chiqardi" bu hisob

emas), shuning uchun bu kamchilikni to'g'rilaymiz. O'tilgan fanlar ro'yxatidan takrorlanganlarini olib tashlaymiz.

```
<? $dfan = array("Matematika","Fizika","Falsafa","Matematika","Infor-
matika", "Tarix","Fizika", );
print_r(array_unique($dfan)); ?>
Natija:Array ( [0] => Matematika [1] => Fizika [2] => Falsafa [4] =>
Informatika [5] => Tarix )
```

8.4. Massivlarni saralash

Turli xil masalalarni yechishda tez-tez massiv ko'rinishida saqlanayotgan ma'lumotlarni saralash zarur bo'lib turadi. Agar bu masalani Si da yechmoqchi bo'lsak, u holda o'ntagacha kod satrlari yozish kerak bo'ladi, PHP da esa bu bitta oddiy buyruq bilan amalga oshiriladi.

8.2-jadval

Saralash funksiyalari

Funksiya	Saralash	Teskari tartib	<kalit/qiymat>juftligi saqlash
sort	Qiymat	Yo'q	Yo'q
rsort	Qiymat	Ha	Yo'q
asort	Qiymat	Yo'q	Ha
arsort	Qiymat	Ha	Ha
ksort	Qiymat	Yo'q	Ha
krsort	Qiymat	Ha	Ha
usort	Qiymat	?	Yo'q
uasort	Qiymat	?	Ha
uksort	Kalit	?	Ha

sort funksiyasining ishlatilishi

sort() funksiyasi massivni saralab, uning elementi qiymatlarini o'sish tartibi (kichigidan kattasiga qarab) bo'yicha tartiblaydi. Bu funksiya massivning barcha mavjud kalitlarini o'chirib, ularni elementlarning yangi tartibiga mos keluvchi sonli indeksga almashtiradi. Agar funksiya ishni muvaffaqiyatli yakunlasa true ni, aks holda — false ni qaytaradi.

Sintaksisi:

sort (massiv [, bayroq]);

Qo'shimcha parametr bayroq sifatida quyidagi konstantalardan birortasi ishlatilishi mumkin:

SORT_REGULAR – massiv elementlarini oddiy ravishda taqqoslaydi;

SORT_NUMERIC – massiv elementlarini son kabi taqqoslaydi;

SORT_STRING – massiv elementlarini satr kabi taqqoslaydi.

8.12-misol. **sort()** funksiyasining qo'llanilishi:

Bizda ikkita massiv mavjud: fanlar nomi – ularga ajratilgan soat va aksincha, fanlarga ajratilgan soat – ularning nomi. Bu massivlarni o'sish tartibi bo'yicha tartiblaymiz.

```
<? $soatlar= array("informatika" => 56, "fizika" => 34, "matematika" => 36);
sort($soatlar);
// sonlar o'sish tartibi bo'yicha saralanadi, kalitlar tushirib qoldiriladi
print_r($soatlar);
$fanlar= array(56 => "informatika", 34 => "fizika", 36 => "matematika");
sort($fanlar);
// qiymatlar alifbo tartibida saralanadi, kalitlar tushirib qoldiriladi
print_r($fanlar);
?>
Natija:Array ( [0] => 34 [1] => 36 [2] => 56 ) Array ( [0] => fizika [1] => informatika [2] => matematika )
```

asort, rsort, arsort funksiyalari

Bu funksiyalar sintaksisi ham **sort()** funksiyasi kabi bo'lib, vazifalari birmuncha farq qiladi.

Massiv elementlarini kalitlari saqlangan holda saralash uchun **asort()** funksiyasi ishlatiladi.

Sintaksisi:

asort(massiv [, bayroq]);

Agar massivni teskari tartibda, ya'ni katta qiymatdan boshlab kichigiga qarab saralash kerak bo'lsa, u holda **rsort()** funksiyasi ishlatiladi.

Sintaksisi:

rsort(massiv [, bayroq]);

Shu holda, ya'ni teskari tartibda saralangan holda yana kalitlarini ham saqlash kerak bo'lsa, **arsort()** funksiyasidan foydalaniladi.

Sintaksisi:


```
arsort(massiv [, bayroq]);
```

Mos holda bayroq qiymatlari ham sort() funksiyasi kabi bo'ladi: SORT_REGULAR, SORT_NUMERIC, SORT_STRING.

SORT_NUMERIC bayrog'i faqatgina PHP4 da qo'shilgan.

8.13-misol. asort, funksiyasining qo'llanilishi.

```
<? $fanlar= array(56 => "informatika", 34 => "fizika", 36 => "matema-  
tika");
```

```
asort($fanlar); // kalit qiymatlari saqlangan holda, massivni saralaymiz  
print_r($fanlar); ?>
```

```
Natija:Array ( [34] => fizika [56] => informatika [36] =>matematika )
```

8.14-misol. O'quv manbalari katalogini yaratamiz. Har bir manbani muallifi, nomi, chiqarilgan sanasi va qisqacha izohi mavjud. Bu elementlarni massiv elementlari sifatida qarab, ularni teskari tartibda saralaymiz. Buning uchun quyidagi formani yaratamiz va ushbu sahifani biror nom (insert_book.php) bilan saqlaymiz:

8.14a. 8.14-misol uchun forma. insert_book.php fayli

```
<form action=test.php>  
<table border=1>  
  <tr> <td>Nomi </td>  
  <td><input type=text name=nomi size=50> </td> </tr>  
  <tr> <td>Qisqa izohi</td>  
  <td><input type=text name=izohi size=50> </td> </tr>  
  <tr> <td>Muallifi</td>  
  <td><input type=text name=muallifi size=50></td> </tr>  
  <tr> <td>Chiqqan sanasi</td>  
  <td><input type=text name=chiqqan_sana size=50></td> </tr>  
</table>  
<input type=submit value="Jo'natish"> </form>
```

Ikkinchi sahifada insert_book.php faylidagi formada berilgan ma'lumotlarni kalitlari saqlangan holda qiymatlarini kamayish (yoki teskari alifbo) tartibi bo'yicha tartiblaymiz. Buning uchun arsort() funksiyasidan foydalanish qulay. Bu funksiya yordamida berilgan massivning kalitlari saqlangan holda, yangi tartibdagi massivni hosil qilish mumkin. Ikkinchi sahifani test.php deb nomlaymiz va quyidagi skriptni yozamiz, chunki oldingi forma ma'lumotlari test.php fayliga jo'natiladi.

http://localhost:8080/learn_php/insert_book.php - Windows internet Explorer неподдерживает

http://localhost:8080/learn_php/insert_book.php

Файл Правка Вид Избранное Сервис Справка

★ Избранное http://localhost:8080/learn... http://localhost:8080/le... X

Nomi	Informatika va hisoblash texnikasi asoslari
Qisqa izohi	Ushbu kitob informatika yo'nalishi talabalari uchun mo'
Muallifi	Aripov M.M.
Chiqqan sanasi	2001

8.1-rasm. O'quv manbalarini kiritish uchun forma

8.14 b. 8.14-misol uchun berilgan formadagi ma'lumotlarni qayta ishlovchi dastur. test.php fayli

```
<?php
print_r($_GET); echo "<br>";
arsort($_GET);
// kalitlari saqlangan holda massivni teskari tartibda saralaymiz.
print_r($_GET); echo "<br>";
$ordered_names = array_keys($_GET); // yangi massiv yaratamiz
foreach($ordered_names as $key => $val)
echo "$key :$val <br>"; // yangi massiv elementlarini chiqazamiz
?>
```

Massivni kaliti bo'yicha saralash: ksort(), krsort()

Ba'zan massivni kalit qiymatlari bo'yicha saralash zarurati tug'iladi. Misol uchun, agar bizda yuqoridagi misolda berilgani kabi kitoblar haqidagi ma'lumotlar massivi bor bo'lsa, u holda kitoblarni muallifi bo'yicha saralashimiz mumkin. Buning uchun PHP da bir qancha satr kodlarni yozish shart emas, chunki PHP ning bu vazifalarni bajaruvchi funksiyalari mavjud.

ksort() funksiyasi — massiv kalitlarini kalitning mos qiymatlari saqlangan holda, o'sish tartibi (to'g'ri tartib) bo'yicha saralaydi.

ksort() funksiyasi — massiv kalitlarini kalitning mos qiymatlari saqlangan holda, kamayish tartibi (teskari tartib) bo'yicha saralash uchun ishlatiladi. Bu funksiyalarni sintaksisi ham sort() funksiyasi sintaksisiga o'xshash.

8.15-misol. Massivni kaliti bo'yicha saralash

```
<? $fanlar= array(56 => "informatika", 34 => "fizika", 36 => "matematika");  
ksort($fanlar); // kalit qiymatini saqlagan holda, massivni saralaymiz  
print_r($fanlar); ?>  
Natija:Array ( [34] => fizika [36] => matematika [56] => informatika )
```

8.5. Ko'p o'lchovli massivlarni saralash

Massivlarni saralashda, bitta o'zgartirish kiritish yordamida ularning tartibini alifbo yoki raqamli tartibga o'tkazish mumkin. PHP da ikkita son yoki ikkita matnli satrni taqqoslash imkoniyati mavjud, biroq ko'p o'lchovli massivlarning elementlari ham massivdan iborat. PHP ning ikkita massivni taqqoslash imkoniyati mavjud emas, shuning uchun ularni taqqoslashda metodlar yaratish zarur.

Massivlarni foydalanuvchi tomonidan aniqlangan funksiya yordamida saralash: usort(),uksort(),uasort().

Massiv qiymatlarini saralashni oddiy ikkita yo'li (kamayish yoki o'sish tartibi) dan tashqari PHP da foydalanuvchi tomonidan berilgan biror-bir mezon (kriteriya) yordamida ham saralash mumkin. Bu mezonlar funksiya yordamida berilib, funksiya nomi maxsus usort() yoki uksort() saralash funksiyalarida parametr sifatida ko'rsatiladi. Ushbu maxsus funksiyalar foydalanuvchi tomonidan aniqlangan funksiyalar yordamida mos holda usort() massiv elementlari qiymatlarini, uksort() esa — massiv kaliti qiymatlarini saralaydi. Bu ikkala funksiya ham agar saralash ijobiy yakun topsa, true ni, aks holda false ni qaytaradi.

Sintaksisi:

usort(massiv, saralovchi_funksiya)

uksort(massiv, saralovchi_funksiya)

Albatta, massivlarni foydalanuvchi tomonidan aniqlangan funksiya yordamida saralab bo'lmaydi. Bu funksiya massiv elementlarini taqqoslash imkonini beruvchi aniqlangan kriteriyalarni qoniqtirishi kerak. Saralovchi funksiya qanday yaratilgan bo'lishi kerak? degan

savol tug'iladi. Birinchidan, ikkita parametrga ega bo'lishi kerak. Ularga interpretator usort() funksiyasi uchun element qiymati yoki uksort() funksiyasi uchun massiv kaliti juftligini berishi mumkin. Ikkinchidan, saralovchi funksiya quyidagilardan birortasini qaytarishi kerak:

noldan kichik butun sonni, agar birinchi parametr ikkinchisidan kichik bo'lsa;

nolga teng sonni, agar ikkala parametr teng bo'lsa;

noldan katta sonni, agar birinchi parametr ikkinchisidan katta bo'lsa.

Boshqa saralash funksiyalari kabi usort() funksiyasining ham o'xshashi mavjud, kalit qiymati o'zgarmaydigan uasort() funksiyasi.

Quyida avval ko'rilgan massiv berilgan. Bu massivda talabalar haqidagi ma'lumotlar, ularning kodi, familiyasi, ismi, stipendiyasi, kursi, manzili, tug'ilgan sanasi, universiteti berilgan.

8.16-misol. Ko'p o'lchovli massivni foydalanuvchi tomonidan aniqlangan funksiya yordamida saralash

```
<? $talabalar=array(
array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992", "TDPU"),
array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990",
"TDPU"),
array("6","Salimov", "Void", 150, 4, "Namangan", "7/06/1989", "Nam
DU"),
array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU")); ?>
```

Bu ko'p o'lchovli massivni saralasak, uning qiymatlari qanday tartibda saralanadi? Bizga ma'lumki, massiv tarkibini ikkita tartibda saralash mumkin. Satrli ma'lumotlar (familiyasi, ismi, manzili, universiteti)ni alifbo tartibida, sonlilari (kodi, stipendiyasi, kursi)ni raqamli tartibda saralanadi. Buning uchun usort() funksiyasi va elementlarni qanday taqqoslash kerakligini ko'rsatuvchi foydalanuvchi tomonidan aniqlangan funksiya ishlatiladi.

Quyidagi kod bu massivning uchinchi ustuni (ismi) qiymatlarini alifbo tartibida saralaydi.

```
function ftaf($x, $y)
{ if ( $x[2] == $y[2] ) return 0;
else if ( $x[2] < $y[2] ) return -1;
else return 1; }
usort($talabalar, ftaf);
```


Foydalanuvchi tomonidan aniqlangan funksiyalar qanday yaratilishini avvalgi mavzularda ko'rib o'tganmiz. Bu misolda foydalanuvchi tomonidan aniqlangan funksiya — `ftaf()`. `ftaf()` funksiyasi parametr sifatida ikkita qiymatni qabul qiladi: `x` va `y`. `ftaf()` funksiyasining vazifasi ushbu ikkita qiymatni qabul qilib, ularning tartibini aniqlash.

Qaralayotgan misolda `x` va `y` parametrlari har biri alohida talabani ismini bildiruvchi asosiy massiv ichidagi massivlar. `x` massivining `name(ismi)` elementiga murojaat etish uchun `x[2]` deb yozish kerak. Chunki massiv noldan boshlab indekslanganligi uchun `name(ismi)` — ikkinchi element. Massivdagi `name(ismi)` elementini taqqoslash uchun funksiyaga uzatiladigan `$x[2]` va `$y[2]` o'zgaruvchilari ishlatiladi.

Funksiya o'z ishini tugatgandan so'ng javob sifatida qiymat qaytaradi. Funksiya qiymatlarini qaytarish uchun `return` kalit so'zi ishlatiladi. Masalan, `return 1`; satri funksiyaga murojaat etganda, `1` kodini qaytaradi. `ftaf()` funksiyasi `x` va `y` ni taqqoslash uchun `usort()` funksiyasi bilan ishlatiladi. Agar `x` ning qiymati `y` ning qiymatiga teng bo'lsa, `0` ni, kichik bo'lsa, manfiy sonni, katta bo'lsa, musbat sonni qaytaradi. `x` va `y` ning qiymatiga bog'liq ravishda funksiya `0`, `1` yoki `-1` ni qaytaradi.

Kod satrining oxirida `usort()` funksiyasiga saralash kerak bo'lgan massiv(`$talabalar`) va foydalanuvchi funksiyasi(`ftaf()`) bilan murojaat etiladi.

Agar massivni boshqa tartibda saralash lozim bo'lsa, boshqa funksiya yaratiladi. Masalan, talabalarining kurslari bo'yicha saralash lozim bo'lsa, beshinchi ustun qoraladi va saralash uchun quyidagi funksiya yaratiladi.

```
function ftaf($x, $y)
{ if ( $x[4] == $y[4] ) return 0;
  else if ( $x[4] < $y[4] ) return -1;
  else return 1; }
usort($talabalar, ftaf);
```

`usort($talabalar, ftaf);` funksiyasiga murojaat etgandan, so'ng massiv kurslarini o'sish tartibi bo'yicha tartiblaydi.

`usort()` funksiyasidagi `u` belgisi "user"("foydalanuvchi") degan ma'noni anglatadi, shuning uchun bu funksiya foydalanuvchi funksiyasini talab etadi.

uksort() va uasort() funksiyalari ksort() va asort() larga o'xshash, faqatgina bular ham foydalanuvchi funksiyasini talab etadi.

asort()ga o'xshab uasort() funksiyasi ham assotsiativ massivlarning qiymatlarini saralaydi. Qiymatlar oddiy sonlar va satrlardan iborat bo'lsa, asort() funksiyasi ishlatiladi. Agar qiymatlar massiv singari murakkab obyektlardan tashkil topgan bo'lsa, u holda taqqoslash uchun funksiyani aniqlash va uasort() funksiyasini ishlatish kerak.

sort(), asort() va ksort() funksiyalarining ularga mos nomida "g" belgisi ishtirok etgan teskari tartibda saralovchi o'xshashi mavjud. Saralash uchun foydalanuvchi tomonidan aniqlangan funksiyalarining teskari versiyasi yo'q, biroq ko'p o'lchovli massivlarni teskari tartibda saralash mumkin. Modomiki, taqqoslash funksiyalari dasturchilar tomonidan yaratilgan ekan, demak qarama-qarshi qiymatni qaytaruvchi taqqoslash funksiyalarini ham yaratish mumkin. Saralash teskari tartibda bajarilishi uchun, agar x ning qiymati u dan kichik bo'lsa 1 va agar x ning qiymati u dan katta bo'lsa -1 bo'lishi kerak.

8.17-misol. Ko'p o'lchovli massivni foydalanuvchi tomonidan aniqlangan funksiya yordamida teskari tartibda saralash

```
<? $talabalar=array(
array("1","Islomov", "Ikrom", 1899, 1, "Andijon", "3/12/1992", "TDPU"),
array("3","Fozilov", "Fozil", 1870, 3, "Qashqadaryo", "1/12/1990",
"TDPU"),
array("6","Salimov", "Vohid", 1880, 4, "Namangan", "7/06/1989", "Nam-
DU"),
array("10","Karimov", "Botir", 1885, 2, "Jizzax", "8/12/1991", "TDPU") );
function aksftaf($x, $y)
```

```
{ if ( $x[4] == $y[4] ) return 0; else if ( $x[4] < $y[4] ) return 1;
else return -1; }
usort($talabalar, aksftaf);
for ($i=0; $i<4; $i++) {echo $talabalar[$i][1]. " ".$talabalar[$i][4]. "- kurs<br>";
} ?>
```

Natija: Salimov 4-kurs
Fozilov 3-kurs
Karimov 2-kurs
Islomov 1-kurs

8.6. Massivning har bir elementi uchun ixtiyoriy funksiyani qo'llash

Ba'zan bir xil amalni massivning barcha elementlari uchun qo'llash yoki ularni bir xil shaklda o'zgartirish talab etiladi. Buning uchun `array_walk()` funksiyasidan foydalaniladi.

Sintaksisi:

```
int array_walk(array massiv, string funksiya [, mixed parametr])
```

Ko'rib turganingizdek, bu funksiya uchta parametr qabul qiladi. Birinchisi, massiv — qayta ishlanishi lozim bo'lgan massiv. Ikkinchisi — funksiya — massivning har bir elementini qabul qiluvchi foydalanuvchi tomonidan aniqlangan funksiya. Uchinchi parametr — zarur bo'lmagan parametr bo'lib, u ishlatilganda foydalanuvchi tomonidan aniqlangan funksiyaga parametr sifatida beriladi. Bu funksiya amallar muvaffaqiyatli yakunlanganda `true` ni, aks holda `false` ni qaytaradi.

Qoidaga ko'ra, foydalanuvchi tomonidan aniqlangan funksiyaning ikkita parametri mavjud bo'lib, bu ikki parametrga navbat bilan massivning har bir elementining qiymat va kalitlari beriladi.

Agar `array_walk()` funksiyasiga murojaat etilganda, uchinchi parametr ko'rsatilsa, u holda bu parametr foydalanuvchi tomonidan aniqlangan funksiyaning uchinchi parametri sifatida qaraladi. Agar foydalanuvchi tomonidan aniqlangan funksiya unga berilgandan ko'ra ko'proq parametrlarni talab qilsa, u holda `array_walk()` ga har gal murojaat etilganda ogohlantirish haqida xabar chiqaziladi.

Agar massiv qiymatlarining nusxasi bilan emas, balki asli bilan ishlashga to'g'ri kelinsa, parametрни funksiyaga jo'natma (ssilka) sifatida uzatish kerak bo'ladi. Biroq, massivning elementlarini o'chirish yoki qo'shish va massiv tarkibini o'zgartiruvchi harakatlarni amalga oshirish mumkin emas, chunki bu holda `array_walk()` ishining natijasi noaniq bo'ladi.

8.18-misol. Massivning barcha elementlari uchun funksiyani qo'llash

```
<? $fanlar= array(56 => "informatika", 34 => "fizika", 36 => "matematika"); // massiv elementlariga qo'llash uchun funksiya yaratamiz
function fan_soat($val,$key,$data)
{ echo "\"$val\" $data $key soat ajratilgan.<br>"; }
```

```

/* $fanlar massivning barcha elementlari uchun fan_soat funksiyasining
qo'llanilishi*/
array_walk($fanlar,"fan_soat","faniga"); ?>
Natija: "informatika" faniga 56 soat ajratilgan.
"fizika" faniga 34 soat ajratilgan.
"matematika" faniga 36 soat ajratilgan.

```

Massiv elementi qiymati o'zgartirilmaganligini ko'rish mumkin. Ularni o'zgartirish uchun fan_soat funksiyasidagi \$val o'zgaruvchisiga jo'natma (ssilka) orqali qiymatni uzatish kerak.

8.7. Qism massivni belgilash array_slice() funksiyasi

Massiv bu – elementlar to'plami. Ushbu to'plamdan birorta to'plamosti ajratib olish uchun PHP da array_slice() funksiyasidan foydalaniladi. Ajratib olingan massiv qism massiv deyiladi.

Sintaksisi:

```
array_slice(massiv, element_nomeri[, uzunligi])
```

Bu funksiya element_nomeri parametrda berilgan nomerdagi massiv elementidan boshlab, uzunligi uzunligi parametriga teng bo'lgan uzunlikdagi qism massivni berilgan massivdan ajratib oladi. Musbat element_nomeri element nomerining tartibini mos ravishda massivning boshidan boshlab ko'rsatadi, manfiy element_nomeri esa massivning oxiridan boshlab hisoblaydi.

Agar array_slice() ni ishlatishda uzunligi parametri berilsa, u holda belgilangan qism massiv elementlari soni bu parametrdan berilgan songa teng bo'ladi, uzunligi parametri manfiy son sifatida ham ko'rsatilishi mumkin. Bu holda interpretator massiv oxiridan boshlab, uzunligi parametri moduliga teng sondagi elementni o'chiradi.

8.19-misol. array_slice() funksiyasining qo'llanilishi.

```

<? $arr = array(1,2,3,4,5);
$sub_arr = array_slice($arr, 2, 2);
// massiv 3, 4 elementlaridan tashkil topadi
$sub = array_slice($arr,-3, 2);
// massiv ham 3, 4 elementlaridan tashkil topadi
$sub1 = array_slice($arr,0, -1);
// massiv 1,2,3, 4 elementlaridan tashkil topadi
$sub2 = array_slice($arr,-4, -2);
// massiv 2,3 elementlaridan tashkil topadi
?>

```


array_chunk funksiyasi

array_chunk() funksiyasi array_slice() funksiyasiga o'xshash bo'lib, bu funksiya massivni berilgan uzunligiga qarab bir nechta qism massivga ajratadi.

Sintaksisi:

array_chunk(massiv, hajmi [, kalitlarni_saqlash])

array_chunk() ishining natijasi sifatida elementlari qism massivlardan tashkil topgan ko'p o'lchovli massivni chiqazadi. Agar kalitlarni_saqlash parametri true sifatida berilsa, u holda berilgan massivni ajratishda kalitlari saqlanadi. Aks holda element kalitlari noldan boshlanuvchi sonli indekslarga almashadi.

8.20-misol. array_chunk() funksiyasining qo'llanilishi:

Talabalar familiyasidan tashkil topgan massiv berilgan. Har bir guruhga uchtdan talaba to'g'ri keladi. Talabalarni guruhlarga uchtdan etib taqsimlash kerak.

```
<? $talabalar = array("Ismoilov", "Fozilov", "Samadova", "Karimova",  
"Vohidov");
```

```
$ajrat = array_chunk($talabalar, 3);
```

```
// massivni uchta elementli qism massivga ajratamiz
```

```
foreach ($ajrat as $k => $guruh) { echo "$k - nomerli guruhga: <ul>";
```

```
foreach ($guruh as $talaba) echo "<li>$talaba";
```

```
echo "</ul>"; } ?>
```

Natija:

0 — nomerli guruhga:

Ismoilov

Fozilov

Samadova

1 — nomerli guruhga:

Karimova

Vohidov

8.8. Massiv elementlari yig'indisini hisoblash

Massiv elementlari qiymatlarining yig'indisini hisoblash uchun array_sum() funksiyasidan foydalaniladi. Bu funksiyaning parametri sifatida esa element qiymatlari yig'indisi hisoblanishi kerak bo'lgan massiv ko'rsatiladi.

Sintaksisi:

```
mixed array_sum(array massiv);
```

Bu funksiyaning ishlatilishiga birmuncha qiyinroq misol ko'ramiz. Bu misolda avvalroq ko'rib o'tgan array_slice() funksiyasini ham ishlatamiz.

8.21-misol. Natural sonlar massivi berilgan. Bu massivdan shunday son topish kerakki, shu elementning o'ng tomondagi elementlari yig'indisi, chap tomondagi elementlari yig'indisiga teng bo'lishi kerak.

```
<? $arr = array(2,1,3,4,5,6,4);  
/* Sikl ichida $k o'zgaruvchisi massivning joriy kalitidan, $v o'zgaruvchisi  
esa massivning joriy qiymatidan tashkil topgan*/  
foreach ($arr as $k => $v){ $p = $k + 1;  
/*array_slice (massiv, element_nomer[, uzunligi]) array_slice funksiyasi  
uzunligi uzunligi parametriga teng*/  
// element_nomeri parametrida berilgan nomerli elementdan boshlangan  
//qism massivni belgilaydi  
$out_next = array_slice($arr,$p);  
$out_prev = array_slice($arr,0,$k);  
/* array_sum funksiyasi berilgan massiv elementlari yig'indisini hisoblay-  
di*/  
$next_sum = array_sum($out_next);  
$prev_sum = array_sum($out_prev);  
if ($next_sum===$prev_sum) echo "qiymat:$v"; } ?>  
Natija:qiymat:5
```

8.9. Massiv elementlarining joylashgan tartibini o'zgartirish

Ba'zi masalalarda massiv elementlari tartibini turli usullar bilan o'zgartirish talab etiladi. Shuffle() funksiyasi massiv elementlarini tasodifiy tartibda joylashtiradi. array_reverse() funksiyasi massiv elementlari teskari tartibda joylashgan massiv nusxasini qaytaradi.

shuffle() funksiyasining ishlatilishi

Talabalar portalining asosiy sahifasiga a'lochi (eng ko'p stipendiya oluvchi) talabalarining rasmlarini joylashtiraylik. Bunday talabalar ko'pchilikni tashkil etishi mumkin, lekin ularning hammasini emas, tasodifiy to'rtta a'lochi talabaning rasmi asosiy sahifada chiqib tursin. Chunki, har doim bir xil rasmni chiqib turishi portal foydalanuvchilarini zeriktirib qo'yishi mumkin. Agar barcha rasmlar massivda saqlansa, bu masalani hal etish oson.

Dastur 8.22-listingda berilgan. Ushbu dastur to'rtta ixtiyoriy tanlangan rasmlarni ekranga chiqazadi. Avval massiv elementlari ixtiyoriy tarzda tartiblanadi, so'ng ulardan birinchi to'rttasi tasvirlanadi.

8.22-listing. forn_page.php – talabalar portalini asosiy sahifasini takomillashtirish uchun PHP ning ishlatilishi.

```
<? $pictures = array("upload/1.jpg", "upload/3.jpg", "upload/5.jpg",  
"upload/10.jpg", "upload/16.jpg");  
shuffle($pictures);  
?>  
<html>  
<head> <title>Talabalar portali</title> </head>  
<body> <center>  
<h1> Talabalar portali</h1>  
<table width = 100%>  
<tr>  
<? for ($i = 0; $i < 4; $i++)  
{ echo "<td align = center><img src=\"";  
echo $pictures [ $i] ;  
echo "\" width = 200 height = 250></td>\" ; } ?>  
</tr> </table> </center>  
</body>
```

array_reverse() funksiyasining ishlatilishi

array_reverse() funksiyasi parametr sifatida massiv qabul qiladi va shu massiv elementlarini teskari tartibda joylashtirib yangi massivni hosil qiladi.

8.23-misol. 10 dan 1 gacha teskari tartibda nomerlangan elementlardan tashkil topgan massivni yaratamiz.

range() funksiyasi ketma-ket o'sib boruvchi sonlar massivini yaratadi, sonlarni kamayish tartibida saralash uchun rsort() funksiyasi ishlatiladi. Yoki for siklini ishlatib ham, bittadan element qo'shish yordamida massivni hosil qilish mumkin:

```
<? $numbers = array ();  
for($i=10; $i>0; $i--)  
array_push( $numbers,$i );  
print_r($numbers); ?>  
Natija:Array ( [0] => 10 [1] => 9 [2] => 8 [3] => 7 [4] => 6 [5] => 5 [6]  
=> 4 [7] => 3 [8] => 2 [9] => 1 )
```

for() sikli ushbu misolda ko'rsatilganidek, boshqaruvchi o'zgaruvchi yordamida kamayish tartibini bajaradi. Boshlang'ich qiymatga eng katta qiymat o'rnatiladi, so'ng har bir sikl oxirida hisoblagich qiymatni -- amali yordamida bittaga kamaytiradi. Avval bo'sh massiv yaratiladi, so'ng siklning har bir elementiga array_push() funksiyasini qo'llab, massivning oxiriga yangi element qo'shib boriladi va natijada yangi massiv hosil bo'ladi. array_push() funksiyasiga teskari funksiya bu — array_pop(). Bu funksiya elementni o'chirib, massiv oxiridan bitta elementni qaytaradi.

Xuddi shunday, range() funksiyasi yaratgan massiv elementlari-ning tartibini o'zgartirish uchun array_reverse() funksiyasidan foydalaniladi.

```
<? $numbers = range(1,10);
$numbers = array_reverse($numbers);
print_r($numbers); ?>
Natija:Array ( [0] => 10 [1] => 9 [2] => 8 [3] => 7 [4] => 6 [5] => 5
[6] => 4 [7] => 3 [8] => 2 [9] => 1 )
```

E'tibor bering, array_reverse() funksiyasi o'zgartirilgan massivning nusxasini qaytaradi. Yangi nusxa esa eskisini ustiga saqlanadi.

8.10. Ma'lumotlarni fayldan massivga yuklab olish

8.2-masala. Avvalgi mavzularimizda talabalar haqidagi ma'lumotlarni faylda saqlab qo'ygandik. Faylning har bir satri quyidagicha ko'rinishda edi:

1 Islomov Ikrom 150 1 Andijon 3/12/1992 TDPU

Talabalar ma'lumotlarini qayta ishlash uchun uni qayta massivga yuklab olish mumkin. 8.24-listingda berilgan sahifa joriy faylning tarkibini chiqazadi.

8.24-listing. view.php. Fayl tarkibini chiqazish uchun PHPning ishlatilishi

```
<? $talaba= file("student.txt") ;
// massivdagi talabalar ma'lumotlari sonini hisoblash
$number_of_talaba = count($talaba);
if ($number_of_talaba==0)
{ echo "<p><strong>Talaba ma'lumotlari kiritilmagan.
Yana urinib ko'ring.</strong></p>"; }
for($i=0;$i<$number_of_talaba;$i++) {echo $talaba[$i] . "<br>"; } ?>
```


Natija:

1 Islomov Ikrom 150 1 Andijon 3/12/1992 TDPU

3 Fozilov Fozil 200 3 Qashqadaryo 1/12/1990 TDPU

6 Salimov Vohid 150 4 Namangan 7/06/1989 NamDU

10 Karimov Botir 100 2 Jizzax 8/12/1991 TDPU

Bu sahifa avvalgi ma'ruzalarda yaratilgan 9.14-listingda ko'rsatilgan natijani chiqazadi. Bitta farqi faylning barchasini massivga yuklab olish uchun file() funksiyasi ishlatiladi. Faylning har bir satri massivning alohida elementi bo'ladi. Bu sahifada ham massiv elementlari sonini aniqlash uchun count() funksiyasi ishlatiladi.

Bundan tashqari talabalar ma'lumotlari satri bo'limlarini keyinchalik qayta ishlash uchun alohida massivning elementiga yuklab olish mumkin. Bu vazifalarni bajarish uchun 8.25-listingda dastur kodi berilgan.

8.25-listing. vieworders.php fayli.

```
<html> <head>
<title>Talabalar portali-talabalar haqidagi ma'lumotlar</title></head>
<body>
<h1>Talabalar portali</h1> <h2>Talabalar haqidagi ma'lumotlar</h2>
<? //faylni hammasini o'qib olamiz
//Har bir talaba ma'lumotlari massiv elementlari bo'ladi
$talaba= file("student.txt") ;
// massivdagi talabalar ma'lumotlarini sonini hisoblash
$number_of_talaba = count($talaba);
if ($number_of_talaba==0)
{ echo "<p><strong>Talaba ma'lumotlari kiritilmagan.
Yana urinib ko'ring.</strong></p>"; }
echo "<table border = 0 cellpadding = 3>";
echo "<tr><td bgcolor = \"#CCCCCC\" align = center>Talaba id</td>
<td bgcolor = \"#CCCCCC\" align = center>Familiyasi</td>
<td bgcolor = \"#CCCCCC\" align = center>Ismi</td>
<td bgcolor = \"#CCCCCC\" align = center>Stipendiyasi</td>
<td bgcolor = \"#CCCCCC\" align = center>Kursi</td>
<td bgcolor = \"#CCCCCC\" align = center>Manzili</td>
<td bgcolor = \"#CCCCCC\" align = center>Tug'ilgan sanasi</td>
<td bgcolor = \"#CCCCCC\" align = center>Universiteti</td>
<tr>";
```

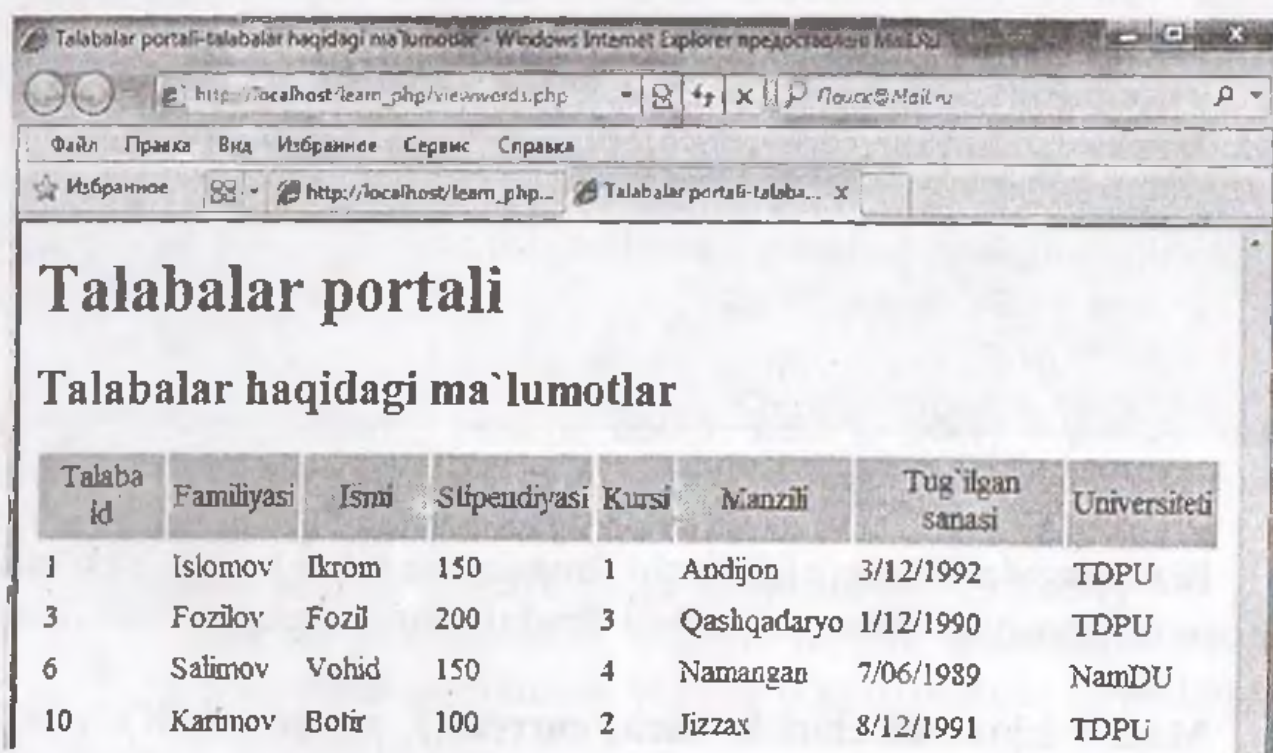


```

for($i=0;$i<$number_of_talaba;$i++)
{ //har bir satrni ajratish
$result = explode( "\t", $talaba[$i] );
//string tipidagi ma'lumotlarni integer tipiga o'tkazish
$result[0] = intval( $result[0]); $result[3] = intval( $result[3]);
$result[4] = intval( $result[4]);
//talabaning har bir ma'lumotini chiqazish
echo "<tr><td>$result[0]</td> <td>$result[1]</td>
      <td>$result[2]</td> <td>$result[3]</td>
      <td>$result[4]</td> <td>$result[5]</td>
      <td>$result[6]</td> <td>$result[7]</td>
      </tr>"; } echo "</table>"; ?>

```

8.25-listingda berilgan dastur kodi faylning barchasini massivga yuklab oladi. 8.24-listingdan farqli ravishda bu listingda explode() funksiyasi har bir satrni bo'limlarga (talabaning har bir ma'lumotini) keyinchalik ular ustida ishlash uchun ajratadi. Bu sahifaning natijasi 8.2-rasmda ko'rsatilgan.



Talaba id	Familiyasi	Ismi	Stipendiyasi	Kursi	Manzili	Tug'ilgan sanasi	Universiteti
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	TDPU
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid	150	4	Namangan	7/06/1989	NamDU
10	Karimov	Botir	100	2	Jizzax	8/12/1991	TDPU

8.2-rasm. Talaba ma'lumotlarini fayldan massivga yuklab olib tasvirlash

Ushbu misolning dastur kodini biroz boshqacharoq qilib ham yozishimiz mumkin:

8.26-listing. vieworders2.php fayli.

```
<html>
<head> <title>Talabalar portali-talabalar haqidagi ma'lumotlar</title>
</head>
<body> <h1>Talabalar portali</h1>
<h2>Talabalar haqidagi ma'lumotlar</h2>
<table border = 0 cellpadding = 3>    <tr>
    <td bgcolor = "#CCCCCC" align = center>Talaba id</td>
    <td bgcolor = "#CCCCCC" align = center>Familiyasi</td>
    <td bgcolor = "#CCCCCC" align = center>Ismi</td>
    <td bgcolor = "#CCCCCC" align = center>Stipendiyasi</td>
    <td bgcolor = "#CCCCCC" align = center>Kursi</td>
    <td bgcolor = "#CCCCCC" align = center>Manzili</td>
    <td bgcolor = "#CCCCCC" align = center>Tug'ilgan sanasi</td>
```

```
<td bgcolor = "#CCCCCC" align = center>Universiteti</td>
</tr>
<? $h = fopen("student.txt","r");
while (!feof($h)) {
    $talaba = fgetcsv($h,filesize("student.txt"),"\t");
    echo "<tr>";
    while (list( $key, $value) = @each($talaba) )
    { print "<td>". $value. "</td>"; }
    echo "</tr>"; } ?>
</table> </body> </html>
```

8.11. Massivlar ustida boshqa amallarni bajarish

Biz massivlarni qayta ishlovchi funksiyalar bilan tanishib chiqdik. Boshqa massivlar bilan ishlashda foydali funksiyalarni ham ko'rib o'tamiz.

Massiv ichida ko'chirish: each, current(), reset(), end(), next(), pos() va prev()

Avval aytib o'tganimizdek, har bir massiv ichida massivning joriy elementini ko'rsatib turuvchi ichki ko'rsatkichi bo'ladi. Biz ushbu ko'rsatkichni each() funksiyani ishlatganimizda ko'rib o'tdik, biroq uni bevosita ishlatish va qo'llash mumkin.

Yangi massivni yaratishda joriy ko'rsatkich massivning birinchi elementini ko'rsatish uchun o'rnatiladi. `current($array_name)` funksiyasiga murojaat etganda, birinchi elementni chiqazadi.

`next()` yoki `each()` funksiyasiga murojaat etganda, ko'rsatkich bitta element oldinga ko'chadi. `each($array_name)` funksiyasiga murojaat etganda, ko'rsatkich ko'chmasdan avval, joriy elementni qaytaradi. `Next()` funksiyasi esa biroz boshqacha, `next($array_name)` funksiyasiga murojaat etganda, ko'rsatkichni ko'chirib, so'ng yangi joriy elementni qaytaradi.

`reset()` funksiyasi ko'rsatkichni massivning birinchi elementiga qaytaradi. Shunga o'xshash, `end($array_name)` funksiyasiga murojaat etganda, ko'rsatkich massivning oxiriga ko'chadi. `reset()` va `end()` funksiyalari mos ravishda massivning birinchi va oxirgi elementlarini qaytaradi.

Massiv ichida ko'chirishni teskari yo'nalishda bajarish uchun `end()` va `prev()` funksiyalaridan foydalaniladi. `prev()` funksiyasi `next()` funksiyasiga nisbatan teskari funksiya. U joriy ko'rsatkichni bitta element orqaga ko'chiradi, so'ng yangi joriy elementni qaytaradi.

Masalan, quyidagi kod massiv elementlarini teskari tartibda chiqazadi:

8.27-misol.

```
<? // $array massivi quyidagicha e'lon qilingan
$array = array (1, 2, 3) ;
$value = end ($array) ;
while ($value)
{ echo "$value<br>"; $value = prev($array); } ?>
```

`each()`, `current()`, `reset()`, `end()`, `next()`, `pos()` i `prev()` funksiyalari ishlatilganda massivda ixtiyoriy tartibda o'zgartirish uchun kod yaratish mumkin.

Massivni skalyar o'zgaruvchiga almashtirish: `extract()`

Kalit-qiymat juftliklaridan tashkil topgan assotsiativ massivni `extract()` funksiyasi yordamida skalyar o'zgaruvchilar to'plamiga almashtirish mumkin.

Sintaksisi:

```
extract (array var_array [, int extraot_type] [, string prefix] );
```

`extract()` funksiyasi skalyar o'zgaruvchilarni yaratishda ularning nomlarini massiv kalitlari nomi bilan nomlaydi. O'zgaruvchilar massivdagi mos qiymatlarga tenglanadilar.

8.28-misol.

```
<? $array = array( "key1" => "value1", "key2" => "value2", "key3" =>
"value3" );
extract($array);
echo "$key1 $key2 $key3"; ?>
```

Natija: value1 value2 value3

Massiv kalitlari key1, key2 va key3 dan iborat uchta elementdan tashkil topgan. extract() funksiyasi ishlatilganda uchta \$key1, \$key2 va \$key3 skalyar o'zgaruvchilar yaratildi. Natijada ko'rib turganinigizdek, \$key1, \$key2 va \$key3 o'zgaruvchilarning qiymatlari mos ravishda "value1", "value2" "value3" lar bo'ladi.

extract() funksiyasi ikkita zarur bo'lmagan parametrga ega: extract_type va prefix. extract_type parametri extract() funksiyasiga kalit nomi bilan bir xil o'zgaruvchilar mavjud bo'lganda yuzaga keladigan konfliktlarni qayta ishlash usulini beradi. Odatga ko'ra mavjud o'zgaruvchining qiymati yangisiga almashtiriladi. extract_type parametrining to'rtta mumkin bo'lgan qiymatlari jadvalda keltirilgan.

8.3-jadval

extract() funksiyasining extract_type parametrini mumkin bo'lgan qiymatlari

Qiymati	Vazifasi
EXTR_OVERWRITE	Konflikt yuzaga kelganda mavjud o'zgaruvchini qayta yozadi.
EXTR_SKIP	Konflikt yuzaga kelganda elementni tashlab o'tadi (qoldirib ketadi).
EXTR_PREFIX_SAME	Konflikt yuzaga kelganda \$prefix_key o'zgaruvchisini yaratadi. Bu holda funksiyada prefix parametri ko'rsatilishi kerak.
EXTR_PREFIX_ALL	Barcha o'zgaruvchilarga prefix old qo'shimchasini beradi. Bu holda funksiyaga bu parametr berilishi kerak.

Odatga ko'ra ko'p ishlatiladigan qiymatlar: (EXTR_OVERWRITE) va (EXTR_PREFIX_ALL). Qolgan ikkita qiymat avvaldan konflikt yuzaga keladigan joyi ma'lum bo'lsa va kalit tushirib qoldirilgan va old qo'shimcha ishlatilishi talab etilayotgan hollarda ishlatiladi.

Quyida EXTR_PREFIX_ALL ning parametr sifatida ishlatilishiga misol ko'rsatilgan. Ko'rib turganingizdek, yaratilgan o'zgaruvchi old_qo'shimcha_kalit_nomi nomini oladi.

8.29-misol.

```
<? $array = array( "key1" => "value1", "key2" => "value2", "key3" => "value3");
extract($array, EXTR_PREFIX_ALL, "old_qushimcha");
echo "Sold_qushimcha_key1 Sold_qushimcha_key2 Sold_qushimcha_key3"; ?>
Natija: value1 value2 value3
```

8.3-masala. Formani dinamik yaratish

Tashrif buyuruvchilarga yuqori darajadagi dizaynga ega saytlar ro'yxatini tavsiya etish lozim bo'lsin. Buning uchun saytlar ro'yxatidan tashkil topgan ochiluvchi ro'yxatni yaratamiz. Bu ro'yxatdagi sayt nomlari massivning elementlari bo'ladi.

8.30-listing. Ochiluvchi ro'yxatni dinamik yaratish (qurish)

```
<? if (isset($site)) : header("Location: http://$site"); exit;
else : ?>
<html> <head> <title>Formani dinamik yaratish</title> </head>
<body bgcolor="#ffffff" text="#000000" link="#cbda74"
link="#808040" alink="#808040">
<? $favsites = array ("www.tdpu.uz", "www.pedagog.uz", "www.ziyo-
net.uz", "www.nuu.uz", "www.edu.uz");
// forma yaratamiz ?>
<form action = "learn.php" method="post">
<select name="site"> <option value = "">Saytni tanlang:
<? $x = 0;
while ( $x < sizeof ($favsites) ) :
print "<option value='<? $favsites[$x]>$favsites[$x]>";
$x++; endwhile; ?>
</select> <input type="submit" value="go!">
</form> </body>
</html> <? endif;?>
```

Avval \$site o'zgaruvchisining qiymati bor-yo'qligi tekshiriladi. Agar tekshirish ijobiy natijani bersa, PHP ning maxsus header() funksiyasi parametr sifatida berilgan \$site o'zgaruvchisining qiymati bilan chaqiriladi. header() funksiyasiga murojaat etilganda, funk-

siya brauzerni \$site o'zgaruvchisida ko'rsatilgan URL manziliga yo'naltiradi. Agar \$site o'zgaruvchisi qiymati berilmagan bo'lsa, forma brauzerga chiqaziladi.

Ochiluvchi ro'yxat siklda quriladi, iteratsiyalar soni \$favsites massivi elementlari soniga bog'liq bo'ladi.

header() funksiyasiga sahifaning yuqori qismida murojaat etish kerak. Unga PHP ssenariysining ixtiyoriy nuqtasida murojaat etish mumkin emas. O'z vaqtida murojaat etilmagan header() funksiyasi muammo tug'dirishi mumkin.

Natija



8.3-rasm. Ochiluvchi ro'yxatni dinamik yaratish (qurish). Tanlangandan so'ng mos saytga o'tiladi

Formani dinamik yaratish ayniqsa katta hajmdagi ma'lumotlarni qayta ishlashda, ularni ixtiyoriy vaqtda o'zgartirishga qulay.

IX bob. SATRLAR VA DOIMIY IFODALAR

Bu bobda satrlar bilan ishlash, turli xil amaliy masalalarni yechishda foydali funksiyalarni o'rganish to'liqroq muhokama qilinadi. Satrlarni chiqazishning turli usullari, satrlarni bo'lish va birlashtirish (explode, implode funksiyalari), satr uzunligini aniqlash (strlen), qism satrlarni belgilash (strpos, substr) ko'rib chiqiladi.

9.1. Satrlar bilan ishlash

Satrlar ma'lumotlar tipi va bunday tipdagi o'zgaruvchini qanday yaratish haqida oldingi ma'ruzalarda aytib o'tdik. Oldingi ma'ruzalarda aytib o'tganimizdek, satrli o'zgaruvchilar uchta usul: bittalik qo'shtirnoq yordamida, ikkitalik qo'shtirnoq yordamida va heredoc-sintaksisi yordamida beriladi.

Doimiy ifodalar — shablon bo'yicha qidirishning barcha zamonaviy texnologiyalarning asosi hisoblanadi. Doimiy ifodalar qidirilayotgan matnda yozilgan oddiy va xizmatchi belgilarni ketma-ketligini saqlagan holda tasvirlaydi. Ba'zan doimiy ifodalar oddiy va tushunarli (masalan, dog so'zi) bo'ladi, lekin ko'pincha ularning sintaksisida alohida mazmunga ega bo'lgan xizmatchi belgilar qatnashadi, masalan `<(?)>.*<\/.?>`.

Doimiy ifodalar bilan ishlovchi funksiyalardan tashqari PHP ning satrlar ustida mantiqiy amallarni bajaruvchi 70 dan ortiq funksiyalari mavjud.

Katta hajmdagi ma'lumotlarni qayta ishlashda, doimiy ifodalar bilan ishlovchi funksiyalarning ishlash tezligi dastur bajarilishida pasayadi. Bu funksiyalarni doimiy ifodalar haqiqatda zarur bo'lgan, nisbatan murakkabroq satrlarni faqatgina qayta ishlashda qo'llash kerak bo'ladi. Agar matnni tahlil etish nisbatan oddiy qoidaga ko'ra bajarilsa, qayta ishlashni sezilarli darajada tezlashtiradigan PHP ning standart funksiyalaridan foydalanish mumkin.

9.1-masala. Foydalanuvchi ma'lumotlarini berilgan elektron pochta manziliga avtomatik jo'natish.

Ma'lumotlarni qayta ishlashning yana bir usulida ma'lumotlar elektron pochta manziliga, masalan, sayt administratoriga avtomatik ravishda jo'natiladi. mailto giper murojaati yordamida xabarni brauzerning o'zidan ham jo'natish mumkin.

Quyidagi misolda, foydalanuvchi Talabalar portali haqidagi fikr va mulohazalarini jo'natish mumkin bo'lgan forma yaratiladi. Bu forma portalning Bizga yozing! sahifasida yaratiladi. Ma'lumotlar mos shaklda formatlanib, PHP ning standart mail() funksiyasi yordamida jo'natiladi.

9.1-listing.

```
<html>
<head> <title>Bizga yozing!</title> </head>
<body bgcolor="#ffffff" text="#000000" link="#cbda74"
vlink="#808040" alink="#808040">
<? /* $form ichidagi qo'shtirnoqda joylashganlar hammasi ekranga chiq-
ishi kerak aks holda xatolik yuzaga keladi. */
$form = " <form action=\"feedback.php\" method=\"post\">
<input type=\"hidden\" name=\"seenform\" value=\"y\">
<h1>Bizga yozing!...</h1>
<h2>Bizga fikr va mulohazalaringizni jo'nating!</h2>
Ismingiz:<br> <input type=\"text\" name=\"name\" size=\"20\" max-
length=\"20\" value=\"\"><br>
```

```
Email manzilingiz:<br><input type=\"text\" name=\"email\" size=\"20\"
maxlength=\"40\" value=\"\"><br>
```

```
Fikr va mulohazalaringiz:<br> <textarea name=\"comments\" rows=\"3\"
cols=\"30\"></textarea><br>
```

```
<input type=\"submit\" value=\"submit!\">
</form>;
```

```
/* agar forma avval chiqazilmagan bo'lsa, uni chiqazishni tekshirish
uchun $seenform yashirin o'zgaruvchisining qiymati ishlatiladi */
```

```
if (!isset($seenform)) : print "$form"; else :
```

```
$oluvchi = "admin@mail.ru";
```

```
// $oluvchi o'zgaruvchisi forma ma'lumotlarini oluvchisini aniqlaydi
```

```
$mavzu = "($name) ismli foydalanuvchining xabari"; // Xabar mavzusi
```

```
$sarlavha = "From: $email"; // Qo'shimcha sarlavha
```

```
mail($recipient, $mavzu, $xabar, $sarlavha)or die("Xabarlaringiz
jo'natilmadi!");
```

```
print "Xabarlaringiz jo'natildi!";
```

```
endif;?>
```

Shu kabi masalalarda forma ma'lumotlarini jo'natishdan avval forma maydonlari to'ldirilganligini tekshirish lozim.

Foydalanuvchi tomonidan formaga ma'lumot kiritilganda ortiqcha bo'sh belgilar (probellar), ortiqcha teglar va boshqalar ishlatilishi mumkin. Bu holatda forma maydonlari yordamida kiritilgan satrlarni tahrirlash va o'zgartirish kerak bo'ladi. Bunda bizga PHP ning satrlar bilan ishlovchi funksiyalari yordam beradi.

9.2. Satrlarni tahrirlash

Satrni qisqartirish va kengaytirish: `trim()`, `ltrim()`, `chop()`, `str_pad()`

Formatlash jarayonida, satrga belgilarni qo'shish va o'chirish natijasida tez-tez satr uzunligini o'zgartirishga to'g'ri keladi. PHP da bunday masalalarni hal etish uchun bir nechta funksiyalar mavjud.

trim() funksiyasi

`trim()` funksiyasi satrni ikkala tomonidagi barcha bo'sh joylarni o'chiradi va hosil bo'lgan satrni chiqazadi. Ushbu funksiya bo'sh joylardan tashqari `\n`, `\r`, `\t`, `\v` i `\0` maxsus belgilarni ham o'chiradi.

Sintaksisi:

`string trim(string satr)`

ltrim() funksiyasi

`ltrim()` funksiyasi barcha bo'sh joylarni va maxsus belgilarni satrning o'ng tomonidan o'chiradi va hosil bo'lgan satrni chiqazadi. Bu funksiya ham bo'sh joylardan tashqari `trim()` funksiyasi kabi `\n`, `\r`, `\t`, `\v` i `\0` maxsus belgilarni o'chiradi.

Sintaksisi:

`string ltrim (string satr)`

chop() funksiyasi

`chop()` funksiyasi berilgan satrdan yakunlovchi belgilar va yangi qator belgilarini o'chiradi.

Sintaksisi:

`string chop(string satr)`

9.2-misol. Satrdan ortiqcha yangi qator belgilari (`\n`)ni `chop()` funksiyasi yordamida o'chirish.

```
<? $header = "Table of Contents\n\n";  
$header = chop($header); // $header = "Table of Contents"  
echo $header; ?>  
Natija:Table of Contents
```


str_pad() funksiyasi

str_pad() funksiyasi berilgan satrni ko'rsatilgan belgilar ketma-ketligi bilan to'ldiradi.

Sintaksisi:

string str_pad (string satr, int uzunligi [, string qo'shimcha [, int qo'shimcha_tipi]])

Bu yerda satr parametrda berilgan satr, uzunligi parametrda esa hosil bo'ladigan satr uzunligi ko'rsatiladi. Agar "qo'shimcha" parametri ko'rsatilmagan bo'lsa, satr probellar bilan to'ldiriladi, aks holda satr berilgan belgilar bilan to'ldiriladi. Odatga ko'ra satrning o'ng tomoni to'ldiriladi; satrni ko'rsatilgan yo'nalish bo'yicha to'ldirish uchun "qo'shimcha_tipi" parametrda STR_PAD_RIGHT(o'ng), STR_PAD_LEFT(chap) yoki STR_PAD_BOTH(ikkala tarafi) konstantalarini ko'rsatish kerak.

9.3-misol. str_pad() funksiyasini qo'llab satrni to'ldirish.

```
<? $food = "osh";  
$a="paz" ;  
print str_pad($food, 6, $a); // "oshpaz" satrini chiqazadi ?>
```

9.4-misol. str_pad() funksiyasining zarur bo'lmagan parametrlarining qo'llanilishi

```
<? $header = "Table of Contents";  
print str_pad($header, 27, "====", STR_PAD_BOTH);  
// ===== Tab e of Contents===== satrini chiqazadi ?>
```

Satrlarni turli usullar bilan qayta tahrirlash uchun ishlatish mumkin bo'lgan PHP ning funksiyalar to'plami mavjud.

HTML-tahrirlashning ishlatilishi: nl2br()

nl2br() funksiyasi parametr sifatida satrlarni qabul qiladi va undagi barcha yangi satr belgilarini
 HTML deskriptoriga almashtiradi. Bu uzun satrlarni brauzerga chiqazishda foydali.

Masalan, foydalanuvchi xabarini tahrirlash uchun ekranga chiqazamiz:

```
<p>Sizning xabaringiz jo'natildi.</p>  
<p><? echo nl2br ($comments) ; ?> </p>
```

Satrlarni nashrga berish uchun tahrirlash

Birinchi ma'ruzadan boshlab echo funksiyasini bir necha bor ishlatdik. echo — bu funksiya emas, balki konstruktsiya shuning

uchun unga murojaat etishda yumaloq qavs ishlatish shart emas. echo ekranga parametrlar bilan berilgan satrlarni chiqazishda yordam beradi. echo parametrlarini turlicha: vergul bilan ajratib yozish yoki konkatenatsiya amali bilan birlashtirish mumkin, lekin qavs ichiga olib yozish mumkin emas.

9.5-misol. echo funksiyasining ishlatilishi

```
<? echo "O'yla", "izla", "top";  
/* "O'yla izla top". echo ni bir necha parametrlarini konkatenatsiya  
amali yordamida berish mumkin*/  
echo "O'yla " . "izla " . "top ";  
// bu ham " O'yla izla top" yozuvini chiqazadi  
echo ("O'yla ", "izla ", "top "); // unexpected xatoligini chiqazadi  
>
```

echo buyrug'ining qisqartirilgan sintaksisi mavjud:

```
<?=chiqazish_uchun_satr?>
```

Bu yerda, chiqazish_uchun_satr parametri ekranga chiqarilishi lozim bo'lgan ixtiyoriy usulda berilgan satrni o'z ichiga oladi.

Misol uchun, quyidagi skript ekranga qizil yozuv bilan "Mening ismim Lola" satrini chiqazadi:

```
<? $name="Lola"?>  
<font color=red> Mening ismim <?=$name?></font>  
Natija: Mening ismim Lola
```

Satrlarni ekranga chiqazish uchun echo konstruksiyasidan tashqari bir qancha funksiyalar mavjud. Ulardan biri print() funksiyasi va uni boshqa xil turlari printf, sprintf va h.k.

Biz shu vaqtgacha satrlarni brauzer oynasiga chiqazish uchun echo konstruksiyasidan foydalandik. PHP ushbu vazifani bajaruvchi print() funksiyasini ham ishlatadi, biroq uning bitta farqli tarafi natija qiymatlarini (bajarish natijasiga qarab 0 yoki 1 ni) qaytaradi.

Ushbu ikkala konstruksiya satr qanday bo'lsa, o'sha holda ekranga chiqazadi. printf() va sprintf() funksiyalari yordamida esa bir-muncha qiyinroq tahrirlash ishlarini olib borish mumkin. Bu ikkala funksiya amalda asosan bir xil ishlaydi, lekin printf() tahrirlangan satrni brauzer oynasiga chiqazadi, sprintf() esa tahrirlangan satrni qaytaradi.

Ushbu funksiyalarning sintaksisi:

string sprintf (string format [, mixed args ...])

int printf (string format [, mixed args ...])

Ushbu ikkala funksiyada birinchi parametr sifatida tahrirlash uchun satr beriladi, bu yerda agar satr o'zgaruvchilardan iborat bo'lsa, o'zgaruvchilar o'rniga tahrirlanadigan kod ishlatiladi. Qolgan parametrlar esa satrni tahrirlashda kodlar o'rniga qo'yiladigan o'zgaruvchilardan iborat bo'ladi.

9.6-misol. echo konstruksiyasida kerakli satrni chiqazuvchi o'zgaruvchini quyidagicha ishlatdik:

```
<? $sum="2 so'm 50 tiyin";  
echo "Daftar $sum turadi."; ?>
```

Xuddi shunday natijani printf() funksiyasi yordamida olish uchun quyidagicha o'zgartirish kiritiladi:

```
<? $sum="2 so'm 50 tiyin";  
printf ("Daftar %s turadi.", $sum); ?>  
Natija:Daftar 2 so'm 50 tiyin turadi.
```

Satrni tahrirlashda %s ketma-ketligi almashtirish tasnifi deb yuritiladi.

Berilgan tasnif "satrni almashtirish" ma'nosini anglatadi. Ushbu holda bu ketma-ketlik satr sifatida berilgan \$sum o'zgaruvchisining qiymatiga almashadi.

Agar \$sum o'zgaruvchisida saqlanayotgan qiymat 2.5 bo'lsa, ikkala berilgan ifoda natijaga 2.5 ni chiqazadi.

printf() funksiyasida haqiqiy sonlarni tasvirlash uchun boshqacharoq almashtirish tasnifini ishlatish mumkin.

Satrlarni tahrirlash uchun ishlatiladigan bir qancha almashtirish tasniflari mavjud. Satrlarni tahrirlashda n ta almashtirish tasniflari ishlatilganda, n ta parametrlar ko'rsatilishi zarur. Har bir almashtirish tasnifi o'rniga berilgan tartibi bo'yicha parametrlar qo'yib boriladi.

9.7-misol.

```
<? $sum=2.5;  
$sum_sum=3.8;  
printf("Daftar %.2f(umumiy daftar %.2f) turadi.", $sum, $sum_sum);  
?>  
Natija:Daftar 2.50(umumiy daftar 3.80) turadi.
```

Ushbu holda birinchi almashtirish tasnifi \$sum o'zgaruvchisi uchun, ikkinchisi esa — \$sum_sum o'zgaruvchisi uchun ishlatildi.

Barcha almashtirish tasniflari bir xil formatga ega:

%[‘qo‘shimcha_belgi][-][kengligi][.aniqligi]tipi

Barcha almashtirish tasniflari % belgisi bilan boshlanadi. Agar % belgisini chiqazish talab etilsa, %% ketma-ketligi qo‘llaniladi.

qo‘shimcha_belgi shart bo‘lmagan parametr. Bu parametr ko‘rsatilgan kenglikkacha o‘zgaruvchilarni qo‘shish mumkinligini bildiradi. O‘zgaruvchilar noldan boshlab ushbu parametrda ko‘rsatilgan songacha hisoblagich qiymati sifatida qo‘shib borilishi mumkin.

“-” belgisi ham shart bo‘lmagan parametr. Bu parametr ma’lumotlar o‘ng tarafdin emas, chap tarafdin boshlab to‘g‘rilanishi uchun ko‘rsatiladi.

kengligi parametri printf() funksiyasida o‘zgaruvchi qiymatlarini almashtirish uchun qancha joy(belgi) zarurligini ko‘rsatish uchun ishlatiladi.

aniqligi parametri o‘nlik nuqtadan boshlanishi zarur. Bu parametr verguldan so‘ng tasvirlanadigan o‘nlik belgilarning miqdorini aniqlaydi.

Tasniflashning oxirgi parametri kod tipini bildiradi.

Ishlatiladigan kodlarning qisqacha tasnifi 9.1-jadvalda berilgan.

9.1-jadval

Almashtirish tasniflari tipi kodlari

Tipi	Vazifasi
b	Butun son sifatida sharhlanadi va ikkilik sanoq sistemasida chiqaziladi
c	Butun son sifatida sharhlanadi va belgi sifatida chiqaziladi
d	Butun son sifatida sharhlanadi va o‘nlik sanoq sistemasida chiqaziladi
f	Ikki karra aniqlikdagi son sifatida sharhlanadi va o‘nli kasr ko‘rinishida chiqaziladi
o	Butun son sifatida sharhlanadi va sakkizlik sanoq sistemasida chiqaziladi
s	Satr sifatida sharhlanadi va satr sifatida chiqaziladi
x	Butun son sifatida sharhlanadi va a-f kichik harflari bilan ifodalan-gan o‘n oltilik sanoq sistemasida chiqaziladi
X	Butun son sifatida sharhlanadi va A-F katta harflari bilan ifodalan-gan o‘n oltilik sanoq sistemasida chiqaziladi

Satr registrini (katta va kichik belgilarni) almashtirish

Satr registrini o'zgartirish uchun PHP da to'rtta funksiya mavjud:

`strtolower();`

`strtoupper();`

`ucfirst();`

`ucwords();`

strtolower() funksiyasi

`strtolower()` funksiyasi satrdagi barcha alifbo belgilarini kichigiga (kichik registrga) almashtiradi.

Sintaksisi:

`string strtolower(string satr)`

Alifbo tarkibida bo'lmagan belgilarni funksiya almashtirmaydi.

9.8-misol. `strtolower()` funksiyasining qo'llanilishi.

```
<? $gap = "TENNIS O'YNASH va PHPDA DASTURLASH mening  
sevimli mashg'ulotim!";  
$gap = strtolower($gap);  
Echo $gap; ?>  
Natija:tennis o'ynash va phpda dasturlash mening sevimli mashg'ulotim!
```

Satr belgilarini nafaqat kichigiga, balki kattasiga ham almashtirish mumkin.

strtoupper() funksiyasi

Satr belgilarini kattasiga almashtirish `strtoupper()` funksiyasi yordamida amalga oshiriladi.

Sintaksisi:

`string strtoupper (string satr)`

Alifbo tarkibida bo'lmagan belgilarni funksiya almashtirmaydi.

9.9-misol. `strtoupper()` yordamida satr belgilarini kattasiga almashtirishni ko'rib o'tamiz.

```
<? $gap = "tennis o'ynash va PHPda dasturlash mening sevimli  
mashg'ulotim!";  
$gap = strtoupper($gap);  
echo $gap; // funksiya ishlatilgandan so'ng $gap satri  
/* " TENNIS O'YNASH VA PHPDA DASTURLASH MENING  
SEVIMLI MASHG'ULOTIM!" */ ?>
```

ucfirst() funksiyasi

ucfirst() funksiyasi shartga ko'ra satrdagi birinchi belgini, katasiga almashtiradi.

Sintaksisi:

string ucfirst (string satr)

Alifbo tarkibida bo'lmagan belgilarni funksiya almashtirmaydi.

9.10-misol. ucfirst() funksiyasining ishlatilishi

```
<? $gap = "tennis o'ynash va PHPda dasturlash mening sevimli mashg'ulotim!";  
$gap = ucfirst($gap); echo $gap;?>  
Natija:Tennis o'ynash va PHPda dasturlash mening sevimli mashg'ulotim!
```

ucwords() funksiyasi

ucwords() funksiyasi satr tarkibidagi har bir so'zning birinchi belgisini kattasiga almashtiradi.

Sintaksisi:

string ucwords (string satr")

Alifbo tarkibida bo'lmagan belgilarni funksiya almashtirmaydi.

"So'z" satrning boshqa elementlaridan probel bilan ajratilgan belgilar ketma-ketligi sifatida aniqlanadi.

```
<? $gap = "tennis o'ynash va PHPda dasturlash mening sevimli mashg'ulotim!";  
$gap = ucwords($gap); echo $gap; ?>  
Natija:Tennis O'ynash Va PHPda Dasturlash Mening Sevimli Mashg'ulotim!
```

9.11-misol. ucwords() funksiyaning ishlatilishi

Satrlarni saqlash uchun tahrirlash

PHP da belgilarni bekor qilish uchun maxsus belgilangan ikkita funksiya mavjud. Ixtiyoriy satrni ma'lumotlar bazasiga yozishdan avval ularni AddSlashes() funksiyasi yordamida tahrirlash lozim.

9.12-misol. AddSlashes() funksiyasining ishlatilishi

```
<? $comments = AddSlashes($comments); ?>
```

Boshqa satrlar bilan ishlovchi funksiyalar kabi AddSlashes() funksiyasi ham parametr sifatida satrni qabul qiladi va tahrirlangan satrni qaytaradi.

AddSlashes() funksiyasi ishlatilganda satr ma'lumotlar bazasida teskari chiziq belgisi bilan birga saqlanadi. Satrni olishda teskari

chiziq belgisini o'chirishni unutmaslik lozim. Teskari chiziq belgisini o'chirish **StripSlashes()** funksiyasi yordamida amalga oshiriladi:

9.13-misol. StripSlashes() funksiyasining ishlatilishi

```
<? $comments = StripSlashes($comments); ?>
```

9.3. Satrlarni taqqoslash

Satr uzunligini tekshirish (aniqlash): strlen() funksiyasi

Belgilardan iborat satr uzunligini strlen() funksiyasi yordamida aniqlash mumkin.

Sintaksisi:

```
int strlen(string satr)
```

9.14-misol. Berilgan satr uzunligini strlen() funksiyasi yordamida hisoblash.

```
<? $string = "hello";  
$length = strlen($string); print $length; ?>  
Natija:5
```

Bu funksiya yordamida kiritilgan ma'lumotlarning to'g'riligini ham tekshirish mumkin. \$email o'zgaruvchisida saqlanadigan, formada yaratilgan elektron pochta manzilini ko'rib o'tamiz. \$email o'zgaruvchisida saqlanayotgan elektron pochta manzili to'g'riligini tekshirishning asosiy usullaridan biri – uning uzunligini tekshirish. O'ylab qaraganda eng kichik elektron pochta manzilining uzunligi oltita belgiga teng – masalan, a@a.to kabi. Shunga ko'ra, dastur agar manzil uzunligi oltidan kichik bo'lsa, xato haqidagi xabarni chiqazsin:

```
<? if (strlen($email) < 6) { echo "email manzil noto'g'ri kiritilgan"; }  
exit; // PHP sahifani bajarilishini to'xtatish ?>
```

Bu albatta ma'lumot to'g'riligini tekshirishning eng sodda yo'li. Keyinroq birmuncha murakkabroq usullarini ham ko'rib o'tamiz.

**Ikkita satrni taqqoslash: strcmp(), strcasecmp(),
strspn(), strcspn()**

Ikkita satrni taqqoslash ixtiyoriy dasturlash tilining satrlar bilan ishlash amallarining asosiy qismi hisoblanadi. Bu masalani bir nechta turli usullar bilan yechish mumkin, PHP da satrlarni taqqoslash uchun to'rtta funksiya mavjud:

```
strcmp();
```

strcasecmp();

strspn();

strcspn().

Strcmp() funksiyasi

strcmp() funksiyasi ikkita satr belgilarini taqqoslaydi. Bu funksiya satrlarning registriga (katta-kichikligiga) ahamiyat beradi.

Sintaksisi:

int strcmp(string satr1, string satr2)

0, agar satr1 va satr2 mos tushsa;

< 0, agar satr1 satr2 dan kichik bo'lsa;

> 0, agar satr2 satr1 dan kichik bo'lsa.

9.15-misol. strcmp() funksiyasini qo'llab ikkita satrni taqqoslash.

```
<? $string1 = "non";  
$string2 = "non";  
if ((strcmp($string1, $string2)) == 0) :  
print "$string1 va $string2 satrlari ekvivalent!<br>";  
endif;  
// if buyrug'i true qiymatni beradi  
$string1 = "non";  
$string2 = "tuz";  
if ((strcmp($string1, $string2)) == 0) {  
echo "$string1 va $string2 satrlari ekvivalent!";}  
else {echo "$string1 va $string2 satrlari ekvivalent emas!";} ?>
```

Natija:

non va non satrlari ekvivalent!

non va tuz satrlari ekvivalent emas!

Strcasecmp () funksiyasi

strcasecmp() funksiyasi strcmp() kabi ishlaydi, bitta farqli tomoni – taqqoslashda belgilar registriga (katta-kichikligiga) ahamiyat qaratilmaydi.

Sintaksisi:

int strcasecmp (string satr1, string satr2)

9.16-misol. strcasecmp() funksiyasini qo'llab ikkita satrni solishtirish.

```
<? $string1 = "non"; $string2 = "Non";  
if ((strcasecmp($string1, $string2)) == 0) : print "Satrlar ekvivalent!";  
endif;
```



```
// if buyrug'i true qiymatni beradi ?>  
Natija:Satrlar ekvivalent!
```

strspn() funksiyasi

strspn() funksiyasi birinchi satrl segmenti tarkibidagi belgilarni satr2 da ishtirok etganlarini uzunligini chiqazadi.

Sintaksisi:

```
int strspn (string satr1, string satr2)
```

9.17-misol. strspn() funksiyasini parollarni tekshirishda ishlatilishi.

```
<? $password = "12345";  
if (strspn($password, "1234567890")!=strlen($password)) :  
print "Password cannot consist solely of numbers!";  
endif; ?>  
Natija:
```

Strcspn() funksiyasi

strcspn() funksiyasi birinchi segment satrl tarkibidagi belgilarni satr2 da ishtirok etmaganlarini uzunligini chiqazadi.

Sintaksisi:

```
int strcspn (string satr1, string satr2)
```

9.18-misol. strcspn() funksiyasini parollarni tekshirish uchun ishlatilishi.

```
<? $password = "12345";  
if (strcspn($password, "1234567890") == 0) :  
print "Password cannot consist solely of numbers!"; endif; ?>  
Natija:Password cannot consist solely of numbers!
```

9.4. Satrlar bilan ishlovchi funksiyalar yordamida satrlarni ajratish va birlashtirish

Juda ham kerakli funksiyalardan biri bular – satrni qismlarga ajratish funksiyasi va uning aksi satrlarni bitta satrga birlashtirish funksiyasi. Nima uchun bu funksiyalar kerak? Masalan, agar foydalanuvchi xohishiga ko'ra formani dinamik tarzda generatsiya qilmoqchi bo'lsak, tanlash ro'yxatini yaratish uchun elementlarni biror-bir belgi yordamida ajratib kiritishni taklif etish mumkin. Va hosil bo'lgan qiymatlar ro'yxatini saralash uchun satrni qismlarga ajratish kerak bo'ladi. So'ng olingan ro'yxatlar qiymatlarini qayta ishlash uchun satrni bo'laklarga bo'lish kerak bo'ladi. Bu kabi bo'laklarga

ajratishni amalga oshirish uchun PHP ning bir nechta funksiyalari-dan foydalanish mumkin:

```
explode (ajratuvchi, berilgan_satr  
[,elementlarning_eng_katta_soni])  
split (shablon, berilgan_satr  
[,elementlarning_eng_katta_soni])  
preg_split (shablon, berilgan_satr  
[,elementlarning_eng_katta_soni  
[,bayroq]])
```

Oxirgi ikkita funksiya doimiy ifodalar bilan ishlashadi, shuning uchun biz ularni ko'rib o'tmaymiz. Birmuncha soddaroq bo'lgan funksiya — explode() funksiyasi bilan tanishamiz.

explode() funksiyasi satrni elementlarga ajratadi va bu element-larni massiv sifatida chiqazadi.

Sintaksisi:

```
array explode (string ajratuvchi, string berilgan_satr [,int ele-  
mentlarning_eng_katta_soni])
```

explode() funksiyasi berilgan_satrni ajratuvchi parametrda ko'rsatilgan belgi yordamida qism satrlarga ajratadi va bu qism satr-lardan hosil bo'lgan massivni chiqazadi. Agar qo'shimcha parametr elementlarning_eng_katta_soni berilgan bo'lsa, u holda massiv ele-mentlari soni bu sonidan katta bo'lmaydi, oxirgi elementda satrning barcha qoldig'i yoziladi. Agar ajratuvchi parametri sifatida bo'sh satr " " ko'rsatilsa, u holda explode() funksiyasi false ni qaytaradi. Agar ajratuvchi belgisi berilgan satrda bo'lmasa, u holda berilgan satr o'zgartirilmagan holda chiqaziladi. Masalan,

```
<? $info = "wilson | baseball | indians";  
$user = explode("|", $info);  
// $user[0] = "wilson"; $user[1] = "baseball"; $user[2] = "Indians";  
print_r($user); ?>
```

9.19-misol. explode() funksiyasining qo'llanilishi.

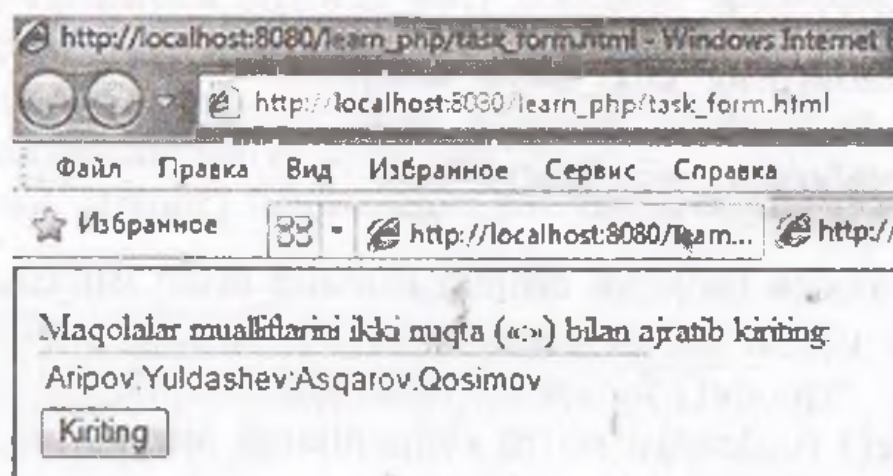
Ochiluvchi ro'yxat forma elementini yaratamiz. Bu forma ele-menti qiymatlari foydalanuvchi tomonidan kiritiladi. Buning uchun quyidagicha dastur kodini yozamiz:

```
<form action=test.php>  
Maqolalar mualliflarini ikki nuqta (":") bilan ajratib kiriting:
```



```
<br> <input type=text name=muallif size=40>
<br> <input type=submit value=Kiriting> </form>
```

Natijada quyidagi formaga ega bo‘lamiz:



9.1-rasm. Ochiluvchi ro‘yxat hosil qilish uchun qiymatlarni kiritish formasi

Bu kiritilgan ma’lumotlar test.php sahifasida qayta ishlanadi:

```
<? $str = $_GET["muallif"];
$names = explode(":", $str);
// ":" yordamida foydalanuvchi tomonidan kiritilganlarni ajratamiz
$s = "<select name= muallif >"; // ochiluvchi ro'yxatni yaratamiz
foreach ($names as $k => $name) {
    $s .= "<option value=$k>$name"; } // ro'yxatga elementlarni qo'shamiz
$s .= "</select>";
echo $s; ?>
```

implode() funksiyasi

Satrni qismlarga ajratishdan tashqari ba’zida, buning aksi, bir nechta satrni bitta butun satrga birlashtirish zarurati tug‘iladi. PHP da buning uchun implode() funksiya taklif etiladi. explode() funksiyasi satrni massiv elementlariga ajratsa, hosil bo‘lgan massivni implode() funksiyasi yordamida bitta satrga birlashtirish mumkin.

Sintaksisi:

implode (satr_massivi, birlashtiruvchi)

Bu funksiya birlashtiruvchi belgisi yordamida berilgan massiv elementlarini birlashtiradi. explode() funksiyasidan farqli ravishda implode() funksiyasida parametrlar tartibining ahamiyati yo‘q.

9.20-misol. implode() funksiyasining qo'llanilishi.

Biror-bir shaxsning manzili: respublikasi, shahar yoki viloyati, tumani, mahallasi alohida-alohida maydonlarda saqlangan. Ularni birlashtirib manzil sifatida ko'rsatilishi kerak. Bu maydonlardagi satrlarni bitta satrga birlashtirish uchun implode() funksiyasini ishlatish mumkin:

```
<? $aj_manzil = array ("O'zbekiston", "Toshkent  
Shahri", "Yunusobod tumani", "Chinor mahallasi");  
$manzil = implode(",", $aj_manzil);  
Echo $manzil; ?>  
Natija:O'zbekiston,Toshkent shahri,Yunusobod tumani,Chinor mahallasi
```

join() funksiyasi implode() funksiyasining o'xshashi bo'lib, u bajaragan barcha vazifalarni bajaradi, bu ikkala funksiya bir-biridan faqatgina nomi bilan farq qiladi.

strtok() funksiyasi

strtok() funksiyasi satrni ikkinchi parametrda berilgan ajratuvchiga asosan leksemalarga ajratadi.

Sintaksisi:

string strtok (string satr, string ajratuvchi)

strtok() funksiyasining bitta noqulay tarafi bor: satrni to'liq ajratish uchun funksiyaga bir necha marotaba ketma-ket murojaat etish kerak. Navbatdagi murojaatda funksiya satrdan keyingi leksemani belgilaydi. Shu bilan birga "satr" parametri funksiyaga bir marta beriladi, funksiya esa har gal satrni ajratgandan so'ng ketma-ket joriy pozitsiyani belgilab boradi. Bu jarayon satr leksemalarga to'liq ajratilguncha bajariladi.

9.21-misol. Bir nechta ajratuvchilar yordamida berilgan satrni ajratish.

```
<? $info = "TDPU:tdpu_info@edu.uz | O'zbekiston, Toshkent";  
// Chegaralar - ikki nuqta (:), vertikal chiziq (|) va vergul (,)  
$ajrat = ":", "|";  
$ajratilgan = strtok($info, $ajrat);  
while ($ajratilgan) :echo "Element = $ajratilgan<br>";  
$ajratilgan = strtok($ajrat);endwhile; ?>  
Natija:  
Element = TDPU  
Element = tdpu_info@edu.uz  
Element = O'zbekiston
```


substr() funksiyasi

Ba’zida biz qidirilayotgan satr qaysi belgidan boshlanishini bilmaymiz, lekin masalan, bu satr beshinchi elementdan boshlanib, berilgan satr oxiriga qarab ikkita belgidan so’ng tugatiladi. Bunday holatda qism satrni belgilash qanday amalga oshiriladi? Buning uchun substr() funksiyasidan foydalanamiz.

substr() funksiyasi berilgan_satrdan boshlanish_belgisining_o’rnidan boshlab uzunligi uzunligi parametriga teng bo’lgan qism satrni ajratib oladi.

Sintaksisi:

string substr(string berilgan_satr, int boshlanish_belgisining_o’rni [, int uzunligi])

Agar uzunligi parametri ko’rsatilmasa, u holda funksiya ko’rsatilgan boshlanish_belgisi_o’rnidan boshlab berilgan_satr oxirigacha bo’lgan qism satrni chiqazadi. Bu funktsiyani ishlatishda quyidagicha holatlar yuzaga keladi. boshlanish_belgisining_o’rni parametri ham musbat butun son hamda manfiy butun son bo’lishi mumkin.

agar “boshlanish_belgisining_o’rni” parametri musbat butun son bo’lsa, qism satr berilgan satrning “boshlanish_belgisining_o’rni” parametrda berilgan nomer o’rnidan boshlanadi;

agar “boshlanish_belgisining_o’rni” parametri manfiy butun son bo’lsa, qism satr (satr uzunligi- boshlanish_belgisining_o’rni =) o’rnidan boshlanadi;

agar “uzunligi” parametri musbat butun bo’lsa, qism satr “boshlanish_belgisining_o’rni”dan boshlab (boshlanish_belgisining_o’rni +uzunligi=) o’rnigacha bo’lgan barcha belgilarni chiqazadi. Agar bu qiymat satr uzunligidan katta bo’lsa, u holda satr oxirigacha bo’lgan belgilar chiqaziladi;

“uzunligi” parametri manfiy son ham bo’lishi mumkin. Agar “uzunligi” parametrda manfiy son berilsa, u holda satr oxiridan ko’rsatilgan songa teng belgi tashlab yuboriladi.

9.22-misol. substr() funksiyasining qo’llanilishi

```
<? $car = "1998 Nexia";  
$model = substr($car, 5);  
// 5 belgidan boshlab satr oxirigacha bo’lgan belgini chiqazadi
```

```
echo $model; // $model = "Nexia" ?>
```

Natija:Nexia

9.23-misol. substr() funksiyasining qo'llanilishi
"uzunligi" parametri musbat bo'lgan holat:

```
<? $car = "1998 Nexia";  
$model = substr($car, 0, 4);  
/* 0 belgidan boshlab 4 ta belgili qism satrni belgilaydi*/  
echo $model; ?>  
Natija:1998
```

9.24-misol. substr() funksiyasining qo'llanilishi
"uzunligi" parametri manfiy bo'lgan holat:

```
<? $car= "<b>1998 Nexia</b>";  
echo $car, "<br>";  
$model = substr($car, 3, -4);  
/* 3 belgidan boshlanib oxiridan 4 ta belgi tushirib qoldirilgan qism satrni  
belgilaydi*/  
echo $model; ?>  
Natija:  
1998 Nexia  
1998 Nexia
```

9.24-misoldagi kabi misollarni yechishda strip_tags() funksiyasidan foydalanish substr() funksiyasiga nisbatan qulayroq.

strip_tags() funksiyasi

Sintaksisi:

strip_tags(satr [, qoldiriladigan_teglar])

Bu funksiya satrdagi barcha html va php teglarni o'chiradi. Qo'shimcha parametr — qoldiriladigan_teglar yordamida satrdan o'chirilishi kerak bo'lmagan teglarni berish mumkin. Teglar ro'yxati hech qanday belgi bilan ajratilishi kerak emas. Agar noto'g'ri yoki to'liq bo'lmagan teg uchrasa funksiya bu haqida xabar beradi.

9.25-misol. strip_tags() funksiyasining qo'llanilishi

```
<?php  
$string = "<b>Bold text</b>  
<i>Italic text</i>";  
$str = strip_tags($string); // satrdan barcha teglarni o'chiradi  
$str1 = strip_tags($string, '<i>');
```



```
// <i> dan tashqari barcha teglarni o'chiradi
$str2 = strip_tags($string, '<i><b>');
// <i> va <b> dan boshqa barcha teglarni o'chiradi
echo $str."<br>".$str1."<br>", $str2;
?>
```

Natija:

Bold text *Italic text*

Bold text *Italic text*

Bold text *Italic text*

substr() funksiyasi yordamida boshqa misolni ko'ramiz.

9.26-misol. Salomlashuv va imzosi bilan berilgan qandaydir xabar bor. Birinchi navbatda salomlashuvni, so'ng esa imzoni o'chiramiz va faqatgina xabarning asosiy qismini qoldiramiz.

```
<?php
$text = "Salom! Bugun biz satrlar bilan ishlashni o'rganamiz. Muallif.";
$no_hello = substr($text, 7); // salomlashuvni o'chiramiz
$content = substr($text, 7, 45);
// substr($text, 7, -9) kabi imzoni o'chiramiz
echo $text, "<br>", $no_hello,
"<br>", $content; ?>
```

Natija:

Salom! Bugun biz satrlar bilan ishlashni o'rganamiz. Muallif.

Bugun biz satrlar bilan ishlashni o'rganamiz. Muallif.

Bugun biz satrlar bilan ishlashni o'rganamiz.

Agar satrdan aniq bir belgi kerak bo'lsa va bu belgini tartib nomerini bilsak, bu holda substr() tipidagi funksiyalarni ishlatish shart emas. Bundan ham oddiyroq sintaksisdan foydalanish mumkin – buning uchun satriy o'zgaruvchidan so'ng belgi nomerini figurali qavs ichiga olib yozsak kifoya. Misol uchun berilgan satrda "a" harfi hisob bo'yicha ikkinchida turibdi, u holda:

```
$text = "Salom! Bugun biz satrlar bilan ishlashni o'rganamiz. Muallif.";
echo $text{1}; // "a" belgisini chiqazamiz
```

Bu belgi nomeri ikki emas, balki bir, bundan bilish mumkinki, satr belgilari tartib nomeri noldan boshlanadi.

Satrnı belgilari va ularni tartib nomeri haqida gapirar ekanmiz, u holda satrda nechta belgi bor va ularni qanday hisoblash mumkin

kabi savollar yuzaga keladi. Satr belgilari soni – bu satr uzunligidir. Satr uzunligini strlen () funksiyasi yordamida hisoblash mumkin.

substr_count() funksiyasi

substr_count() funksiyasi berilgan satrdagi qism satrlarni necha marta uchraganligi sonini chiqazadi.

Sintaksisi:

int substr_count (string berilgan satr, string qism satr)

9.27-misol. substr_count() funksiyasi yordamida ain qism satri necha marta uchraganligini hisoblang:

```
<? $tng_twist = "The rain falls mainly on the plains of Spain";  
$count = substr_count($tng_twist, "ain"); // $count = 4  
print $count; ?>  
Natija:4
```

Demak, qism satrlarni qanday topish va belgilashni ko'rib o'tdik. Endi berilgan satrdagi qism satrlarni boshqa yangi qism satrlarga almashtirishni o'rganamiz.

9.5. Qism satrlarni qidirish va almashtirish

Qism satrlar o'rnini aniqlash: strpos(), strrpos()

strpos() funksiyasi

Berilgan satr tarkibida qism satr mavjudligini tekshirish uchun strpos() funksiyasi ishlatiladi.

Sintaksisi:

strpos (berilgan_satr, qidirilayotgan_qism_satr
[,qaysi_belgidan_boshlab_qidirish_kerak])

Qidirilayotgan_qism_satrni berilgan_satrdagi o'rnini chiqazadi, aks holda agar qidirilayotgan_qism_satr berilgan_satr tarkibidan topilmasa, false mantiqiy javobni qaytaradi. Qo'shimcha parametr qidiruv satrning qaysi belgisidan boshlanishi kerakligini ko'rsatish uchun imkoniyat yaratadi. Mantiqiy false dan tashqari funksiya false (masalan, 0 yoki " ") ning o'rnini bosuvchi boshqa qiymatlarni ham chiqazishi mumkin. Shuning uchun qidirilayotgan_qism_satr mavjudligini tekshirish uchun "===" ekvivalent amalini qo'llash tavsiya etiladi.

9.28-misol. strpos() funksiyasi yordamida satr tarkibidagi qism satrni mavjudligini tekshirish.

```
<? $str = " Albatta aytish kerak-ki, PHP dasturlash tillari orasida eng  
yaxshisiga aylandi va ulardagi yetishmovchiliklarni maksimal darajada bar-
```


taraf qila oldi. PHP texnologiyasining vazifasi - Webda ishlovchilarga html-sahifani oson va tez, dinamik holda ishlashi va o'zgartirish uchun imkoniyatlar yaratishdan iborat. PHP texnologiyasi ularni a'lo darajada bajaradi.”;

```
$pos = strpos($str,"PHP");
```

```
if ($pos!== false) echo "Qidirilayotgan satr $pos o'rinda topildi";
```

```
else echo "Qidirilayotgan satr topilmadi"; ?>
```

Natija:Qidirilayotgan satr 26 o'rinda topildi

Agar qidirilayotgan_qism_satr parametri qiymati satrli bo'lmasa, u holda butun tipga akslantiriladi va ASCII-kod belgisi sifatida qaraladi. PHP da ixtiyoriy belgini ASCII-kodini olish uchun ord ("belgi") funksiyasidan foydalanish mumkin.

Masalan, agar \$pos = strpos(\$str,116); kabi yozilsa interpretator "t" belgisi qidirilayotgan deb hisoblaydi. Agar bu satrni yuqoridagi misolga qo'yilsa va natija chiqazilsa, u holda qidirilayotgan satr 5-o'rinda deb chiqaziladi.

Ord() funksiyasiga teskari funksiya bu chr("kod") funksiyasidir. Bu funksiya berilgan kodga mos ASCII-kodi belgisini chiqazadi.

strpos funksiyasi berilgan_satr tarkibidagi qidirilayotgan_qism_satrni faqatgina birinchi uchraganini joylashgan o'rnini chiqazadi. Berilgan_satr tarkibidagi qidirilayotgan_qism_satrni oxirgi uchraganini joylashgan o'rnini aniqlovchi funksiya ham mavjud. Bu funksiya strrpos().

strrpos() funksiyasi

Sintaksisi:

```
strrpos (berilgan_satr, qidirilayotgan_belgi)
```

strpos() dan farqli ravishda bu funksiya qidirilayotgan_belgini berilgan_satrdagi oxirgi uchragan o'rnini topish uchun imkon yaratadi. Imkoniyatlariga ko'ra bu funksiyaning strpos() funksiyasidan farqi shundaki, bu funksiya butun satr o'rnini emas, balki faqatgina alohida belgi o'rnini aniqlay oladi. Agar strrpos() da ikkinchi parametr sifatida satr berilsa, u holda qidirish vaqtida uning faqatgina birinchi belgisi ishlatiladi.

Shunday holatlar bo'ladiki, qidirilayotgan satr qayerda joylashganligini o'rnini bilish zarur bo'lmaydi, balki shu o'rindan boshlab satrda joylashgan barcha belgilarni olishga to'g'ri keladi. Bunday ho-

latlarda strpos() va strrpos() funksiyalarini emas, balki qism satrlarni belgilash uchun yordam beruvchi funksiyalardan foydalanish kerak.

**Satrdagi qism satrlarni qidirish: strstr(), strchr(),
strrchr(), stristr()**

Strstr() funksiyasi

PHP da berilgan satrdan qism satrni ajratib olish haqida gap ketganda birinchi o'rinda strstr() funksiyasiga to'xtalishimiz mumkin.

Sintaksisi:

string strstr (string berilgan_satr, string qidirilayotgan_satr)

strstr() funksiyasi qidirilayotgan_satrning birinchi uchraganidan boshlab berilgan_satr oxirigacha bo'lgan qism satrni ajratib oladi. Masalan,

```
<? $url = "http://www.apress.com";  
$domain = strstr($url, ".");  
echo $domain; ?>  
Natija: .apress.com
```

Agar qidirilayotgan_satr topilmasa, u holda funksiya false ni qaytaradi. Agar qidirilayotgan_satr satrli ma'lumotlar tipi ko'rinishida bo'lmasa, u holda butun tipga o'tkaziladi va belgi kodi sifatida qaraladi. Bundan tashqari, bu funksiya registrga (katta-kichikligiga) ahamiyat qaratadi, agar biz "PHP" va "php" so'zlarini parallel ravishda qidirmoqchi bo'lsak, u holda natija turlicha bo'ladi. strstr() bilan birgalikda unga juda o'xshash bo'lgan funksiya strchr() ni ham ishlatish mumkin.

9.29-misol. strstr() funksiyasining ishlatilishi:

Nomi, muallifidan iborat bo'lgan satrdan "Nomi" so'zi bilan boshlanuvchi qism satrni ajratib olamiz.

```
<? $str = "Muallif: Azimov Akrom(<a href=mailto:akrom@mail.ru>xat  
yozing</a>), Nomi: 'Web dasturlash tili'";  
echo "<b>Berilgan satr: </b>",$str;  
if (!strstr($str, "Nomi")) echo "Satr topilmadi<br>";  
else echo "<p><b>Olingan satr: </b>";  
    strstr($str, "Nomi"); ?>  
Natija:  
Berilgan satr: Muallif: Azimov Akrom(xat yozing), Nomi: 'Web dastur-  
lash tili'  
Olingan satr: Nomi: 'Web dasturlash tili'
```


Qidirishda qism satrni registrga (katta-kichikligiga) ahamiyat qaratmasligi uchun ushbu funksiyaga o'xshash funksiya – `stristr()` (berilgan_satr, qidirilayotgan_satr) funksiyasidan foydalanish mumkin. Bu funksiya xuddi `strstr()` kabi ishlaydi, faqatgina qidirishda qidirilayotgan_satr belgilari registrga (katta-kichikligiga) ahamiyat qaratmaydi. `strchr()` funksiyasi `strstr()` kabi ishlaydi, uning ikkinchi varianti `strrchr()` esa `strstr()` funksiyasiga o'xshash bitta farqi berilgan_satrda qism satrni qidirilayotgan_satrni oxirgi uchraganidan boshlab chiqazadi.

Amaliyotda juda kam holatlarda aniq so'z yoki satr bilan boshlanuvchi qism satrni olish kerak bo'ladi, shuning uchun `strstr()` funksiyasi tez-tez ishlatilmaydi. Bundan tashqari, PHP ning qidirish uchun qulay boshqa funksiyalari ham mavjud.

Qism satrlarni almashtirish: `str_replace()`, `substr_replace()`

`str_replace()` funksiyasi

Bu funksiya oddiy va qulay bo'lib, almashtiriladigan qism satrga alohida talablar bo'lmagan holda, ko'pgina masalalarni hal qilishga imkon beradi. Murakkab shartlar bilan berilgan almashtirishlarda doimiy ifodalar usuli va mos funksiyalar `ereg_replace()`, `preg_replace()` ishlatiladi.

Sintaksisi:

`string str_replace (string qism_satr, string yangi_qism_satr, string satr)`

`str_replace()` funksiyasi satr tarkibida uchraydigan qism_satrni boshqa yangi_qism_satrga almashtirish uchun ishlatiladi.

yoki

`str_replace (qiymat, yangi qiymat, obyekt)`

`str_replace()` funksiyasi qaralayotgan obyektidan qiymatni qidiradi va ushbu qiymatni belgilangan yangi qiymatga almashtiradi.

Keyingi sintaksisda parametrlar satr emas, balki obyekt va qiymat deb yuritilganligining sababi, PHP 4.0.5 versiyasidan boshlab, bu funksiyaning ixtiyoriy parametri massiv bo'lishi mumkin.

Agar qism satr berilgan satrda uchramasa, berilgan satr o'zgarmaydi.

9.30-misol. `str_replace()` funksiyasining qo'llanilishi

```
<?php
```

```
$sevimli_fan= "Mening sevimli fanim matematika va fizika";
```

```
$sevimli_fan= str_replace("matematika", "informatika", $sevimli_fan);
// $sevimli_fan= " Mening sevimli fanim informatika va fizika"
echo $sevimli_fan; ?>
```

Natija: Mening sevimli fanim informatika va fizika

Agar almashtirish bajariladigan obyekt massiv bo'lsa, u holda bu amal massivning barcha elementlari uchun bajariladi va natija yangi massiv sifatida chiqaziladi.

9.31-misol. str_replace() funksiyasining qo'llanilishi. 1-usul.

```
<?php
$salomlashuv = array("Salom", "Salom hammaga!", "Salom, do'stlar!");
$yangi_salomlashuv = str_replace("Salom", "Xayrli tong", $salomlashuv);
// almashtirishdan so'ng
print_r($yangi_salomlashuv);
?>
Natija: Array (
[0] => Xayrli tong
[1] => Xayrli tong hammaga!
[2] => Xayrli tong, do'stlar! )
```

Agar qidirilayotgan qiymat va almashtiriladigan qiymat – massivlar bo'lsa, u holda har bir massivdan bittadan qiymat olinib, obyektida qidirish va almashtirish bajariladi. Agar almashtiriladigan qiymat qidirilayotgan qiymatdan kichik bo'lsa, u holda yangi qiymat sifatida bo'sh satr olinadi.

9.32-misol. str_replace() funksiyasining qo'llanilishi. 2-usul.

```
<?php
$greeting = array("Salom", "Salom hammaga!", "Salom, do'stlar!", "Assalomu alaykum", "Assalomu alaykum, o'rtoqlar", "Hi");
// obyekt
$search = array ("Salom", "Assalomu alaykum", "Hi");
// qidirilayotgan qiymat
$replace = array ("Xayrli tong", "Xayrli kun");
// almashtirilishi kerak bo'lgan qiymat
$new_greet = str_replace($search, $replace, $greeting);
// almashtirishni bajarish
print_r($new_greet); //hosil bo'lgan massivni chiqarish
?>
```


Natija:

Array ([0] => Xayrli tong

[1] => Xayrli tong hammaga!

[2] => Xayrli tong, do'stlar!

[3] => Xayrli kun

[4] => Xayrli kun, o'rtoqlar

[5] =>)

Agar qidirilayotgan qiymat massiv, almashtiriladigan qiymat esa satr bo'lsa, u holda bu satr barcha topilgan qiymatlarni almashtirish uchun ishlatiladi.

`str_replace()` funksiyasi registrga (belgilarning katta-kichikligiga) ahamiyatli, lekin registrga ahamiyatli bo'lmagan unga o'xshash funksiya mavjud, bu funksiya — `str_ireplace()`. Lekin bu funksiya PHP ning hamma versiyalarida ham qo'llanilmaydi.

`substr_replace()` funksiyasi

Bu funksiya ko'rib o'tilgan ikkita funksiya `str_replace()` va `substr()` larning xususiyatini o'zida jamlagan.

Sintaksisi:

`substr_replace (berilgan_satr, almashtiriladigan_satr, belgining_boshlang'ich_o'rni[, uzunligi])`

Bu funksiya parametrlarga asosan berilgan_satrni hammasini yoki qismini belgining_boshlang'ich_o'rni parametrida ko'rsatilgan o'rnidan boshlab almashtiriladigan_satrga almashtiradi. Qo'shimcha parametr sifatida berilgan uzunligi yordamida almashtiriladigan belgilar sonini chegaralash mumkin. Aslida, satrdagi almashtirilishi kerak bo'lgan qism satr ko'rsatilmaydi, faqatgina u qayerda joylashgani va qanday uzunlikka ega ekanligi ko'rsatiladi va shu xususiyati bilan `substr_replace()` funksiyasi `str_replace()` funksiyasidan farq qiladi.

`substr()` funksiyasidagi kabi belgining_boshlang'ich_o'rni va uzunligi parametrlari manfiy bo'lishi mumkin:

agar "belgining_boshlang'ich_o'rni" parametri musbat bo'lsa, almashtirish ko'rsatilgan pozitsiyadan boshlanadi;

agar "belgining_boshlang'ich_o'rni" parametri manfiy bo'lsa, almashtirish (satr uzunligi — belgining_boshlang'ich_o'rni=) pozitsiyadan boshlanadi;

agar "uzunligi" parametri musbat bo'lsa, satr berilgan uzunlik bo'yicha almashtiriladi;

agar “uzunligi” parametri manfiy bo’lsa, satr oxiridan boshlab nechta belgi almashtirish kerak emasligini ko’rsatadi yoki almashtirish (satr_uzunligi-uzunligi=) pozitsiyada tugatiladi;

agar “uzunligi” parametri ko’rsatilmasa, u holda almashtirish satr oxirigacha bajariladi.

9.33-misol. substr_replace() funksiyasining ishlatilishi

```
<?php
$text = "Mening ismim Komil.";
echo "Berilgan satr: $text<hr>\n";
/* quyidagi berilgan satr barcha berilgan satrlarni 'Meniki esa – Farhod'
satriga almashtiradi*/
echo substr_replace($text, 'Meniki esa – Farhod', 0) . "<br>\n";
echo substr_replace($text, 'Meniki esa – Farhod', 0, strlen($text)) .
"<br>\n";
// quyidagi satr berilgan satr boshiga 'Salom!' so'zini qo'shadi
echo substr_replace($text, 'Salom!', 0, 0) . "<br>\n";
// Quyidagi ikkita satr berilgan satrdagi Komil ismini
// Ilhom ismiga almashtiradi
echo substr_replace($text, 'Ilhom', 13, -1) . "<br>\n";
echo substr_replace($text, 'Ilhom', -6, -1) . "<br>\n"; ?>
```

Natija:

Berilgan satr: Mening ismim Komil.

Meniki esa – Farhod

Meniki esa – Farhod

Salom! Mening ismim Komil.

Mening ismim Ilhom.

Mening ismim Ilhom.

9.6. html-kodlardan tarkib topgan satrlar

html-teglaridan tarkib topgan satrlar bilan tez-tez ishlaymiz. Agar bunday satrni brauzerga echo() yoki print() yordamida chiqazsak, u holda biz html-teglarni o'zini ko'rmaymiz, balki shu teglarga mos formatlangan satrni olamiz. Brauzer barcha html-teglarini HTML tili standartlariga mos holda qayta ishlaydi. Ba'zan biz satrni qayta ishlanmagan holda brauzerda ko'rishni xohlaymiz. Buning uchun htmlspecialchars() funksiyasidan foydalaniladi.

htmlspecialchars() funksiyasi

htmlspecialchars (satr [, qo'shtirnoq_stili [, kodirovka]])

htmlspecialchars() funksiyasi "<", ">", "&", "\"", "'" kabi maxsus belgilarini HTML tilidagi mos "<", ">", "&", """, "'" belgilarga almashtiradi.

Qo'shimcha parametr qo'shtirnoq_stilida ikkitalik va bittalik qo'shtirnoqlarni qanday interpretatsiya qilinishi kerakligi ko'rsatiladi. qo'shtirnoq_stili uchta qiymat(ENT_COMPAT, ENT_QUOTES, ENT_NOQUOTES)dan biriga teng bo'lishi kerak. ENT_COMPAT konstantasi ikkitalik qo'shtirnoq maxsus belgisi sifatida tarjima qilinishini, bittalik qo'shtirnoq esa o'zgarishsiz qolishini anglatadi. ENT_QUOTES esa ham ikkitalik qo'shtirnoq, ham bittalik qo'shtirnoq o'zgarishi kerakligini, ENT_NOQUOTES esa barcha qo'shtirnoqlar o'zgarishsiz qolishi kerakligini bildiradi.

Kodirovkalar parametrida UTF-8, ISO-8859-1 va boshqa kodirovkalar berilishi mumkin, lekin birorta rus kodirovkalari bu yerda qo'llanilmaydi.

9.34-misol. htmlspecialchars() funksiyasining qo'llanilishi

```
<?php
$new = htmlspecialchars("<a href='mailto:au@mail.ru'> Xat yozish </a>", ENT_QUOTES);
echo $new;
/* berilgan satr quyidagicha kodirovkalanadi: &lt;a href=&#039;mailto:au@mail.ru&#039;&gt; Xat yozish &lt;/a&gt; */ ?>
Natija:<a href='mailto:au@mail.ru'> Xat yozish</a>
```

htmlspecialchars() funksiyasi faqatgina tez-tez ishlatiladigan maxsus belgilarni qayta kodirovkalaydi. Agar HTML tarkibidagi barcha belgilarni o'zgartirish zarur bo'lsa, u holda htmlentities() funksiyasidan foydalaniladi. Bu funksiyani ishlatishda ham ruscha harflar maxsus ketma-ketliklarga kodlanadi. Masalan, "A" harfi "À" kombinatsiyasiga almashadi. Bu funksiyaning sintaksisi va ishlatilishi qonun-qoidasi htmlspecialchars() funksiyasining sintaksisi va ishlatilish qonun-qoidasiga o'xshash.

9.7. Satrlar bilan ishlashga misollar

9.2-masala. Xatolarni tekshirish

Foydalanuvchi ma'lumotlarini qayta ishlash faqatgina ma'lumotlar to'g'ri strukturaga ega bo'lgandagina kutilgan natijani beradi. Kiri-

tilgan ma'lumotlarning ishonchliligini, aniqligini tekshirish mumkin emas, biroq ularni to'liqligini (masalan, elektron pochta manzili standart shablona mos kelishini aniqlash mumkin) tekshirish mumkin. Garchi ma'lumotlarni tekshirish uchun JavaScript texnologiyasidan tez-tez foydalanilsa ham, brauzerda mos kelmaslik kabi muammolar ham vujudga kelishi mumkin. Modomiki, PHP kodi server tomonidan bajarilar ekan, har doim forma ma'lumotlarini tekshirish kerakli natija (albatta, tuzilgan dasturning shartlari to'g'riligiga qarab) berishi mumkinligiga ishonch hosil qilish mumkin.

Dastur foydalanuvchi kiritgan ma'lumotlarida xato borligini bilishi bilan unga bu haqida xabar va uni tuzatishga taklif berishi lozim. Bu hollarda yechimning mumkin bo'lgan bir nechta variantlari mavjud — xatolik haqidagi xabarni oddiy chiqazish, alternativ variantlarni taklif etish (masalan, agar foydalanuvchi boshqa foydalanuvchi tomonidan tanlab bo'lingan loginni tanlasa) va h.k.

Formaning barcha maydonlari bo'sh qolmaganligini aniqlash uchun ketma-ket tekshirib chiqamiz. Kerak bo'lgan joyda kiritilgan ma'lumotlar strukturasi to'g'riligi ham tekshiriladi (masalan, elektron pochta manzili maydonida). Agar tekshirish muvaffaqiyatli yakunlansa, keyingi maydonga o'tiladi; aks holda dastur xatolik haqidagi xabarni chiqazadi, formani qayta ekranga chiqazish uchun keyinroq ishlatiladigan bayroqni o'rnatadi va keyingi maydonga o'tadi. Jarayon formaning barcha maydonlari tekshirib bo'lguncha davom etadi. (9.35-listing).

9.35-listing. Forma ma'lumotlarini tekshirish va xatolar haqida xabar chiqazish

```
<html>
<head> <title>Listing 9.35</title></head>
<body      bgcolor="#ffffff"      text="#000000"link="#cbda74"
vlink="#808040" alink="#808040">
  <? $form = "
  <form action=\"listing9_35.php\" method=\"post\">
  <input type=\"hidden\" name=\"seenform\"value=\"y\">
  <b>Ma'lumotlaringizni kiriting!</b><br>
  Ismingiz:<br> <input type=\"text\" name=\"name\" size=\"20\" max-
length=\"20\" value=\"\">
  <br> E-mailingiz:<br>
```



```



```

```

    print "$form";
    // foydalanuvchi formani to'ldirdi kiritilgan ma'lumotlarni tekshirish
    else : $error_flag = "n";
    // ism maydoniga ma'lumot kiritilganligini aniqlash
    echo $name;
    if ($name == "") :
    print "<font color='red'>* Ismingizni kiritmadingiz!</font> <br>";
    $error_flag = "y"; endif;
    // elektron pochta manzili maydoniga ma'lumot kiritilganligini aniqlash
    if ($email == "") :
    print "<font color='red'>* email manzilingizni kiritmadingiz !</font>
    <br>"; $error_flag = "y";
    else : $email = strtolower(trim($email));
    /* elektron pochta manzilidagi barcha alifbo belgilarini kichik harflarga
    almashtirish */
    // elektron pochta manzilini sintaksisi to'g'riligini tekshirish
    if (!@ereg('^[a-zA-Z0-9_]\|-\|\.|'+'.@'. '([a-zA-Z0-9_]\|-\|\.|'+'.[a-zA-Z]
    {2,4}$', $email)) :
    print "<br><font color='red'>* email manzilingizni noto'g'ri kiritdin-
    giz!</font> <br>";
    $error_flag = "y"; endif; endif;
    /* agar xato bayrog'i $error_flag o'rnatilgan bo'lsa, formani qayta chiqa-
    zish */
    if ($error_flag == "y") :
    print "$form";else : // foydalanuvchi ma'lumotlarini qayta ishlash
    print " Sizning ma'lumotlaringiz kiritildi!"; endif; endif;>
</body> </html>

```

9.35-listingdagi dastur ism maydoni va elektron pochta manzili bo'sh qolmaganligi hamda kiritilgan manzil sintaksisi to'g'riligini ko'rsatmoqda. Biror-bir tekshirish natijasida formada xatolik uchrasa, dastur mos xabarni beradi va formani qayta ekranga chiqazadi. Shu bilan birga foydalanuvchi xatolarni to'g'rilashi uchun formada avval kiritilgan ma'lumotlar saqlab qolinadi.

9.3-masala. Foydalanuvchi ma'lumotlarini matnli faylda saqlash.

Foydalanuvchiga ma'lumotlarini to'rtta obyektga kiritish taklif etiladi: ismi, elektron pochta manzili, tili va kasbi. Kiritilgan ma'lumotlar user_information.txt matnli faylda saqlanadi. Ma'lumot elementlari <vertikal chiziq> (|) belgisi bilan ajratiladi.

9.36-listing. Foydalanuvchi ma'lumotlarini matnli faylda saqlash

```
<html>
<head> <title>Bizga yozing! </title> </head>
<body bgcolor="#ffffff" text="#000000" link="#cbda74"
vlink="#808040" alink="#808040">
<? $form = " <form action=\"learn.php\" method=\"post\">
<input type=\"hidden\" name=\"seenform\" value=\"y\">
<b>Shaxsiy ma'lumotlaringizni kiriting!</fb><br>
Ismingiz:<br> <input type=\"text\" name=\"name\" size=\"20\" max-
length=\"20\" value=\"\"><br>
Emailingiz:<br> <input type=\"text\" name=\"email\" size=\"20\"
maxlength=\"20\" value=\"\"><br>
Tilingiz:<br>
<select name=\"language\">
<option value=\"\">Tilni tanlang:
<option value=\"English\">English
<option value=\"Spanish\">Spanish
<option value=\"Italian\">Italian
<option value=\"French\">French
<option value=\"Japanese\">Japanese
<option value=\"uzbek\">Uzbek
</select><br>
Kasbingiz:<br> <select name=\"job\">
<option value=\"\">Nima ish qilasiz?:
<option value=\"student\">Talaba
<option value=\"programmed\">Dasturchi
<option value=\"manager\">Menejer
<option value=\"teacher\">O'qituvchi </select><br>
<input type=\"submit\" value=\"submit!\">
</form>";// forma avval to'ldirilganmi?
if (!isset($seenform)) : $error_flag = "y"; endif;
```



```

// elektron pochta manzili maydoniga ma'lumot kiritilganligini aniqlash
if ($email == "") :
    print "<font color='red'>* email manzilingizni kiritmadingiz !</font>
<br>"; $error_flag = "y";
else : $email = strtolower(trim($email));
// elektron pochta manzilini sintaksisi to'g'riligini tekshirish
if (!@ereg('^([a-zA-Z0-9_]\|-\|\.)+\.'@'([a-zA-Z0-9_]\|-\|\.)+\.'[a-zA-Z]
{2,4}$', $email)) :
    print "<br><font color='red'>* email manzilingizni noto'g'ri kiritdin-
giz!</font> <br>";
    $error_flag = "y"; endif; endif;
/* agar xato bayrog'i $error_flag o'rnatilgan bo'lsa, formani qayta chiqa-
zish */
if ($error_flag == "y") :
    print "$form";else : // foydalanuvchi ma'lumotlarini qayta ishlash
    print " Sizning ma'lumotlaringiz kiritildi!"; endif; endif;?>
</body> </html>

```

Foydalanuvchi ism yoki elektron pochta manzilida <vertikal chiziq> (|) belgisini ishlatmaganligini tekshirish fragmentiga e'tibor qarating. str_replace() funksiyasi bu belgilarni o'chirib, ularni bo'sh satrga almashtiradi. Agar bu ish bajarilmasa, foydalanuvchi belgisi (|) ma'lumotlar fayli strukturasini buzadi va uni to'g'ri qayta ishlashiga halaqit beradi.

Birmuncha kichik hajmdagi ma'lumotlarni qayta ishlash jarayonida matnli fayllardan to'la foydalanish mumkin. Biroq ko'p sonli foydalanuvchi ma'lumotlari bilan ishlash (saqlash va qayta ishlash) uchun ma'lumotlar bazasidan foydalanilgan ma'qul.

2-qism. MA'LUMOTLAR BAZASI BILAN ISHLASH

X bob. MA'LUMOTLAR BAZASI

Bu bobda ma'lumotlar bazasi va MBBT tushunchalari ko'riladi. SQL asosiy so'rov tili: satrlarni saralash, qo'shish, o'zgartirish va o'chirish amallari hamda jadvallarni yaratish, o'zgartirish va o'chirish amallari ko'riladi. Ma'lumotlar bazasi nazariyasining asosiy tushunchalari, MySql ma'lumotlar bazasini boshqarish tizimi, u bilan ishlash usullari va bu MBBT da SQL so'rov tilini amalga oshirish ko'rib chiqiladi.

10.1. Ma'lumotlar bazasi haqida asosiy tushunchalar

Ma'lumotlar bazasi o'zi nima? Ma'lumotlar bazasi o'zida strukturalashgan ma'lumotlarning tizimini saqlaydi. Bu ma'lumotlar turli xil, ya'ni oddiy sotib olinishi kerak bo'lgan narsalar ro'yxatidan tortib to rasmlar galereyasidagi rasmlar ro'yxati, biror tashkilot xodimlarining ma'lumotlari yoki bo'lmasa korporativ tarmoqdagi katta hajmdagi ma'lumotlar bo'lishi mumkin.

Ma'lumotlar bazasini yaratish, ularga ma'lumotlarni kiritish va mavjud ma'lumotlar bazasidan foydalanish, ya'ni saralash, ishlov berish, saqlash, qidirish va qabul qilish kabi vazifalarni bajaruvchi tizim ma'lumotlar bazasini boshqarish tizimi (MBBT) deb yuritiladi.

Relyatsion ma'lumotlar bazasi — bu relyatsion algebrani mustahkam nazariy bazisiga asoslangan, kuchli nazariy fundamentga qurilgan, obyektlar va ularning o'zaro aloqalari ikki o'lchovli jadval ko'rinishida tasvirlangan ma'lumotlar bazasi. Ma'lumotlarning bunday ko'rinishda tasvirlanishi obyektlarning o'zaro aloqalari yaqqol tasvirlanishiga asos bo'ladi. Relyatsion ma'lumotlar bazasi bugungi kunda eng ko'p foydalanadigan baza hisoblanadi. Relyatsion ma'lumotlar bazasida ma'lumotlar bir to'da bo'lgan holda emas, balki alohida jadvallarda joylashadi. Va buning hisobiga tezkorlik bilan egiluvchanlik xususiyatlariga ega.

Shuni ta'kidlash lozimki, hozirgi vaqtda deyarli barcha MBBT-lari, asosan, relyatsion modellar asosida tashkil qilinmoqda.

Quyidagi jadvallardan tashkil topgan o'quv ma'lumotlar bazasi loyihasini tuzib chiqamiz.

student (talaba)

stud_id	surname	name-name	stipend	kurs	city	birthday	univ_id
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	10
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	10
6	Salimov	Vohid	150	4	Namangan	7/06/1989	22
10	Karimov	Botir	0	2	Jizzax	8/12/1991	10
12	Zoitova	Ozoda	250	2	Termiz	1/05/1991	10
265	Fayziyev	Alijon	0	3	Toshkent	5/11/1989	10
32	Komilov	Farhod	150	5	Samarqand	NULL	14
654	Lazizov	Anvar	200	3	Toshkent	1/12/1991	10
276	Fozilov	Ahmad	200	4	NULL	5/08/1991	22
55	Boltayeva	Vasila	250	5	Toshkent	7/01/1990	10
...							
	...				...		

stud_id – sonli kod, talaba identifikatori

surname – talabaning familiyasi

name – talabaning ismi

stipend – talaba oladigan stipendiyasi

kurs – talaba o'qiydigan kursi

city – talabaning yashash manzili (viloyati)

birthday – talabaning tug'ilgan sanasi

univ_id – sonli kod, talaba o'qiydigan universitet identifikatori

lecturer (o'qituvchi)

lecturer_id	Surname	name	city	univ_id
24	Qobulov	Botir	Toshkent	10
46	Nasriddinov	Islom	Toshkent	10
74	Hasanov	Farhod	Namangan	22
108	Sobirov	Nizom	Jizzax	22
276	Nosirova	Vasila	Toshkent	10
328	Saidazimov	Akrom	Andijon	10
....				

lecturer_id – sonli kod, o'qituvchining identifikatori

surname – o'qituvchining familiyasi

name – o'qituvchining ismi

city – o'qituvchining yashash manzili (viloyati)

univ_id – o'qituvchi ishlaydigan universitet identifikatori

10.3-jadval

subject (o'quv fani)

subj_id	subj_name	hour	semestr
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingliz tili	56	3
73	Mehnat	34	5
....			

subj_id – o'quv fani identifikatori

subj_name – o'quv fani nomi

hour – fanning o'quv soati

semestr – fan o'tiladigan semestr

10.4-jadval

university (universitet)

univ_id	univ_name	rating	city
22	Texnika davlat universiteti	34	Toshkent
10	Toshkent davlat pedagogika universiteti	29	Toshkent
11	Andijon davlat universiteti	49	Andijon
32	Termiz davlat universiteti	41	Termiz
14	Samarqand davlat universiteti	36	Samarqand
15	O'zbekiston Milliy universiteti	28	Toshkent
18	Toshkent davlat axborot universiteti	36	Toshkent
....			

univ_id – universitet identifikatori

univ_name – universitet nomi

rating – universitet reytingi

city – universitet joylashgan viloyat yoki shahar

10.5-jadval

exam_marks (imtihon bahosi)

exam_id	stud_id	subj_id	mark	exam_date
145	12	10	5	12/01/2010
34	32	10	4	23/01/2010
75	55	10	5	05/01/2010
238	12	22	3	17/06/2009
639	55	22	NULL	22/06/2009
43	6	22	4	18/01/2010
....				

exam_id – imtihon identifikatori
stud_id – talabani identifikatori
subj_id – o'quv fani identifikatori
mark – imtihon bahosi
exam_date – imtihon sanasi

10.6-jadval

subj_lect (o'qituvchining o'quv dissiplinasi)

lecturer_id	subj_id
24	24
46	46
74	74
108	108
276	276
328	328
....	

lecturer_id – o'qituvchining identifikatori
subj_id – o'quv fani identifikatori

Jadvallar. Relyatsion modellar asosiga qurilgan ma'lumotlar bazasi jadval ko'rinishida aks etadi. Bunday jadvaldagi ustunlar maydon deb, satrlar esa yozuv deb ataladi. Jadval nomi – student (talabalar), har birida aniq ma'lumotlar joylashgan bir qancha ustunlar va talaba haqidagi ma'lumot joylashgan satrlardan iborat.

Masalan, 10.1-jadval.

Ustun – ma'lumotlarni tashkil etishning oddiy birligi bo'lib, ma'lumotning alohida, bo'linmas birligiga egalik rekvizit mos keladi.

Demak, maydon MBning asosiy tuzilmali elementi bo'lib, quyidagi parametrlar bilan ifodalanadi:

– uzunligi (belgi va simvollarida ifodalanib, baytlarda o'lchanadi),

– nomi (maydonning o'ziga xos alohida xususiyati);

– imzo (podpis) (ustun sarlavhasi haqida ma'lumot).

Ustunlar

Jadvaldagi har bir maydon (ustun) o'zining unikal nomiga ega va turli xil ma'lumotlar qabul qiladi. Ustunlarni ba'zan maydonlar deb ham yuritiladi. Har bir maydon turlicha tipga ega.

Jadvallar orasidagi munosabatlar ishonchli ishlashi va bir jadvaldagi yozuv orqali ikkinchi jadvaldagi yozuvni topish uchun jadvalda alohida maydon-unikal maydon bo'lishini ta'minlash kerak.

Unikal maydon – bu qiymatlari takrorlanmaydigan maydondir.

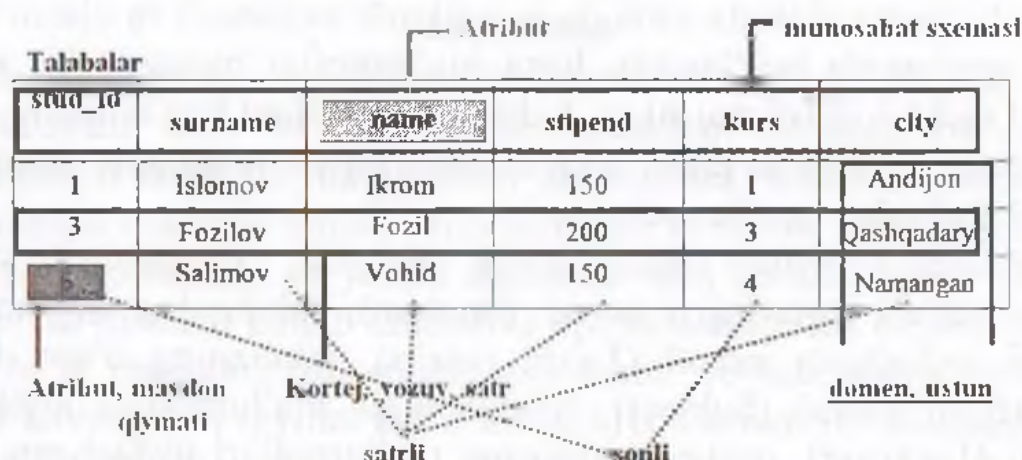
Satrlar

Satr mantiqiy bog'langan rekvizitlarga mos keluvchi maydonlar yig'indisidir. Satrning tuzilishi o'z tarkibiga mos har bir oddiy ma'lumotga ega maydonlar tarkibi va ketma-ketligi bilan belgilanadi.

Har bir satr alohida bitta shaxs yoki narsa haqida bo'ladi. Har bir satrning formati va atributi bir xil. Satrlar kortejlar yoki yozuv (qator) lar ham deb ataladi.

Qiymatlar

Har bir satr ustunga bog'liq holda bir qancha qiymatlarni o'z ichiga oladi. Har bir qiymatning tipi ustunga berilgan tipga mos bo'lishi kerak.



10.1-rasm. Jadval strukturasi

Kalitlar

Ba'zan jadvallardagi har bir satrni bir-biridan ajratib olishga to'g'ri keladi. Aynan shuning uchun kalitdan foydalaniladi. Masalan, talabalar haqida ma'lumotlar kiritilgan bir qancha satrlardan iborat. Talaba jadvalidan TDPU da tahsil oluvchi Axmedov Sadridin familiya va ismli talaba haqida ma'lumot olmoqchi bo'lsak, bu jadvalda boshqa yana bir xuddi shunday ism va familiyali talaba haqida ma'lumot bo'lishi mumkin. Ularni bir-biridan talaba kodini kiritish orqali farqlay olamiz. Buni amalga oshirish uchun, ya'ni maydondagi qiymatlarni bir-biridan farqlab olish uchun shu maydonga (talaba kodi maydoniga) kalit qo'yamiz. Relyatsion ma'lumotlar bazasida bunday kalitni birlamchi kalit yoki bosh kalit (Primary key) deb ataladi. Bosh kalit qo'yiladigan maydon (ustun) ikkita xossaga ega bo'lishi kerak:

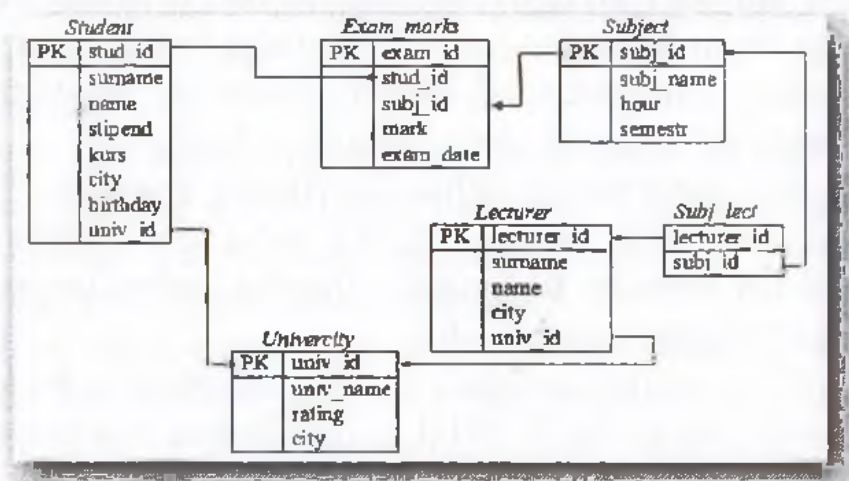
1. Ustundagi yozuvning yagonaligi. Kalitdagi qiymat o'zgarmasligi lozim.

2. Ko'p ma'nolikka yo'l qo'ymaslik. Ustundagi birorta belgini o'zgartirish mumkin emas, aks holda bosh kalit o'zgaradi.

Bundan tashqari bir nechta jadvallarni ham kalit yordamida bog'lash mumkin. Masalan, talaba haqidagi ma'lumotlar (shaxsiy, imtihon baholari, o'quv fanlari, o'qituvchilari, universiteti) ni jadvalga joylashtirish kerak bo'lsin. Agar bu ma'lumotlar bitta jadvalga joylashtirilsa, talabaning imtihon bahosini bilish uchun ham barcha ma'lumotlarga murojaat etishga to'g'ri keladi. Bu esa ishlash tezligini pasaytiradi. Shuning uchun talaba haqidagi ma'lumotlar kategoriyalari bo'yicha alohida-alohida jadvallarda saqlanadi va ularni talaba kodi yordamida bog'lanadi. Bitta ma'lumotlar bazasida bir nechta jadval tashkil qilish mumkin, kalit ular orasidagi bog'lanishni yaratib beradi. Relyatsion jadvallar atamalari bo'yicha bunday munosabat tashqi kalit deb nomlanadi.

Masalan, bizda oltita jadval mavjud: talabaning shaxsiy ma'lumotlari joylashgan jadval (Student), talabaning imtihon baholari joylashgan jadval (Exam_marks), talabaning o'quv fanlari joylashgan jadval (Subject), o'qituvchilar ma'lumotlari joylashgan jadval (Lecturer), universitetlarning ma'lumotlari joylashgan jadval (Univercity) va o'qituvchilar o'qitadigan fanlari joylashgan jadval (Subj_lect). Ushbu barcha jadvallarda birlamchi kalit mos ravish-

da stud_id (Student jadvalida), exam_id (Exam_marks jadvalida), subj_id (Subject jadvalida), univ_id (University jadvalida), lecturer_id (Lecturer jadvalida) (10.2-rasm). Exam_marks jadvalida Student jadvalidagi stud_id maydonidagi qiymatga teng identifikator yoziladigan stud_id maydoni mavjud, ushbu maydon Exam_marks jadvali uchun tashqi kalit, Student jadvali uchun esa birlamchi kalit hisoblanadi. Xuddi shunday subj_id Exam_marks jadvali uchun tashqi kalit, Subject jadvali uchun esa birlamchi kalitdir. Subj_lect jadvalida esa birlamchi kalit ishtirok etmagan, bu jadvalda faqatgina tashqi kalitlar mavjud. Tashqi kalitlar ma'lumotlar bazasidagi jadvallar o'rtasida aloqa tashkil etish uchun ishlatiladi.



10.2-rasm. Jadvallar o'rtasida aloqa o'rnatish uchun kalitlarni ishlatish.

Indekslash

Ma'lumotlar bazasi bilan ishlashda yuzaga keladigan asosiy masalalardan biri bu — qidirish masalasi. Shunga ko'ra, agar ma'lumotlar bazasida ma'lumotlar ko'p bo'lsa, dasturchi oldida oddiy qidirish masalasi emas, balki samarali qidirish, ya'ni qidiruv uncha vaqtni olmasligi va yetarli darajada aniq bo'lishi kerak. Buning uchun (so'rovlarni ishlab chiqishni optimallashtirish uchun) jadvalning ba'zi maydonlarini indekslash kerak bo'ladi. Bitta ustundagi satrni ko'rsatilgan qiymat bilan tez qidirish uchun indeksni ishlatish foydali. Indeks berilmaganda, jadvalni o'qish barcha jadvalning birinchi yozuvidan boshlab toki mos satr uchraguncha amalga oshiriladi. Jadvallar qancha ko'p bo'lsa, shuncha qo'shimcha vaqt kerak

bo'ladi. Agar jadvalning qaralayotgan ustunida indeks mavjud bo'lsa, u holda ma'lumotlar bazasi qidirish uchun barcha ma'lumotlarni ko'rib chiqmasdan, ma'lumotlar fayli oralig'i o'rnini tezda aniqlaydi. Ma'lumotlar bazasi qiymatlarni tezroq qidirishni amalga oshirishi uchun, indekslangan maydonlarni xotiraga yaqin joyga joylashtiradi. 1000 ta satrdan tarkib topgan jadvallar uchun bu barcha yozuvlarni ketma-ket saralash bilan taqqoslaganda kamida 100 marotaba tezroq amalga oshadi. Biroq ba'zi hollarda barcha 1000 ta satr uchun ruxsat zarur bo'lganda, ketma-ket o'qish tezroq amalga oshadi, shuning uchun diskdan qidirish amali talab qilinmaydi. Shunga ko'ra ba'zan indekslar faqatgina to'siq bo'ladi. Masalan, agar katta hajmdagi ma'lumotlar jadvalga ko'chirilayotgan bo'lsa, u holda hech qanday indeksga ega bo'lmagani ma'qul. Biroq ba'zi hollarda bir vaqtda bir qancha indeksni ishlatish talab etiladi (masalan, tez-tez ishlatiladigan jadvallarda so'rovlarni qayta ishlash uchun).

MySQL da indeksning uchta ko'rinishi mavjud: PRIMARY, UNIQUE va INDEX. Kalit (KEY) so'zi esa indeks (INDEX) so'ziga sinonim sifatida ishlatiladi. Barcha indekslar xotirada daraxtsimon ko'rinishida saqlanadi.

PRIMARY — unikal indeksli (kalitli) maydonlar bo'sh qiymatni qabul qilmaydi (ya'ni NOT NULL ni). Jadval faqatgina bitta birinchi darajali indeksga ega bo'ladi, lekin bir nechta maydonlardan iborat bo'lishi mumkin.

UNIQUE — indeks (kalit), berilgan maydon faqatgina unikal qiymatni qabul qila oladi.

INDEX — oddiy indeks.

MySQL da bundan tashqari satrli maydonlarni satr boshlanishidan boshlab berilgan sondagi belgilarni indekslash mumkin.

Sxemalar

Ma'lumotlar bazasi jadvallari loyihalarining to'liq to'plami ma'lumotlar bazasining sxemasi (tarxi) deyiladi. Sxema qay bir ma'noda chizma (chertyoj) ga o'xshashdir. Sxema jadvallarni ustunlari bilan birga, ustunlarning ma'lumot turlarini, har bir jadval uchun birlamchi kalitni va barcha tashqi kalitlarni ko'rsatib turishi kerak. Sxema o'z ichiga, masalan, qaysi narsa qay maqsadda kerakligini ko'rsatuvchi ma'lum bir namunaviy ma'lumotlarni olmaydi. Sxemalar biz foydalanadigan "mavjudlik-munosabatlar" singari dia-

grammalar ko'rinishida yoki matn ko'rinishida tasvirlanishi mumkin. Masalan, o'quv ma'lumotlar bazasining sxemasini tuzamiz:

student(*stud_id*, surname, name, stipend, kurs, city, birthday, *univ_id*)

exam_marks(*exam_id*, *stud_id*, *subj_id*, mark, exam_date)

subject(*subj_id*, subj_name, hour, semestr)

lecturer(*lecturer_id*, surname, name, city, *univ_id*)

subj_lect(*lecturer_id*, *subj_id*)

university(*univ_id*, univ_name, rating, city)

Sxemadagi belgilab qo'yilgan elementlar, boshqa belgilanganlar bilan munosabatlarning birlamchi kaliti hisoblanadi. Chiziq bilan belgilangan elementlar — boshqa jadvaldagi belgilangan munosabatlardan uchun tashqi kalitlardir.

Munosabatlar

Relyatsion ma'lumotlar bazasida uch turli asosiy munosabatlar mavjud. Ular munosabatning har qaysi tomonida qatnashuvchi elementlar soniga qarab tasniflanadi. Demak, «birga-bir», «bitta-ko'pga» va «ko'p-ko'pga» munosabatlari:

«Birga-bir» munosabati bo'yicha, munosabatda har bir elementdan bittadan qatnashadi. Birga-bir munosabatining ma'nosi shuki, bitta jadvalning bitta satri boshqa jadvaldagi faqat bitta satrga to'g'ri keladi (mos keladi). Misol uchun, agar ikkita jadvalni ko'radigan bo'lsak, ulardan birida talabalar to'g'risida ma'lumotlar, ikkinchisida esa ta'lim muassasalari to'g'risidagi ma'lumotlar bo'lsa, u holda bu jadvallar orasida birga-bir munosabat mavjud, chunki bitta jadvalda ma'lumoti bo'lgan talaba uchun ta'lim muassasalari to'g'risidagi ma'lumotlar bo'lgan ikkinchi jadvaldan faqat bitta satr to'g'ri kelishi mumkin.

Hammadan ko'p uchraydigan ma'lumotlar bazasidagi munosabatlar turi bu bitta-ko'pga munosabati. «Bitta-ko'pga» munosabatida birinchi jadvalning bitta satri ikkinchi jadvalning bir nechta satriga bog'lanishi mumkin. Berilgan munosabatlar turini ko'rsatish uchun, o'qituvchilar va ular dars beradigan fanlar to'g'risida ma'lumotlar bo'lgan jadvallarga murojaat etishimiz mumkin. Boshqa misollar sifatida korxona va unda ishlaydigan ishchilar orasidagi munosabatlar ko'rilishi mumkin. Xuddi shunday munosabatlar kompyuter va unga kiruvchi komponentlar orasida mavjud va h.k.

Agar «ko'p-ko'pga» munosabati mavjud bo'lsa, demak, bir jadvalning bir nechta satri ikkinchi jadvalning bir nechta satriga bog'langan bo'ladi. Ikkita jadvallar orasida ko'p-ko'pga munosabati hosil bo'ladi, qachonki: birinchi jadvaldagi bitta satr ikkinchi jadvaldagi bittadan ortiqroq satr bilan bog'lanishi mumkin bo'lsa; ikkinchi jadvaldagi bitta satr birinchi jadvaldagi bittadan ortiqroq satr bilan bog'lanishi mumkin bo'lsa.

Misol uchun talabalar portalining o'quv ma'lumotlar bazasiga murojaat etamiz. Ikki guruh obyektlarni ko'rib chiqamiz: talabalar ro'yxati va fanlar bo'yicha topshirilgan imtihonlar ro'yxati. Bu ma'lumotlarni o'z ichiga olgan jadvallar orasida «ko'p-ko'pga» munosabati mavjud, chunki har qaysi talaba bittadan ortiq fandan imtihon topshirgan. Xuddi shunday, har qaysi fandan bittadan talaba imtihon topshirgan bo'lishi mumkin.

Jadvallarning turlari

Ma'lumotlar bazasi qoida bo'yicha quyidagi ikki turli jadvallardan tashkil topadi:

- Real obyektlar haqida ma'lumot saqlovchi oddiy jadvallar, ular boshqa sodda obyektlar bilan «birga-bir» va «bitta-ko'pga» munosabatlariga kirisha oluvchi kalitlarga ega bo'ladi.
- Ikkita haqiqiy obyekt orasidagi «ko'p-ko'pga» munosabatlarini aks ettiruvchi bog'lovchi jadvallar. Bunday jadvallar qisman ba'zi bir haqiqiy o'tkazishlar hisobiga tashkillashtiriladi.

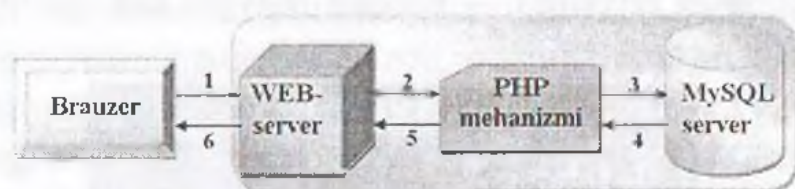
Ko'p miqdordagi bo'sh atributlar (katakalar) dan voz kechish.

Ma'lumotlarda ko'p miqdordagi bo'sh qiymatlarning mavjudligi — amaliy jihatdan yomon holdir. Bu qattiq diskdagi mavjud joydan maqsadli foydalanishda va ustundagi barcha qiymatlar yig'indisi va shu kabi ifodalarni hisoblashda bir qator muammolarga olib keladi. Foydalanuvchi jadvalda bo'sh qiymatni ko'rar ekan, u buni mos bo'lmagan qiymatmi yoki ma'lumotlar bazasidagi xatolikmi, yoki bo'lmasa, shunchaki ma'lumot kiritilmaganmi, qaysi biri ekanligini bilmaydi. Ma'lumotlar bazasini loyihalashda eng maqbul yo'lni tutib, bo'sh qiymatlar bilan bog'liq ko'pgina muammolardan qutulish kerak.

10.2. WEB ma'lumotlar bazasini tashkil etish

WEB ma'lumotlar bazasining ichki arxitekturasini va ishlash prinsipini ko'rib o'tamiz.

WEB ma'lumotlar bazasini bazali arxitekturasini o'zida WEB brauzer, WEB server, ishlovchi mexanizm va ma'lumotlar bazasi serverini jamlaydi, ya'ni ular orasida quyidagicha bog'lanish mavjud.



10.3-rasm. WEB ma'lumotlar bazasining bazali arxitekturasini o'zida WEB brauzer, WEB server, mexanizm sahifalari va ma'lumotlar bazasini jamlaydi.

1. WEB sahifa orqali aniqlangan HTTP so'rovlarni WEB brauzer orqali jo'natadi.

2. WEB server so'rovni qabul qilib, PHP mexanizmiga qayta ishlash uchun jo'natadi.

3. PHP mexanizmi sahifani sintaktik analiz qilishni boshlaydi.

4. MySQL serveri ma'lumotlar bazasiga kelayotgan so'rovlarni qabul qiladi, ularni qayta ishlaydi, undan keyin natijalarni jo'natadi. Ushbu holda ma'lumotlarni PHP mexanizmiga uzatadi.

5. PHP mexanizmi so'rov natijalarini HTML ko'rinishiga o'tkazib, ishni oxiriga yetkazadi, shundan so'ng natijalarni HTML shaklida WEB serverga o'tkazadi.

6. WEB server kerakli ma'lumotlarni foydalanuvchi ko'ra oladigan qilib, HTML ni brauzerga chiqaradi.

10.3. MySQL ma'lumotlar bazasini boshqarish tizimi (MBBT)

MySQL — bu relyatsion ma'lumotlar bazasini boshqarish tizimi. MySQL — ochiq kodli tizim. Bundan tashqari, tezkor, ishonchli va asosiysi MBBT ishlatish uchun sodda bo'lgan tizim. Bu interfeys MySQL da ma'lumotlar bazasi bilan ishlashni anchagina soddalashtiradi.

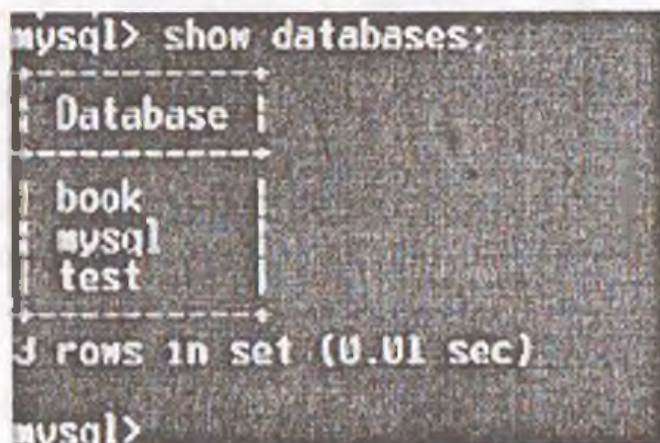
Matnli rejimda ma'lumotlar bazasi bilan ishlash xuddi buyruqlar satrida buyruqlarni kiritish kabi bo'ladi (10.4-rasm), uning natijasi esa alohida jadval shaklida chiqaziladi. (10.5-rasm).

10.1-misol.

```
mysql> show databases;
```

10.4-rasm. Buyruqlar satrida MySQL bilan ishlash.

show databases buyrug'i — barcha mavjud ma'lumotlar bazasini chiqazadi



```
mysql> show databases;
+-----+
| Database |
+-----+
| book     |
| mysql    |
| test     |
+-----+
3 rows in set (0.01 sec)

mysql>
```

10.5-rasm. show databases buyrug'ining natijasi.

MySQL ni o'rnatish va uni ishga tayyorlash oldingi bo'limlarda ko'rib o'tilgan. MySQL ni o'rnatgandan so'ng mysql.exe, mysqld.exe va h.k. fayllari joylashgan katalogga (Windows XP da c:\mysql\bin) kiriladi. mysqld.exe fayli MySQL tizimini ishga tushiradi. MySQL tizimi bir qancha sistemalarda servis sifatida ishga tushadi. Tizim ishga tushgandan so'ng, mysql.exe dasturi yuklanadi. Bu yerda hatto parol ham so'ramaydi. Shundan so'ng, buyruqlar satrida

10.2-misol.

```
shell> mysql.exe -u root
yoki
shell>mysql -u root mysql
```

buyruqlarini yozib, MySQL tizimining barcha administrator huquqlarini olish mumkin.

Birinchidan, ba'zi bir kamchiliklarni to'g'rilaymiz (administrator parolining yo'qligi, boshqa foydalanuvchilar kirishiga imkon yarata-di, shuning uchun unga parol qo'yamiz):

10.3-misol.

```
shell> mysql -u root mysql
mysql> UPDATE user SET Password=PASSWORD('new_password')
WHERE user='root';
mysql> DELETE FROM user WHERE user='';
mysql> FLUSH PRIVILEGES;
```

MySQL foydalanuvchilari haqidagi barcha ma'lumotlar maxsus ma'lumotlar bazasi mysql ning user jadvalida saqlanadi, unga kirish huquqiga faqat tizim administratori ega. Shuning uchun, biror-bir parolni o'zgartirish uchun ushbu jadvalni o'zgartiriladi. Parol sifatida kiritilgan ma'lumotlar shifrlanishi uchun PASSWORD funksiyasi ishlatiladi. Administrator paroli o'zgartirishdan tashqari, barcha foydalanuvchi loginlarini (DELETE buyrug'i) o'chirish kerak. Flush Privileges buyrug'i ma'lumotlar bazasini boshqarish tizimini (MySQL) yangilash uchun imkon yaratadi.

Endi loyihada ko'rilgan talabalar portalining o'quv ma'lumotlar bazasini yaratamiz (biz hali ham tizim administratori sifatida ishlayapmiz):

10.4-misol.

```
mysql>create database talaba;
```

Unix tizimida ma'lumotlar bazasining nomi registrga sezgir, shuning uchun bu tizimda ma'lumotlar bazasiga faqatgina talaba nomi bilan murojaat etish mumkin, Talaba, TALABA yoki yana boshqa nomlar bilan emas. Bu jadvallar nomi uchun ham tegishli. Windows tizimida esa bu chegara qo'llanilmaydi. Unda talaba~Talaba~TALABA nomlar teng kuchli.

Ma'lumotlar bazasini yaratish, undan foydalanish uchun ruxsat bermaydi. talaba ma'lumotlar bazasidan foydalanish uchun aktiv holatga o'tkazish kerak, buning uchun ushbu buyruqni beramiz:

10.5-misol.

```
mysql> USE talaba;
```

Database changed

mysql seansi boshlangandan so'ng, ma'lumotlar bazasi faqatgina bir marotaba yaratiladi, lekin ushbu bazadan bir necha marotaba foydalanish mumkin.

MySQL ning barcha buyruqlar satri nuqta vergul bilan tugatiladi. Agar nuqta vergul ishlatilmagan bo'lsa, u qo'yilmaguncha berilgan buyruq satri buyruqlari bajarilmaydi.

10.6-misol.

```
mysql> show tables
```

```
->
```

```
->;
```


Endi oddiy foydalanuvchi yaratamiz va unga yaratilgan ma'lumotlar bazasida ishlash uchun vakolat beramiz. Buning uchun GRANT buyrug'idan foydalanamiz.

GRANT — foydalanuvchiga ma'lumotlar bazasida ishlash uchun vakolatlarni (imtiyozlar, privilegiya) beradi.

MySQL da 4 ta darajadagi vakolatlar mavjud:

1. Global.
2. Ma'lumotlar bazasi.
3. Jadvallar.
4. Ustunlar.

Sintaksisi:

```
GRANT (ALL PRIVILEGES | (SELECT, INSERT, UPDATE,
DELETE, REFERENCES (column list), USAGE)) ON ITEM TO
user_name[identifiy field by 'password'] [WITH GRANT OPTION]
```

PRIVILEGES (vakolatlar) bir-biridan vergul orqali ajratib yoziladi.

column list (ustunlar ro'yxati) vakolatlar berilayotgan ustunni aniqlash uchun ishlatiladi. Ustunlar soni bittadan ko'p bo'lsa, vergul orqali ajratiladi.

ITEM (element) o'rnida yangi vakolat berilayotgan ma'lumotlar bazasi yoki jadval beriladi. Agar ITEM o'rniga (*.*) ishlatilsa, vakolatlar barcha ma'lumotlar bazasi uchun o'rnatiladi. Bunday amal, ya'ni vakolatlarni barcha ma'lumotlar bazasi uchun o'rnatish global vakolatlar deb ataladi. (*.*) o'rnida (*) amalini ham ishlatish mumkin.

Agar ITEM dbname.*(ma'lumotlar_bazasi_nomi.*) kabi berilsa, ya'ni vakolatlar berilgan ma'lumotlar bazasining barcha jadvallari uchun o'rnatilsa, bu ma'lumotlar bazasi vakolatlari deyiladi.

Agar ITEM dbname.table_name (ma'lumotlar_bazasi_nomi.jadval_nomi) kabi berilsa, demak vakolatlar ko'rsatilgan jadval uchun beriladi va bunday vakolatlar jadvallar vakolati deyiladi.

Agar ITEM o'rnida dbname.table_name va column list da berilgan zarur ustunlar ro'yxati berilsa, demak vakolatlar ko'rsatilgan ustunlar uchun beriladi va bunday vakolatlar ustunlar vakolati deyiladi.

user_name o'rnida foydalanuvchining MySQL ga kirish uchun kerak bo'lgan nomi beriladi. Faqatgina bu nom tizimda mavjud

nom bilan bir xil bo'lmashligi lozim. Foydalanuvchilarni ajratishga qulay bo'lishligi uchun uning nomiga xost nomi qo'shib yozilsa ham bo'ladi, masalan, lola (lola@localhost kabi) va lola@somewhere.com. Ko'pgina foydalanuvchilar turli domenlarda bir xil nomga ega bo'lishadi. Foydalanuvchi nomiga xost nomining qo'shib berilishi aynan shu holatlarda qo'l keladi.

password parametrada tizimga kirish uchun parol beriladi.

Agar with grant options opsiyasi ko'rsatilgan bo'lsa, bu foydalanuvchiga o'z vakolatlarini boshqa foydalanuvchiga uzatish imkonini beradi.

Vakolatlar mysql nomli ma'lumotlar bazasining to'rtta tizimli jadvallarida saqlanadi. Bu jadvallar: mysql.user, mysql.db, mysql.tables_priv, mysql.columns_priv kabi nomlanadi va ular yuqorida ko'rilgan to'rtta darajadagi vakolatlarga tegishli.

```
mysql> grant all
on *
to lola@localhost
identified by '135'
-> with grant options;
```

lola foydalanuvchisi 135 paroli bilan barcha ma'lumotlar bazasiga vakolat va shu vakolatlarni uzatish huquqini oladi.

MySQL da uchta asosiy vakolatlar turi mavjud. Ular:

1. Oddiy foydalanuvchiga berilishi mumkin bo'lgan vakolatlar.
2. Faqat administratorlarga tegishli bo'lgan vakolatlar.
3. Maxsus vakolatlar to'plami.

1. Oddiy foydalanuvchiga berilishi mumkin bo'lgan vakolatlar:

select (jadval va ustunlar uchun) — satrlarni tanlashda yordam beradi;

insert (jadval va ustun uchun) — jadvalga yangi satrlar(yozuvlar) qo'shish uchun ishlatiladi;

update (jadval va ustun uchun) — satrlardagi ma'lumotlarni o'zgartirish uchun ishlatiladi;

delete (jadval uchun) — satrlarni o'chiradi;

index (jadval uchun) — indekslar yaratish va o'chirish uchun ishlatiladi;

alter (jadval uchun) — jadvalga o'zgartirish kiritish uchun ishlatiladi;

create (ma'lumotlar bazasi va jadvallar uchun) — yangi ma'lumotlar bazasi yoki jadvallar yaratish uchun ishlatiladi;

drop (ma'lumotlar va jadval uchun) — ma'lumotlar yoki jadval-larni o'chirishda ishlatiladi.

2. Administrator uchun vakolatlar:

Reload — vakolatlar jadvallarini ishga tushirishda va vakolat jur-nalli fayllar, jadvallarga qo'shimcha kiritishda foydalaniladi;

Shutdown — Server MySQL ni to'xtatish uchun ishlatiladi;

PROCESS — tizimdagi jarayonlarni boshqarish va ularni o'chirish uchun foydalaniladi;

File — jadvallardagi ma'lumotlarni faylga va aksincha fayldagi ma'lumotlarni jadvalga joylashtiradi.

3. Maxsus vakolatlar to'plami:

ALL — yuqoridagi barcha vakolatlarni (oddiy foydalanuvchi va administratorga berilgan vakolatlarni) o'z ichiga oladi. ALLPRIVI-LEGES va ALL ga teng kuchli.

USAGE — foydalanuvchiga sistemaga kirishi uchun imkon be-radi. Lekin ruxsatsiz hech qanday amal bajarib bo'lmaydi.

Endi yangi foydalanuvchi yaratamiz.

10.7-misol.

```
mysql> grant all privileges on talaba.*  
to ilyosbey@localhost identified by '123';
```

Ilyosbey foydalanuvchisi 123 paroli bilan talaba ma'lumotlar ba-zasida ishlash uchun ruxsat huquqini oladi.

REVOKE — grant ga qarama-qarshi amallarni bajaradi. U foy-dalanuvchidan vakolatlarni tortib oladi.

Sintaksisi:

REVOKE privileges [(column)] on item from user_name;

yoki

Mysql > REVOKE grant options on item from user_name;

foydalanuvchidan boshqa foydalanuvchiga beriladigan vakolat-larni tortib oladi.

10.8-misol.

```
Mysql > REVOKE all on * from ilyosbey;
```

ilyosbey foydalanuvchisidan barcha vakolatlarni tortib oladi.

10.9-misol.

```
Mysql > REVOKE alter, create, drop on talaba, * from ilyosbey;
```

ilyosbey foydalanuvchisidagi vakolatlarni cheklab qo'yadi, lekin tortib olmaydi.

Endi lola nomli foydalanuvchi sifatida mos parol bilan tizimga kiramiz:

10.10-misol.

```
shell>mysql -u lola -p
Enter password: ***
Welcome to the MySQL monitor!...
mysql>
```

10.4. MySQL MBBT ning asosiy operatorlari

Ma'lumotlar bazasi bilan quyidagi amallarni bajarish mumkin:

1. Ma'lumotlar bazasini (MB) yaratish, o'chirish va o'zgartirish.
2. Foydalanuvchilarga vakolat berish, o'chirish, o'zgartirish, qo'shish.
3. Ma'lumotlar bazasi (jadval va yozuvlar)ga ma'lumotlarni kiritish, o'chirish va o'zgartirish;
4. MB ma'lumotlarini saralash.

Birinchi ikkita amalni bajarish huquqi faqatgina MBBT administratorlari va maxsus huquqi bor foydalanuvchilarga berilgan.

Ma'lumotlar bilan ishlashda ma'lumotlar bazasini yaratish, lozim bo'lganda bu ma'lumotlar saqlanadigan jadvallar strukturasi o'zgartirish va o'chirish kerak bo'ladi. Buning uchun SQL ning quyidagi operatorlari mavjud CREATE TABLE, ALTER TABLE va DROP TABLE.

Ma'lumotlar bazasini yaratish: CREATE TABLE operatori

CREATE TABLE operatori joriy ma'lumotlar bazasida berilgan nom bilan jadval yaratadi. Agar ma'lumotlar bazasi aktivlashmagan bo'lsa yoki berilgan jadval nomi mavjud bo'lsa, buyruqni bajarishda xatolik haqida ogohlantiradi.

Sintaksisi:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS]
```

```
jadval_nomi [(ustunlarni_anqlash,...)]
```

```
[jadval_opsiyasi] [select_ifoda]
```

MySQL 3.22 va undan keyingi versiyalarida jadval nomini ma'lumotlar_bazasi_nomi.jadval_nomi kabi ko'rsatadi. Yozuvning

bu ko'rinishi ko'rsatilgan ma'lumotlar bazasi joriy yoki joriymasligiga bog'liq emas.

MySQL 3.23-versiyasida vaqtinchalik jadval yaratish uchun TEMPORARY kalit so'zi ishlatiladi. Vaqtinchalik jadval bog'lanish yakunlanganda avtomatik tarzda o'chiriladi, uning nomi esa berilgan bog'lanish davomida haqiqiy hisoblanadi. Bu shuni anglatadiki, ikkita turli bog'lanishda mavjud jadval xuddi shunday nomli jadval bilan yoki bir xil nomli vaqtinchalik jadvallar bir-biri bilan konfliktisiz ishlatilishi mumkin (yoki mavjud jadval vaqtinchalik jadval o'chirilguncha yashirin bo'lib turadi). MySQL 4.0.2-versiyasida vaqtinchalik jadvallarni yaratish uchun CREATE TEMPORARY TABLES vakolatiga ega bo'lish zarur.

MySQL 3.23 va undan keyingi versiyalarida ko'rsatilgan jadval mavjud bo'lsa, xato bermasligi uchun IF NOT EXISTS kalit so'zi ishlatiladi. Bu holda jadval strukturalarining bir xilligi tekshirilmaydi.

Har bir jadval ma'lumotlar bazasi katalogida aniqlangan fayllar to'plami singari tasvirlanadi.

ustunlarni_anqlash parametri jadvalda qanday ustunlar bo'lishi kerakligini belgilaydi. Jadvalning har bir ustuni bo'sh bo'lishi (NULL), odatga ko'ra qiymatga ega bo'lishi, kalitli yoki avtoinkrement(AUTO_INCREMENT) bo'lishi mumkin. Bundan tashqari, har bir ustun uchun unda saqlanadigan ma'lumotlar tipini ko'rsatish kerak. Agar NULL ham, NOT NULL ham ko'rsatilmasa, u holda ustun NULL ko'rsatilgandek, interpretatsiya qilinadi. Agar maydon avtoinkrement sifatida belgilansa, u holda jadvalga ma'lumot kiritilganda va bu maydonga bo'sh qiymat (NULL, yoki hech narsa yozilmaganda) yoki 0 kiritilganda uning qiymati har gal 1 ta qiymatga oshiriladi. Jadvalda avtoinkrement maydon faqatgina bitta bo'lishi mumkin, shuning uchun uni indekslash shart emas. Avtoinkrement ketma-ketligi 1 dan boshlanadi.

Qoidaga ko'ra ustunlar quyidagicha beriladi (ustunlarni_anqlash):

```
ustun_nomi tipi [NOT NULL | NULL]
[DEFAULT odatga ko'ra qiymati]
[AUTO_INCREMENT][PRIMARY KEY]
[reference_definition]
```

Ustun tipi (ustunlarni_aniqlash parametrining ustun_nomi tipi) quyidagilardan birortasini qabul qilishi mumkin:

- butun: INT[(length)] [UNSIGNED] [ZEROFILL]
- haqiqiy: REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
- belgili: CHAR(length) [BINARY] va VARCHAR(length) [BINARY]
- katta obyektlar bilan ishlash uchun: BLOB
- matnli: TEXT
- sana va vaqt: DATE va TIME

Maydon (ustun) uzunligi kiritilayotgan qiymatda jami nechta raqam (belgi) qatnashishi kerakligini aniqlaydi. dec qatnashgan maydonlarda esa bu o'nlik kasrlarda nuqtadan so'ng, ya'ni kasr qismi nechta raqamdan iborat bo'lishi kerakligini bildiradi. Bu sonlar faqatgina ustunning maksimal uzunligini hisoblash uchun ishlatiladi.

ustunlarni_aniqlashda ustunlar soni va tipi bilan birga chegara-lash va tekshirish uchun kalitli va indeksli ustunlar ro'yxatini ham berish mumkin:

PRIMARY KEY (indekslangan_ustun_nomi, ...)

yoki

KEY [indeks_nomi] (indekslangan_ustun_nomi,...)

yoki

INDEX [indeks_nomi] (indekslangan_ustun_nomi,...)

yoki

UNIQUE [INDEX] [indeks_nomi] (indekslangan_ustun_nomi,...)

yoki

FULLTEXT [INDEX] [indeks_nomi] (indekslangan_ustun_nomi,...)

yoki

[CONSTRAINT symbol]

FOREIGN KEY [indeks_nomi] (indekslangan_ustun_nomi,...)

[reference_definition]

yoki

CHECK (expr)

FOREIGN KEY, CHECK va REFERENCES lar ayni vaqtda MySQLda hech qanday vazifani bajarmaydi. Ular boshqa SQL-

serverlar bilan hamkorlik qilish uchun qo'shilgan. Shuning uchun ularga to'xtalmaymiz.

Ko'p hollarda ustunga kiritilgan qiymatlar bir-biridan farq qilishi mumkin. Agar ustun uchun UNIQUE cheklanish o'rnatilsa, bu ustunga mavjud qiymatni kiritishga urinish rad etiladi. Bu cheklanish bo'sh emas (NOT NULL) deb e'lon qilingan ustunlarga qo'llanishi mumkin.

Masalan,

```
CREATE TABLE Student
(id INT NOT NULL UNIQUE,
surname VARCHAR(100), name VARCHAR(50),
stipend INT, kurs INT, city VARCHAR(100),
birthday DATA, univ_id INT );
```

Unikalligi talab qilinadigan ustunlardagi kalitlar (birlamchi kalitlardan tashqari) kandidat kalitlar yoki kalitlar deyiladi.

Jadval cheklanishi UNIQUE ustunlar guruhiga o'rnatilishi mumkin. Bu bir necha ustunlar qiymatlari kombinatsiyasi unikaligini ta'minlaydi.

SQL birlamchi kalitlarni to'g'ridan to'g'ri birlamchi kalit (PRIMARY KEY) cheklanishi orqali ta'riflaydi. PRIMARY KEY jadvalni yoki ustunlarni cheklashi mumkin. Bu cheklanish UNIQUE cheklanishi kabi ishlaydi, jadval uchun faqat bitta birlamchi kalit (ixtiyoriy sondagi ustunlar uchun) aniqlanishi mumkin. Birlamchi kalitlar NULL qiymatiga ega bo'lishi mumkin emas.

Masalan:

```
CREATE TABLE Student
(id INT NOT NULL PRIMARY KEY,
surname VARCHAR(100), name VARCHAR(50),
stipend INT, kurs INT, city VARCHAR(100),
birthday DATA, univ_id INT);
```

Jadval yaratishda, uni ba'zi xususiyatlari (jadval_opsiyasi)ni ham ko'rsatish kerak, masalan:

- jadval tipi: TYPE = {BDB | HEAP | ISAM | InnoDB | MERGE | MRG_MYISAM | MYISAM }

- avtoinkrement hisoblagichining boshlang'ich qiymati: AUTO_INCREMENT = son

- jadval satrining o'rtacha uzunligi: AVG_ROW_LENGTH = son

- jadval uchun izoh (60 belgidan iborat satr): COMMENT = «satr»

- satrning taxminiy eng katta va eng kichik qiymati: MAX_ROWS = son va MIN_ROWS = son

NOT NULL jadval ustunidagi barcha yacheykalar qiymatga ega bo'lishi zarurligini bildiradi, agar NOT NULL ko'rsatilmasa, ustun qiymatlari bo'sh (NULL) bo'lishi mumkin.

AUTO_INCREMENT — sonli ustunlar uchun ishlatiladigan MySQL ning maxsus imkoniyati. Agar jadvalga satrlarni qo'yishda ushbu maydonni bo'sh qoldirilsa, MySQL avtomatik tarzda identifikatorning unikal qiymatini qo'yadi. Bu qiymat maydondagi mavjud maksimal qiymatdan bittaga katta bo'ladi. Har bir jadvalda bunday maydon faqatgina bitta bo'ladi. AUTO_INCREMENT ustunlari indekslangan bo'lishi zarur.

PRIMARY KEY kalit so'zi ustun nomidan so'ng aniqlanib, ushbu ustun jadval uchun birlamchi kalit ekanligini anglatadi. Ushbu ustundagi ma'lumotlar unikal bo'lishi lozim. MySQL ushbu ustunni avtomatik tarzda indekslaydi.

UNSIGNED chegaralashi qo'yilgan ustun qiymati yoki butun son yoki nol bo'lishi mumkin.

10.11-misol. 10.2-rasmda strukturasi berilgan Talabalar portalining o'quv jarayoni jadvallarini yaratamiz.

```
mysql>CREATE TABLE Student
(stud_id INT UNSIGNED NOT NULL PRIMARY KEY,
 surname VARCHAR(100), name VARCHAR(100),
 stipend INT, kurs INT, city VARCHAR(200),
 birthday DATE, univ_id INT UNSIGNED NOT NULL );
mysql>CREATE TABLE exam_marks
(exam_id INT UNSIGNED NOT NULL PRIMARY KEY,
 stud_id INT UNSIGNED NOT NULL,
 subj_id INT UNSIGNED NOT NULL,
 mark INT, exam_date DATA );
mysql>CREATE TABLE subject
(subj_id INT UNSIGNED NOT NULL PRIMARY KEY,
 subj_name VARCHAR(100) NOT NULL,
 hour INT, semestr INT );
mysql>CREATE TABLE lecturer
```



```
(lecturer_id INT UNSIGNED NOT NULL PRIMARY KEY,
  surname VARCHAR(100) NOT NULL,
  name VARCHAR(100) NOT NULL, city VARCHAR(200),
  univ_id INT UNSIGNED NOT NULL );
mysql>CREATE TABLE subj_lect
(lecturer_id INT UNSIGNED NOT NULL,
  subj_id INT UNSIGNED NOT NULL );
mysql>CREATE TABLE univercity
(univ_id INT UNSIGNED NOT NULL PRIMARY KEY,
  univ_name VARCHAR(100) NOT NULL,
  rating INT, city VARCHAR(200));
```

Jadval yaratishda CREATE buyrug'ining yana bir elementi SELECT ifodasi (select_ifodasi) ham qatnashishi mumkin.

[IGNORE | REPLACE] SELECT ...

(SELECT ning ixtiyoriy to'g'ri ifodasi)

Agar CREATE buyrug'i yordamida jadval yaratishda select_ifodasi ko'rsatilsa, u holda tanlashda olingan barcha maydonlar yaratilgan jadvalga qo'shiladi.

MySQL ning maxsus SHOW buyrug'i yordamida ma'lumotlar bazasi tarkibi (ma'lumotlar bazasi jadvallari va jadval maydonlari) ni ko'rish mumkin.

10.12-misol. Barcha ma'lumotlar bazalarini ko'rsatish:

```
mysql>SHOW databases;
```

10.13-misol. talaba ma'lumotlar bazasini joriy etish (aktivlashtirish):

```
mysql>use talaba;
```

10.14-misol. Joriy (aktiv) bazadagi barcha jadvallarini ko'rsatish:

```
mysql>show tables;
```

MySQL ma'lumotlar bazasidagi jadvallar ro'yxatini chiqazadi:

Tables in talaba
student
exam_marks
subject
lecturer
subj_lect
univercity

6 rows in set (0.07 sec)

DESCRIBE buyrug'i jadval haqida qo'shimcha ma'lumot olish imkoniyatini beradi.

10.15-misol. univercity jadvalidagi maydonlar haqida ma'lumot olish.

```
mysql> describe univercity;
```

MySQL foydalanuvchiga jadval haqidagi ma'lumotni beradi.

10.16-misol. student jadvalidagi barcha ustunlarni ko'rsatish:

```
mysql> show columns from student;
```

Ma'lumotlar bazasiga ma'lumotlarni kiritish: INSERT operatori

INSERT operatori mavjud jadvalga yangi satr kiritadi. Bu operatorning bir nechta formasi mavjud bo'lib, jadval_nomi parametri ushbu barcha formalarda satrlar kiritilishi lozim bo'lgan jadval nomini bildiradi. Qiymat kiritiladigan ustunlar ustun nomlari (ustun_nomi) ro'yxatida yoki SET qismida ko'rsatiladi.

Sintaksisi(1-forma):

INSERT [LOW_PRIORITY | DELAYED] [IGNORE]

[INTO] jadval_nomi [(ustun_nomi,...)]

VALUES (qiymat,...),(...),...

INSERT buyrug'ining bu formasi satrni buyruqda ko'rsatilgan qiymatga mos ravishda kiritadi. Jadval nomidan so'ng berilgan qavs ichida ustunlar, VALUES kalit so'zidan so'ng esa ularning qiymatlari kiritiladi.

10.17-misol. Student jadvaliga familiyasi (surname) va tug'ilgan sanasi (birthday) qiymatlari mos ravishda «Akromov» va «1992» ga teng bo'lgan satrlarni kiriting.

```
mysql> INSERT INTO Student (surname, birthday)
VALUES ('Akromov', '1992');
```

Sintaksisi(2-forma):

INSERT [LOW_PRIORITY | DELAYED] [IGNORE]

[INTO] jadval_nomi [(ustun_nomi,...)]

SELECT ...

INSERT buyrug'ining bu formasida boshqa jadval yoki jadvallardan olingan satrlarni kiritadi.

10.18-misol. subj_lect jadvalining o'qituvchining identifikatori (lecturer_id) maydoniga lecturer jadvalidan shart bo'yicha (famili-

yasi 'Qobulov', universitet identifikatori 10 ga teng) tanlab olingan satrni qo'yish.

```
mysql> INSERT INTO subj_lect (lecturer_id)
SELECT lecturer_id FROM lecturer
WHERE surname='Qobulov'
AND univ_id='10';
```

Sintaksisi (3-forma):

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
[INTO] jadval_nomi
SET ustun_nomi = qiymat,
ustun_nomi = qiymat, ...
```

10.19-misol. Student jadvaliga surname maydoni 'Zuparov', name maydoni 'Odil' qiymatiga teng satrni kiriting.

```
mysql> INSERT INTO Student
SET surname='Zuparov', name='Odil';
```

INSERT ... VALUES bir nechta qiymatdan iborat ro'yxatli formasi MySQL 3.22.5 va undan keyingi versiyalarida qo'llaniladi. ustun_nomi =qiymat ifodalari sintaksisi MySQL 3.22.10 va undan keyingi versiyalarida qo'llaniladi.

Jadvalga ma'lumotlarni kiritishda quyidagilarga amal qilinadi:

- Agar INSERT ... VALUES yoki INSERT ... SELECT lar uchun ustunlar ro'yxati ko'rsatilmagan bo'lsa, u holda barcha ustunlar uchun qiymatlar ro'yxati (VALUES()) yoki SELECT ishi-ning natijasi aniqlangan bo'lishi kerak. Agar jadvalda ustunlar tartibi aniqlanmagan bo'lsa, uni olish uchun DESCRIBE jadval_nomi buyrug'i ishlatiladi.

- Qiymati aniq ko'rsatilmagan ixtiyoriy ustun uchun qiymat odatga ko'ra o'rnatiladi. Masalan, agar berilgan ustunlar ro'yxatida jadvalning barcha ustunlari ko'rsatilmagan bo'lsa, u holda tushirib qoldirilgan ustunlarga odatga ko'ra qiymat o'rnatiladi.

- expression ifodasi avvalroq qiymatlar ro'yxatiga kiritilgan ixtiyoriy ustunlarga tegishli bo'lishi mumkin. Masalan, uni quyidagicha ko'rsatish mumkin:

```
mysql> INSERT INTO jadval_nomi (col1,col2)
VALUES(15,col1*2);
```

quyidagicha ko'rsatish mumkin emas:

```
mysql> INSERT INTO jadval_nomi (col1,col2)
VALUES(col2*2,15);
```

Yuqoridagi uchta buyruq formasida ishtirok etuvchi zarur bo'lmagan uchta parametr haqida to'xtalamiz: **LOW_PRIORITY**, **DELAYED** va **IGNORE**.

LOW_PRIORITY va **DELAYED** parametrlaridan ko'p miqdordagi foydalanuvchilar jadval bilan ishlayotganda foydalanishadi. Ular berilgan amal boshqa foydalanuvchilar amallaridan muhimligini (birinchi bajarilishi kerakligini) o'rnatishga ko'rsatma beradi. Agar **LOW_PRIORITY** kalit so'zi ko'rsatilsa, berilgan **INSERT** buyrug'ini boshqa mijozlar bu jadvalni o'qishni tugatgunicha to'xtatib turiladi. Bu holatda mijoz berilgan kiritish buyrug'i o'z ishini tugatguncha kutib turishi shart. Chunki bu paytda shiddat bilan ishlayotgan jadval anchagina vaqt talab qilishi mumkin. **INSERT** ning bu buyrug'ining aksi **DELAYED** esa mijoz uchun boshqa foydalanuvchilarga bog'liq bo'lmagan holda (mustaqil ravishda) amalni hoziroq davom ettirishga imkon yaratadi.

Agar **INSERT** buyrug'ida **IGNORE** kalit so'zi ko'rsatilsa, bu jadvaldagi **PRIMARY** ili **UNIQUE** takroriy kalitlarga ega bo'lgan barcha satrlar bekor qilinadi va jadvalga kiritilmaydi. Agar **IGNORE** ko'rsatilmagan bo'lsa, mavjud kalit qiymatlari takrorlangan satrlar topilganda bajarilayotgan kiritish amali to'xtatiladi.

10.20-misol. Avvalgi ma'ruzalarda yaratilgan talabalar portali-ning jadvallariga ma'lumotlarni kiritamiz:

```
mysql> INSERT INTO Student VALUES
(1, 'Islomov', 'Ikrom', 150, 1, 'Andijon', '3-12-1992', 10),
(3, 'Fozilov', 'Fozil', 200, 3, 'Qashqadaryo', '1-12-1990', 10),
(6, 'Salimov', 'Vohid', 150, 4, 'Namangan', '7-06-1989', 22),
(10, 'Karimov', 'Botir', 0, 2, 'Jizzax', '8-12-1991', 10);
mysql> INSERT INTO lecturer VALUES
(24, 'Qobulov', 'Botir', 'Toshkent', 10),
(46, 'Nasriddinov', 'Islom', 'Toshkent', 10),
(74, 'Hasanov', 'Farhod', 'Namangan', 22),
(108, 'Sobirov', 'Nizom', 'Namangan', 22);
mysql> INSERT INTO subject VALUES
(10, 'Informatika', 56, 1),(22, 'Fizika', 34, 1).
```



```
(43, 'Matematika', 56, 2);
mysql> INSERT INTO univercity VALUES
(22, 'NamDU', '34', 'Namangan'),(10, 'TDPU', '29', 'Toshkent'),
(11, 'AnDU', '60', 'Andijon');
mysql> INSERT INTO exam_marks VALUES
(10, 12, 10, 5, '12-01-2010'),(34, 32, 10, 4, '23-01-2010'),
(75, 55, 10, 5, '05-01-2010'),(238, 12, 22, 3, '17-06-2009');
mysql> INSERT INTO subj_lect VALUES
(24, 24),(46, 46),(74, 74),(108, 108);
```

Ma'lumotlar bazasidan ma'lumotlarni olish: SELECT operatori

SELECT operatori tanlangan bir yoki bir nechta jadvaldan satrlar (yozuvlar)ni olish uchun ishlatiladi. Uning yordamida saralash amalga oshiriladigan table_references jadvalidan olinishi kerak bo'lgan ifoda (select_ifoda) yoki ustunlar hamda ushbu ustundagi ma'lumotlarga mos keluvchi (where_definition) shartlar, bu ma'lumotlarni chiqarish tartibi – group_type va natijaviy ma'lumotlarni qanchasi brauzerga chiqazilish kerak – limit_criteria da ko'rsatiladi.

Sintaksisi:

```
SELECT select_ifoda1, select_ifoda2,
```

```
...
```

```
[FROM table_references
```

```
[WHERE where_definition]
```

```
[ GROUP BY group_type ]
```

```
[ HAVING where_definition]
```

```
[ORDER BY {son | ustun_nomi| formula} [ASC | DESC], ...]]
```

```
[ LIMIT limit_criteria ] ;
```

[] to'rtburchak qavs, uning ichidagi parametrlarni berish shart emasligini bildiradi, | to'g'ri chiziq esa bo'lishi mumkin bo'lgan variantlar. ORDER BY kalit so'zidan so'ng ustun_nomi, son (belgisiz butun) yoki formula va tartiblash usuli (o'sish tartibi bo'yicha – ASC yoki kamayish tartibi bo'yicha – DESC) ko'rsatiladi. Odatga ko'ra tartiblash usuli berilmaganda tartiblash o'sish tartibida amalga oshiriladi.

select_ifoda o'rnida «*» yozilsa, bu barcha ustunlarni chiqazish kerakligini anglatadi. select_ifoda da «*» dan tashqari max, min va avg kabi funksiyalari ham ishlatiladi.

Bundan tashqari SELECT operatori yordamida biror-bir jadvalni ishlatmasdan, biror-bir satrni olmasdan hisoblash amallarini ham bajarish mumkin.

masalan, $2*2$ nechaga tengligini hisoblash uchun, quyidagicha yozish kerak:

```
mysql> SELECT 2*2;
4
```

WHERE konstruksiyasi ishlatilganda where_definition parametrida ma'lumotlarni MB dan olish uchun shartlar beriladi. Ushbu shartlar turli ifodalar yoki amallar bo'lishi mumkin.

10.7-jadval

WHERE konstruksiyasi bilan ishlovchi operatorlar

Amal	Nomi	Misol	Vazifasi
=	teng	stipend=150	Ikkala qiymat tengligini tekshiradi
>	katta	stipend>100	Bitta qiymat ikkinchisidan kattaligini tekshiradi
<	kichik	stipend<150	Bitta qiymat ikkinchisidan kichikligini tekshiradi
>=	Katta yoki teng	stipend>=100	Bitta qiymat ikkinchisidan katta yoki teng munosabatini tekshiradi
<=	Kichik yoki teng	stipend<=150	Bitta qiymat ikkinchisidan kichik yoki teng munosabatini tekshiradi
!= yoki o	Teng emas	hour!=0	Ikkala qiymat teng emasligini tekshiradi

10.21-misol. Student jadvalidan 'Vohid' ismli talabalarning barcha ma'lumotlarini chiqazing:

```
mysql> SELECT * FROM Student
WHERE name='Vohid';
```

10.22-misol. Universitet identifikatori 10 ga teng bo'lgan o'qituvchilar (lecturer jadvalidan)ning ismi va shahrini (name, city) chiqazing:

```
mysql> SELECT name, city
FROM lecturer WHERE univ_id=10;
```

10.23-misol. Student jadvalidan barcha talabalar ro'yxatini tanlash.

```
mysql> SELECT *
FROM student;
```


Agar ushbu so'rovga **ORDER BY** surname ifodasi qo'shilsa, u holda ro'yxat familiya bo'yicha tartiblanadi. Odatga ko'ra tartiblash, o'sish tartibi bo'yicha bajariladi. Agar kamayishi bo'yicha tartiblash kerak bo'lsa, u holda oxirgi ifodadagi atribut nomidan keyin **DESC** so'zi qo'shiladi.

SQL tilida shart ifodalarini tuzish uchun solishtirish va mantiqiy operatorlardan tashqari yana bir qator maxsus operatorlar qo'llaniladi. Bu operatorlar boshqa dasturlash tillarida mavjud emas.

10.8-jadval

WHERE konstruksiyasi bilan ishlovchi maxsus operatorlar

Operator	Nomi	Misol	Vazifasi
IS NOT NULL	Manzili nolga teng emas (bo'sh emas)		Maydon qiymati mavjudligini tekshiradi
IS NULL	Manzili nolga teng (bo'sh)		Maydon qiymati mavjud emasligini tekshiradi
BETWEEN	oralig'ida	BETWEEN 2 AND 3	Biror qiymatlar diapazoniga tegishliligini tekshirish
IN	oralig'ida	IN (2,3)	Biror qiymatlar to'plamiga tegishliligini tekshirish
NOT IN	Oralg'ida emas	NOT IN (2,3)	Biror qiymatlar to'plamiga tegishli emasligini tekshirish
LIKE	moslik	name like('Anvar%')	Namuna bilan mosligini tekshirish
NOT LIKE	moslik	name not like('Anvar%')	Namuna bilan mos emasligini tekshirish
REGEXP	Doimiy ifoda	name regexp	Qiymat doimiy ifodaga mosligini tekshiradi

10.24-misol. Familiyasi K harfi bilan boshlanuvchi talabalar ro'yxatini tanlab olish.

Bunday holatda **LIKE** operatoridan foydalanish qulay hisoblanadi. **LIKE** operatori faqat belgili maydonlar uchun qo'llaniladi va maydon qiymati operatorida ko'rsatilgan namunaga mosligini tekshirish imkonini yaratadi. Namuna quyidagi maxsus belgilardan tashkil topadi:

_ (tagiga chizish belgisi) – bitta ixtiyoriy belgini bildiradi;

% (foiz belgisi) – ixtiyoriy miqdordagi belgilar ketma-ketligini bildiradi.

```
mysql> SELECT surname, kurs  
FROM student  
WHERE surname LIKE 'K%';
```

Ko'pincha ustunlardagi minimal, maksimal yoki o'rtacha qiymatlarni hisoblashga to'g'ri keladi. Masalan, keltirilgan imtihon natijalari jadvalida o'rtacha bahoni hisoblash mumkin. Bunday hisoblashni bajarish uchun SQL tilida maxsus agregat (murakkab amallarni bajaruvchi) funksiyalari mavjud:

MIN() – ustundagi minimal qiymat;
MAX() – ustundagi maksimal qiymat;
SUM() – ustundagi qiymatlar summasi;
AVG() – ustundagi qiymatlarni o'rtachasi;
COUNT() – ustundagi NULL dan farqli qiymatlar miqdori.

13.26-misol. Talabalar olgan baholarning o'rtachasini hisoblang.

```
mysql> SELECT AVG(mark)  
FROM exam_marks;  
Natija:  
AVG(mark)  
4.25
```

Agregat funksiyalarini ham WHERE so'zi bilan birgalikda qo'llash mumkin.

10.25-misol. Identifikator nomeri 10 ga teng bo'lgan fan bo'yicha talabalarning olgan baholari o'rtachasini hisoblang:

```
mysql> SELECT AVG(mark)  
FROM exam_marks  
WHERE subj_id = 10;  
Natija:  
AVG(mark)  
4,6666666666666667
```

Biror xulosaga kelishdan oldin SQL tilining barcha imkoniyatlarini ko'rib chiqish kerak. Masalan, ixtiyoriy matnni so'rov tarkibiga kiritish mumkin. 10.25-misolni quyidagicha yozamiz:

```
mysql> SELECT 'O'rtacha ball =', AVG(mark)  
FROM exam_marks  
WHERE subj_id = 10  
Natija:  
O'rtacha ball =4,6666666666666667
```

Agregat funksiyalarini jadval ustunlarining guruhlangan qiymatlari uchun ham qo'llash mumkin. Bu maxsus GROUP BY konstruk-

siyasi bo'lib, unda ko'rsatilgan ustun qiymatlari bo'yicha guruhlash amalga oshiriladi.

10.26-misol. Fanlar bo'yicha talabalarning olgan baholarini 'rtachasini hisoblang va fanlarni o'sish tartibi bo'yicha guruhlab chiqazing:

```
mysql> SELECT subj_id, AVG(mark)
FROM exam_marks
GROUP BY subj_id;
```

Bu imkoniyat ham odatdagidek WHERE so'zi bilan birgalikda qo'llanilishi mumkin. Bu so'rovni bajarishda MBBT avval jadvaldan WHERE ifodasidagi shartni qanoatlantiruvchi satrlarni tanlab oladi, keyin tanlangan satrlarni guruhlash va agregatlash amalini bajaradi.

10.27-misol. Fanlar bo'yicha 4 bahodan yuqori baho olgan talabalarni identifikator nomeri bo'yicha guruhlab chiqazing:

```
mysql> SELECT stud_id, mark
FROM exam_marks
WHERE mark >=4
GROUP BY stud_id;
```

Guruhlash bittadan ko'p maydonlar bo'yicha ham bajarilishi mumkin. GROUP BY seksiyasiga ega bo'lgan so'rovlar uchun quyidagi muhim cheklanish mavjud: bunday so'rovlar natijasi tarkibiga guruhlash bajarilgan ustunlar va agregatlash natijasiga ega bo'lgan ustunlar kiradi.

Agregat va guruhlash funksiyalarini ishlatishda quyidagilarga e'tibor berish talab etiladi:

ANSI SQL da agar agregat funksiyalari yoki GROUP BY konstruksiyasi ishlatilsa, SELECT konstruksiyasi faqatgina agregat funksiyalarini va GROUP BY konstruksiyasida ko'rsatilgan ustunlarni qabul qiladi. Agar GROUP BY konstruksiyasida ustunni ishlatish talab etilsa, u albatta SELECT konstruksiyasida ko'rsatilishi kerak.

MySQL esa amallardan cheksiz foydalanishni ta'minlaydi va SELECT konstruksiyasida kerak bo'lmagan elementlarni olib tashlash imkoniyatini beruvchi kengaytirilgan sintaksisni qo'llaydi.

Ma'lumotlarni guruhlash va murakkab amallarni bajarishga qo'shimcha ravishda HAVING konstruksiyasi yordamida murakkab natijalarni tekshirishning barcha imkoniyatlari mavjud. U GROUP

BY konstruksiyasidan so'ng ishlatiladi va WHERE ga o'xshash bo'lib, faqatgina guruhlariga qo'llaniladi.

Yuqoridagi misolni kengaytiramiz, fanlar bo'yicha o'rtacha o'zlashtirish ko'rsatkichi 4 dan yuqori bo'lgan fanlarni chiqazamiz:

```
mysql> SELECT subj_id, AVG(mark)
FROM exam_marks
GROUP BY subj_id
HAVING AVG(mark)>4;
```

E'tibor bering, HAVING konstruksiyasi guruhga murojaat etyapti. Ushbu so'rov quyidagicha natijani beradi:

subj_id	mark
10	4,666666666666667

Ushbu so'rov natijasida foydalanuvchi faqat oddiy sonlarni emas, balki matnni ham ko'radi.

SELECT operatorining Web hujjatlarda foydalanish mumkin bo'lgan LIMIT konstruksiyasi bilan tanishamiz. Bu konstruksiyada ekranga natijaning nechta satri chiqazilishi ko'rsatiladi. Ushbu konstruksiyada ikkita parametr ko'rsatilishi zarur: satr nomeri (chiqariladigan satrlarning boshlang'ich nomeri) va satrlar soni. Satrlarni nomerlash noldan boshlanadi.

10.28-misol. Talabalar jadvalidan barcha talabalarning ismlarini olib, natijaning ikkinchi satridan boshlab ikkita satr chiqazing.

```
mysql> SELECT name FROM student LIMIT 1,2;
```

Masalan, MB dan olingan talabalar ro'yxati ko'pgina satrlardan iborat ularni alohida sahifalarda tasvirlash uchun natijani sahifaga bo'lib-bo'lib chiqazish lozim. Ushbu hollarda LIMIT konstruksiyasi foydalanish uchun juda qulay.

Ma'lumotni bir nechta jadvallardan olish

Ikkita jadvalni bog'lash

10.29-misol. Andijonda tug'ilgan talabalarning ismi, familiyasi va ta'lim muassasasi nomini chiqazing.

```
mysql> SELECT student.surname, student.name, univercity.univ_name
FROM student, univercity
Where student.city= 'Andijon'
and student.univ_id=univercity.univ_id;
```

10.30-misol. Imtihonlarda 4 yoki 5 baho olgan talabalar ro'yxatini chiqazish.

Buning uchun WHERE so'zidan keyin solishtirish amallarini (<, >, =, <> va x.k.) va mantiqiy operatorlar qatnashgan ifodalarni joylashtirish mumkin.

```
mysql> SELECT student.surname, student.name, subject.subj_name,  
exam_marks.mark  
FROM student, subject, exam_marks  
WHERE student.stud_id=exam_marks.stud_id  
AND subject.subj_id= exam_marks.subj_id  
AND (( mark >= 4 ) AND ( mark<= 5 ));
```

10.30-misolni IN operatoridan foydalanib yozamiz (imtihonlarda kamida bitta 4 yoki 5 baho olgan talabalar familiyasi, ismi, olgan bahosini tanlab olish).

```
mysql> SELECT student.surname, student.name, subject.subj_name,  
exam_marks.mark  
FROM student, subject, exam_marks  
WHERE student.stud_id=exam_marks.stud_id  
AND subject.subj_id= exam_marks.subj_id  
AND mark IN (2,3);
```

Ushbu natijani BETWEEN operatoridan foydalanib ham olish mumkin:

```
mysql> SELECT student.surname, student.name, subject.subj_name,  
exam_marks.mark  
FROM student, subject, exam_marks  
WHERE student.stud_id=exam_marks.stud_id  
AND subject.subj_id= exam_marks.subj_id  
AND mark BETWEEN 2 AND 3;
```

Ma'lumotlar bazasidagi satrlarni (yozuvlarni) yangilash: UPDATE operatori

UPDATE operatori jadvalda mavjud ustun qiymatini mos kiritilgan qiymatga o'zgartiradi.

Sintaksisi:

```
UPDATE [LOW_PRIORITY] [IGNORE] jadval_nomi  
SET ustun_nomi1=ifodal  
[, ustun_nomi2=ifoda2, ...]  
[WHERE where_definition]  
[LIMIT limit_criteria]
```

Bu yerda SET ifodasida aynan qaysi ustun o'zgartirilishi va qanday qiymat unga o'rnatilishi kerakligi ko'rsatiladi. WHERE dan so'ng, yangilanishi kerak bo'lgan satrlar ko'rsatiladi, agar hech qan-

day satr ko'rsatilmasa, barcha satrlar yangilanadi. Agar ORDER BY ifodasi berilgan bo'lsa, u holda satrlar unda ko'rsatilgan tartibda yangilanadi.

Agar LOW_PRIORITY kalit so'zi ko'rsatilsa, berilgan UPDATE buyrug'ini bajarish boshqa mijozlar bu jadvalni o'qishni tugatguncha to'xtatib turiladi.

Agar IGNORE kalit so'zi ko'rsatilsa, u holda yangilash buyrug'i to'xtatilmaydi, hatto agar kalitlar takrorlansa ham xato chiqazmaydi. Konflikt holatlar olib keladigan satrlar yangilanmaydi.

Agar ustunga yangi qiymat berilayotgan ifodada ushbu maydon (ustun) nomi ishlatilsa, u holda UPDATE ushbu ustun uchun uning joriy qiymatini ishlatadi.

10.31-misol. hour ustuniga uning joriy qiymatidan bitta katta qiymatni o'rnating:

```
mysql> UPDATE subject  
SET hour=hour+1;
```

MySQL 3.23-versiyasida LIMIT # parametrini faqatgina # da berilgan sondagi satrlar o'zgartirilganligiga ishonch hosil qilish uchun ishlatiladi.

10.32-misol. Jadvaldagi birinchi satrni ko'rsatilgan tartibda o'zgartiring.

```
mysql> UPDATE subject  
SET hour=55 Limit 1;
```

Jadval yaratilgandan so'ng o'zgartirish: ALTER TABLE operatori

ALTER TABLE operatori mavjud jadval strukturasini o'zgartirish uchun imkon yaratadi. Masalan, jadvalga ustun qo'shish yoki olib tashlash, indeks yaratish yoki o'chirish, ustun nomini o'zgartirish kabi amallarni bajaradi. Hattoki jadval uchun izohni va uning tipini ham o'zgartirish mumkin.

Sintaksisi:

ALTER [IGNORE] TABLE jadval_nomi

alter_specification [, alter_specification ...]

Jadvalni o'zgartirish uchun quyidagilar bajariladi (quyidagilarning barchasi alter_specification o'rnida yoziladi):

- maydon qo'shish:

ADD [COLUMN] ustunlarni_anqlash [FIRST | AFTER ustun_nomi]

Yoki

ADD [COLUMN] (ustunlarni_aniqlash1, ustunlarni_aniqlash2,...)

Bu va bundan keyin ustunlarni_aniqlash jadval yaratishda qanday yozilgan bo'lsa, xuddi shunday yoziladi.

- indekslarni qo'shish:

ADD INDEX [indeks_nomi] (indekslangan_ustun_nomi,...) yoki

ADD PRIMARY KEY (indekslangan_ustun_nomi,...) yoki

ADD UNIQUE [indeks_nomi] (indekslangan_ustun_nomi,...)

yoki

ADD FULLTEXT [indeks_nomi] (indekslangan_ustun_nomi,...)

- maydonni o'zgartirish:

ALTER [COLUMN] ustun_nomi

{SET DEFAULT literal | DROP DEFAULT}

yoki

CHANGE [COLUMN] ustunning_eski_nomi ustunlarni_aniqlash

lash

yoki

MODIFY [COLUMN] ustunlarni_aniqlash

- kalit, indeks, maydonni o'chirish:

DROP PRIMARY KEY

DROP INDEX indeks_nomi

DROP [COLUMN] ustun_nomi

- jadval nomini o'zgartirish:

RENAME [TO] jadvalning_yangi_nomi

- jadval maydonlarini qayta tartiblash:

ORDER BY maydon

yoki

jadval_opsiyasi

ALTER TABLE operatori yordamida ustun tipini o'zgartirgandan so'ng bu buyruq bajarilganligini tekshirish uchun **DESCRIBE jadval_nomi** buyrug'i ishlatiladi. Agar ustun tipi o'zgartirilmagan bo'lsa, demak MySQL ma'lumotlar shaklini o'zgartirishni bitta sababga ko'ra inkor etgan. Masalan, ustun tipini **VARCHAR** dan **CHAR** ga o'zgartirishda, berilgan jadval o'zgaruvchan uzunlikka ega ustunlardan tarkib topgan bo'lsa, MySQL yana **VARCHAR** ni ishlatishda davom etadi.

ALTER TABLE ishlash jarayonida mavjud jadvalning vaqtinchalik nusxasini yaratadi. Talab qilinayotgan o'zgartirishlar nusxa ustida bajariladi, so'ng eski jadval o'chiriladi va uni o'rniga yangi yaratilgan jadval nomi qayta nomlanadi. Bu barcha o'zgartirishlar (muvaffaqiyatsizlaridan tashqari) yangi jadvalga avtomatik tarzda tushishi uchun bajariladi. **ALTER TABLE** ishlayotgan vaqtda, eski jadval boshqa mijozlarni ishlashi uchun ochiq bo'ladi. Jadvalga ma'lumotlarni kiritish va o'zgartirish vaqtincha to'xtatib turiladi va qachonki yangi jadval tayyor bo'lgandan so'ng, bu amallar bajariladi.

Agar **IGNORE** aniqlangan bo'lsa, u holda nusxa ko'chirish bekor qilinadi. **ALTER TABLE** da **RENAME** dan boshqa ixtiyoriy opsiya bajarilganda, MySQL har doim vaqtinchalik jadval nusxasini yaratadi.

10.33-misol. Student jadvaliga izoh yozish uchun yangi maydon qo'shamiz:

```
mysql> ALTER TABLE Student  
ADD comment varchar(255) AFTER univ_id;
```

Demak, jadvallar bilan ishlashni: ularni yaratish, o'chirish va o'zgartirishni o'rgandik. Endi bu jadvallarda saqlanayotgan ma'lumotlar bilan ishlashni: ma'lumotlarni kiritish, o'chirish va o'zgartirishni o'rganamiz.

Ma'lumotlar bazasidan satrlarni (yozuvlarni) o'chirish: DELETE operatori

DELETE operatori jadval_nomi jadvalidan where_definition shartini qanoatlantiruvchi satrni o'chiradi va o'chirilgan yozuvlar sonini chiqazadi.

Agar **DELETE** operatori **WHERE** siz ishlatilsa, barcha satrlar o'chiriladi.

Sintaksisi:

```
DELETE [LOW_PRIORITY] FROM jadval_nomi  
[WHERE where_definition]  
[LIMIT rows]
```

10.34-misol. subj_lect jadvalidan o'quv fani identifikatori maydoni bo'sh bo'lgan barcha satrlarni o'chiring.

```
mysql> DELETE FROM subj_lect WHERE subj_id="" ;
```

10.35-misol. Jadvaldagi barcha yozuvlarni o'chirish uchun quyidagi buyruqdan ham foydalanish mumkin.


```
mysql> DELETE FROM Student WHERE I>0;
```

Quyidagi ko'rinishi ushbu buyruqni shartsiz ishlatishdan ko'ra sekinroq ishlaydi:

```
mysql> DELETE FROM student;
```

MySQL ning maxsus DELETE buyrug'i uchun LIMIT opsiyasi mijozga boshqaruvni qaytarguncha o'chirilgan satrlarni maksimal sonini tizimga ko'rsatadi.

Jadvalni o'chirish: DROP TABLE operatori

DROP TABLE operatori bir yoki bir nechta jadvallarni o'chiradi. Bu buyruq ishlatilganda jadvalning barcha ma'lumotlari va strukturasini o'chiriladi, shuning uchun bu buyruq bilan ishlaganda ehtiyot bo'lish kerak.

Sintaksisi:

```
DROP TABLE [IF EXISTS] jadval_nomi  
[, jadval_nomi,...]  
[RESTRICT | CASCADE]
```

MySQL 3.22 va undan keyingi versiyalarida agar ko'rsatilgan jadval mavjud bo'lmasa, xato haqida ogohlantirish uchun IF EXISTS kalit so'zi ishlatiladi.

RESTRICT va CASCADE opsiyasi dasturni boshqa MBBT ga o'tkazishni soddalashtirish uchun imkon yaratadi.

10.36-misol. DROP TABLE operatorining ishlatilishi

```
mysql> DROP TABLE IF EXISTS Student, lecturer, test;
```

Ma'lumotlar bazasini o'chirish: DROP DATABASE operatori

DROP DATABASE operatori butun ma'lumotlar bazasini o'chirish uchun ishlatiladi. Bu buyruq bilan ishlaganda ham ehtiyot bo'lish talab etiladi.

Sintaksisi:

```
DROP DATABASE baza_nomi;
```

Natijada barcha satrlar, jadvallar, indekslar (kalitlar) va ma'lumotlar bazasining o'zi o'chiriladi.

Demak, buyruqlar satri yordamida MySQL MBBT bilan ishlashni ko'rib chiqdik. Endi Mysql MBBT bilan ishlash uchun phpmyadmin dan foydalanamiz.

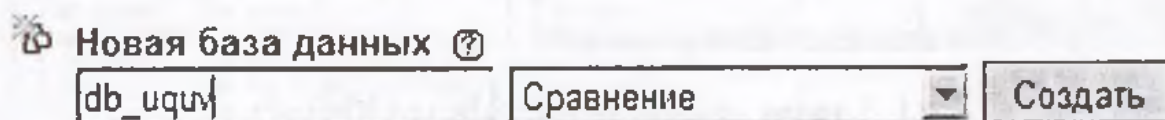
XI bob. PHP VA MYSQL NING O'ZARO BOG'LANISHI

Bu bobda PHP va MySQL MBBT ning o'zaro bog'lanishi usullari bilan tanishasiz. Asosiy e'tibor ma'lumotlar bazasi bilan aloqa o'rnatish, so'rovlarni jo'natish funksiyasi va javoblarni qayta ishlashga qaratiladi (mysql_connect, mysql_query, mysql_result, mysql_num_rows, mysql_close). MySQL ma'lumotlar bazasi bilan bog'lanish uchun PhpMyadmin ni ishlatish ham ko'rib chiqiladi.

11.1. PhpMyadmin yordamida MySQL ma'lumotlar bazasi bilan ishlash

PhpMyadmin brauzerning barcha imkoniyatlaridan to'liq foydalanish imkoniyatini beradi, jumladan, ekranga joylashmay qolgan rasmlarni surib ko'rish va b. PhpMyadminda SQL ning ma'lumotlar bilan ishlovchi ko'pgina bazaviy funksiyalari Web sahifalardagi kabi gipermurojaatga o'xshash, intuitiv tushunarli harakat va interfeyslar ko'rinishiga keltirilgan. Shunga qaramay, MySQL ning funksiyalarini yaxshiroq o'zlashtirib olish uchun matnli rejimida ham ishlab ko'rish tavsiya etiladi.

Avval buyruqlar satrida bajarilgan amallarni phpmyadmin da bajarish uchun brauzerning manzil qatorida <http://localhost/phpmyadmin/> kiritiladi. phpmyadmin yordamida MySQL MBBT ga kirib unda ma'lumotlar bazasi yaratiladi. Buning uchun oynadagi Новая база данных satrida yaratilishi kerak bo'lgan ma'lumotlar bazasi nomi (db_uquv) kiritiladi (bu nom chap tomondagi ma'lumotlar bazalari nomlari bilan bir xil bo'lmasligi kerak). Va Создать tugmasi bosiladi (11.1-rasm).



11.1-rasm. Yangi ma'lumotlar bazasini yaratish.

Ma'lumotlar bazasi yaratilgandan so'ng uni tarkibidagi jadvallar yaratiladi. Ushbu oynada jadval nomi va jadvalda bo'lishi kerak bo'lgan maydonlar soni ko'rsatiladi va ok tugmasi bosiladi (11.2 -rasm).

Таблиц в базе данных не обнаружено.

Создать новую таблицу в базе данных db_uquv

Имя: student Количество полей: 8

OK

11.2-рasm. Ma'lumotlar bazasida yangi jadval yaratish

Bu oynada yaratilgan jadvalning maydonlari nomi, tipi, uzunligi ko'rsatiladi. Bunga qo'shimcha ravishda maydonlarga autoincrement ko'rsatkichi va kalit (birinchi darajali va indeksli kalit) qo'yish mumkin (11.3-rasm).

phpMyAdmin

Сервер: localhost База данных: db_uquv Таблица: student

Поле	Тип	Длина/значение	По умолчанию
stud_id	INT	10	Нет
loginname	VARCHAR	100	Нет
name	VARCHAR	100	Нет
strend	INT	6	Нет
pass	TINYTEXT	4	Нет
city	VARCHAR	200	Нет
birthday	DATE		Нет
univ_id	TINYTEXT	4	Нет

Комментарий к таблице:

Определение разделов (PARTITION):

Тип таблиц: MyISAM

Сравнение:

11.3-rasm. Jadval maydonlarini kiritish

Xuddi shunday boshqa jadvallar ham yaratiladi va html forma yoki phpmyadmin yordamida bu jadvallarga ma'lumotlar kiritiladi.

Demak, ma'lumotlar bazasi va ma'lumotlar bazasida jadvallar yaratishni ko'rib chiqdik. Endi ushbu MySQL MBBT ga brauzer orqali bog'lanib, yaratilgan ma'lumotlar bazasi bilan ishlash uchun foydalanuvchi yaratiladi. Buning uchun Server:localhost sahifasida Привилегии bo'limi tanlanadi (11.4-rasm).



11.4-rasm. Foydalanuvchiga vakolat berish.

Bu bo'limda MySQL MBBT bilan ishlovchi foydalanuvchilar ro'yxati va ular haqidagi ma'lumotlar (MB bilan ishlash vakolatlari) joylashgan. Yangi foydalanuvchi yaratish uchun foydalanuvchilar oynasidan **Добавить нового пользователя** murojaati (ссылка) bosiladi.

Bu sahifada yangi foydalanuvchining nomi, xost nomi, paroli va MB bilan ishlash vakolatlari kiritiladi (11.5-rasm).

Добавить нового пользователя

Информация учетной записи

Имя пользователя:

Хост:

Пароль:

Подтвердите:

Создать пароль:

База данных для пользователя

☒ Не создавать

☐ Создать базу данных с именем пользователя и предоставить на нее полные привилегии

☐ Предоставить полные привилегии на базы данных подпадающие под шаблон (имя пользователя, %)

Глобальные привилегии (Отметить все / Снять выделение)

Примечание: для привилегий MySQL используются префиксы:

Данные	Структура	Администрирование
☒ SELECT	☒ CREATE	☒ GRANT
☒ INSERT	☒ ALTER	☒ SUPER
☒ UPDATE	☒ INDEX	☒ PROCESS
☒ DELETE	☒ DROP	☒ RELOAD
☒ FILE	☒ CREATE TEMPORARY TABLES	☒ SHUTDOWN
	☒ SHOW VIEW	☒ SHOW DATABASES
	☒ CREATE ROUTINE	☒ LOCK TABLES
	☒ ALTER ROUTINE	☒ REFERENCES
	☒ EXECUTE	☒ REPLICATION CLIENT
	☒ CREATE VIEW	☒ REPLICATION SLAVE

11.5-rasm. Yangi foydalanuvchi yaratish

Demak, ushbu sahifada lola ismli foydalanuvchini 135 paroli bilan localhost da ishlashi uchun barcha vakolatlarni berdik. Endi lola ismli foydalanuvchi MySQL tizimiga murojaat eta olishi mumkin.

11.2. MySQL tizimi bilan bog'lanishni o'rnatish va tizimda ishlash uchun ma'lumotlar bazasini tanlash

MySQL ma'lumotlar bazasiga WEB orqali PHP texnologiyasi yordamida so'rovlarni jo'natish algoritmi:

1. Foydalanuvchi tomonidan kiritilgan ma'lumotlarni filtrlash va tekshirish.

2. MySQL tizimi bilan bog'lanishni o'rnatish. Agar urinish muvaffaqiyatsiz yakunlansa, mos xabar chiqaziladi va jarayon yakunlanadi.

3. MySQL tizimida ishlash uchun ma'lumotlar bazasini tanlash. Agar tanlash urinishi muvaffaqiyatsiz yakunlansa, mos xabar chiqaziladi va jarayon yakunlanadi. So'rovlarni qayta ishlash uchun bir vaqtning o'zida bir nechta ma'lumotlar bazasini ochishga ruxsat beriladi.

4. Tanlangan bazada so'rovlarni qayta ishlash (ma'lumotlarni qo'shish uchun interfeys yaratish, ma'lumotlar bazasiga ma'lumotlarni yozish, ma'lumotlar bazasida saqlanayotgan ma'lumotlarni foydalanuvchiga taqdim etish).

5. So'rovlarni qayta ishlash tugatilgandan so'ng ma'lumotlar bazasi tizimi bilan bog'lanishni yopish yoki tugatish.

1. Foydalanuvchi tomonidan kiritilgan ma'lumotlarni filtrlash va tekshirish.

Avval foydalanuvchi tomonidan kiritilgan ma'lumot atrofidagi tasodifan yozilgan barcha ortiqcha bo'sh belgilarni (probellarni) o'chirish zarur. Buning uchun trim() funksiyasini qo'llanadi.

11.1-misol.

```
trim($city);
```

Keyingi bosqich — foydalanuvchi barcha maydonlarni to'ldirganligini tekshirish. trim() funksiyasi yordamida o'zgaruvchilar tarkibidagi qiymatlar tahrirlangandan so'ng o'zgaruvchilarni qiymatlari mavjud yoki mavjud emasligi tekshiriladi:

11.2-misol.

```
if (!$stud_id || !$name || !$surname || !$kurs || !$univ_id)
{ echo "Ko'rsatilgan maydonlar to'ldirilmagan. To'ldirilishi zarur
bo'lgan maydonlarga ma'lumotlarni kiriting va qayta urinib ko'ring." ;
exit; }
```

Foydalanuvchi ma'lumotlarini MySQL ma'lumotlar bazasiga yozishdan avval oldingi ma'ruzalarda ko'rib o'tilgan addslashes() va stripslashes() funksiyalari yordamida tahrirlab olish mumkin.

htmlspecialchars() funksiyasi ma'lumotlarda uchraydigan ((&) ampersand, (<) kichik, (>) katta, (") ikkitalik qo'shtirnoq) belgilarini HTML kodirovkasiga o'tkazish uchun ishlatiladi.

2. MySQL tizimi bilan bog'lanishni o'rnatish.

mysql_pconnect() funksiyasi

mysql_connect() funksiyasi MySQL tizimi bilan bog'lanishni o'rnatadi. Sintaksisi:

```
int mysql_pconnect ([string xost [:port][:soketga yo'l] [, string  
foydalanuvchi_nomi] [,string parol])
```

yoki

```
resource mysql_pconnect ( [server satri [,username satri[,password  
satri
```

```
[, int new_link [, int client_flags]]])
```

xost parametrinda MySQL tizimining vakolatlar jadvalida ko'rsatilgan kompyuter xosti nomi beriladi. Bu nom MySQL ishlayotgan Web-serverga so'rovlarni yo'naltirish uchun ishlatiladi, biroq MySQL tizimiga masofadan turib ham bog'lanish mumkin. Xost nomi berilganda shart bo'lmagan parametrlari – port nomeri, soket(lokal xost uchun) yo'lini ham ko'rsatish mumkin. foydalanuvchi_nomi va parol parametrlari MySQL vakolatlari jadvalida berilgan foydalanuvchi nomi va paroli bilan mos tushishi kerak. Agar xost parametri berilmagan bo'lsa, mysql_pconnect() lokal xost bilan aloqani o'rnatishga harakat qiladi.

11.3-misol. MySQL bilan ochiq bog'lanish:

```
@ $db = mysql_pconnect("localhost", "ilyosbey", "135");
```

Berilgan misolda localhost – kompyuter xost nomi, ilyosbey-foydalanuvchi nomi, 135 esa – parol. mysql_pconnect() funksiyasi oldidan ishlatiladigan @ belgisi bog'lanish urinishi muvaffaqiyatsiz yakunlanganda chiqadigan xatolarni yashirish uchun ishlatiladi. Bu belgi die() funksiyasi chiqazadigan xatolar haqidagi xabarlarni almashtiradi.

Agar funksiyaga bir xil avvalgi parametrlar bilan ikki marta murojaat etilsa, yangi bog'lanish o'rnatilmaydi, lekin murojaat (ssilka)ni eski bog'lanishga qaytaradi. Bu holatga duch kelmaslik uchun,

new_link parametri ishlatiladi. Bu parametr ixtiyoriy holatda yana bir bog'lanishni ochishga yordam beradi.

client_flags parametri — bu quyidagi konstantalarni kombinatsiyasi: MYSQL_CLIENT_COMPRESS (siqish protokolini ishlatadi), MYSQL_CLIENT_IGNORE_SPACE (funksiya nomidan so'ng probellarni o'qish uchun imkoniyat yaratadi), MYSQL_CLIENT_INTERACTIVE (bog'lanish yopilguncha- wait_timeout bilan birga — interactive_timeout sekund kutadi).

new_link parametri PHP 4.2.0-versiyasidan, client_flags parametri esa PHP 4.3.0-versiyasidan boshlab qo'shilgan.

Tizim bilan bog'lanish agar mysql_close() funksiyasi yordamida yopilmagan bo'lsa, skript bajarilishi tugatilganda yopiladi.

11.4-misol. ilyosbey foydalanuvchisi uchun «135» paroli bilan lokal serverdagi ma'lumotlar bazasiga bog'lanishni o'rnatamiz:

```
<? $conn = mysql_pconnect("localhost","ilyosbey", "135")
or die ("Bog'lanish o'rnatilmadi: ". mysql_error());
echo "Bog'lanish o'rnatildi"; ?>
Natija: Bog'lanish o'rnatildi
```

mysql_connect amali quyidagi buyruq satriga teng kuchli

```
shell>mysql -u ilyosbey -p135
```

Agar dastur turli xostlarda bir nechta MySQL tizimlari bilan bog'lanishni o'rnatasa, kerakli MySQL tizimiga mysql_pconnect() funksiyasi qaytargan bog'lanish identifikatordan foydalanib so'rovlarni yo'llash mumkin.

11.5-misol.

```
<? $link1 = @mysql_pconnect("localhost", "ilyosbey", "135")
or die("localhost serveriga bog'lana olmadi!");
$link2 = @mysql_pconnect("www.someotherhost.com","usr", "secret")
or die("Could not connect to MySQL server!"); ?>
```

\$link1 va \$link2 identifikatorlari ma'lumotlar bazasiga so'rovlar bilan oxirgi murojaatni uzatadi.

Ko'p foydalanuvchilar muhitida tizim resurslarini tejash uchun mysql_pconnect() ni mysql_connect() bilan birgalikda ishlatish tavsiya etiladi. mysql_pconnect() funksiyasining parametr tiplari va qaytaradigan qiymatlari mysql_connect() funksiyasi bilan mos tushadi, bittagina farqi mysql_connect() funksiyasi ma'lumotlar bazasiga doimiy bog'lanib turish uchun ishlatiladi.

MySQL tizimida ishlash uchun ma'lumotlar bazasini tanlash.

MySQL tizimiga muvaffaqiyatli bog'lanib bo'lgandan so'ng, tizimda mavjud so'rovlar olib boriladigan ma'lumotlar bazasi tanlanadi. Bizning ma'lumotlar uquv ma'lumotlar bazasida saqlanmoqda. MySQL ma'lumotlar bazasidan foydalanish uchun avval uquv ma'lumotlar bazasi tanlanadi (aktivlashtiriladi). MB ni tanlash uchun buyruqlar satrida use buyrug'idan foydalangan edik (mysql>use uquv;):

PHP da buning uchun mysql_select_db funksiyasi ishlatiladi.

mysql_select_db() funksiyasi

Sintaksisi:

```
int mysql_select_db (  
    string ma'lumotlar_bazasi_nomi  
    [, int bog'lanish_identifikatori])
```

ma'lumotlar_bazasi_nomi parametrida tanlanishi lozim bo'lgan ma'lumotlar bazasi ko'rsatiladi, bog'lanish identifikatori parametrida esa bir nechta ochiq bog'lanishlardan ma'lumotlar_bazasi_nomi nomli bazaning bog'lanish identifikatori ko'rsatiladi.

Bu funksiya ma'lumotlar bazasini muvaffaqiyatli tanlaganda TRUE ni, aks holda FALSE ni qaytaradi.

11.6-misol. Uquv ma'lumotlar bazasini tanlaymiz:

```
<? @mysql_connect("localhost","ilyosbey","135")  
or die ("Bog'lanish o'rnatilmadi");  
echo "Bog'lanish o'rnatildi<br>";  
@mysql_select_db("uquv","link1")  
or die ("uquv ma'lumotlar bazasi tanlanmadi!"); ?>
```

Ma'lumotlar bazasiga bog'lana olmaganida yoki ma'lumotlar bazasini tanlamaganida brauzerda quyidagicha xabarlar chiqaziladi.

```
Natija: Bog'lanish o'rnatildi  
uquv ma'lumotlar bazasi tanlanmadi!
```

Agar dasturda faqatgina bitta ma'lumotlar bazasi tanlanayotgan bo'lsa, uning identifikatorini saqlash shart emas. Lekin bir nechta ma'lumotlar bazasini tanlashda ularning qaytaradigan identifikatorlari saqlanishi kerak. Chunki, keyinchalik kerakli bazaga so'rovlarni qayta ishlash uchun jo'natish imkoniyati mavjud bo'lsin. Agar identifikator ko'rsatilmagan bo'lsa, oxirgi tanlangan ma'lumotlar bazasi ishlatiladi.

mysql_close() funksiyasi

mysql_close() funksiyasi shart bo'lmagan bog'lanish_identifikatori parametri bilan bog'lanishni yopadi. Agar parametr berilmagan bo'lsa, mysql_close() funksiyasi oxirgi ochiq bog'lanishni yopadi.

Sintaksisi:

```
int mysql_close ([int bog'lanish_identifikatori])
```

mysql_pconnect() funksiyasi yordamida o'rnatilgan ochiq bog'lanishni yopish shart emas.

11.3. Tanlangan bazada so'rovlarni qayta ishlash

Tanlangan bazada so'rovlarni qayta ishlash quyidagi tartibda amalga oshiriladi:

- a) ma'lumotlarni qo'shish uchun interfeys yaratish;
 - b) ma'lumotlar bazasiga ma'lumotlarni yozish;
 - c) ma'lumotlar bazasida saqlanayotgan ma'lumotlarni olish.
- a) Ma'lumotlarni qo'shish uchun interfeys yaratish

Demak, bizda ma'lumotlar bazasi va unda qandaydir jadvallar mavjud. Bu jadvallarga ma'lumotlarni qo'shish interfeysini yaratish uchun uni strukturasi (ya'ni maydonlar to'plamini) html-formada tasvirlash kerak.

Ushbu masalani quyidagicha qism masalalarga bo'lib chiqamiz:

- jadval maydonlari ro'yxatini olish;
- html-formada maydonlarni tasvirlash.

Shundan so'ng, formaga kiritilgan ma'lumotlarni ma'lumotlar bazasiga yozish mumkin. Ushbu masalalarni tartib bilan ko'rib chiqamiz.

Jadval maydonlari ro'yxatini olish

Jadval maydonlari ro'yxatini qanday olish mumkin? Juda oddiy. PHP da bu holat uchun o'zining buyrug'i bor - mysql_list_fields.

mysql_list_fields() funksiyasi

Sintaksisi:

```
resource mysql_list_fields (  
    string database_name,  
    string table_name  
    [,resource link_identifier])
```

Bu funksiya database_name ma'lumotlar bazasida joylashgan table_name jadvalidagi maydonlar ro'yxatini chiqazadi. Bu funk-

siya ishining natijasi — resurs tipidagi o'zgaruvchi. Bu nomi, tipi va bayrog'ini o'zida jamlagan jadval maydonlari haqidagi ma'lumotlarni olish uchun ishlatish mumkin bo'lgan murojaat (ssilka).

**mysql_field_name(), mysql_field_len(), mysql_field_type(),
mysql_field_flags() funksiyalari**

mysql_field_name funksiyasi so'rov bajarilishi natijasida olingan maydon nomini chiqazadi. mysql_field_len funksiyasi maydon uzunligini chiqazadi. mysql_field_type funksiyasi maydon tipini, mysql_field_flags funksiyasi esa maydon bayroqlari ro'yxatini probel yordamida ajratib chiqazadi. Maydon tipi int, real, string, blob bo'lishi mumkin. Bayroqlar esa not_null, primary_key, unique_key, blob, auto_increment va h.k. bo'lishi mumkin.

Bu buyruqlarning barchasini sintaksisi bir xil:

```
string mysql_field_name (  
    resource result, int field_offset)  
string mysql_field_type (  
    resource result, int field_offset)  
string mysql_field_flags (  
    resource result, int field_offset)  
string mysql_field_len (  
    resource result, int field_offset)
```

Bu yerda result — so'rov (masalan, mysql_list_fields yoki mysql_query (bu haqida keyinroq to'xtalamiz) funksiyalari yordamida jo'natilgan so'rov) natijasi identifikatori, field_offset esa — so'rov natijasidagi ustunning tartib nomeri.

Umuman olganda mysql_query yoki mysql_list_fields tipidagi funksiya natijasi sifatida jadval, aniqroq aytganda unga ko'rsatkichni chiqazadi. Bu jadvaldan aniq qiymatlarni olish uchun jadvalni satrma-satr o'qiy oladigan maxsus funksiyalarni ishlatish (zadeystvovat) kerak. Shunday funksiyalardan biri mysql_field_name va b. So'rov bajarilishi natijasidagi jadvaldan barcha satrlarni saralash uchun bu jadvaldagi satrlar sonini bilish kerak. mysql_num_rows (resurs result) buyrug'i — bir qancha result natijalarining satrlari sonini chiqazadi.

Misol sifatida o'quv ma'lumotlar bazasining jadvallari bilan ishlaymiz. O'quv ma'lumotlar bazasidagi univercity jadvalidagi ustunlar ro'yxatini olishni ko'ramiz.


```

<? $conn = mysql_connect("localhost","ilyosbey", "135")
or die("Bog'lanish o'rnatilmadi: ". mysql_error());
echo "Bog'lanish o'rnatildi";
mysql_select_db("uquv") or die("tanlanmadi");
$list_f = mysql_list_fields ("uquv","univercity",$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 "<br>Maydon nomi: ". $name_f;
    echo "<br>Maydon tipi: ". $type;
    echo "<br>Maydon uzunligi: ". $len;
    echo "<br>Maydon bayrog'i satri: ".
        $flags_str . "<br>"; } ?>

```

html-formada maydonlar ro'yxatini aks ettirish

Yuqoridagi misolga ozgina o'zgartirish kiritamiz. Maydon ma'lumotlarini oddiy holda chiqazmaymiz, balki ularni html-formaning mos elementlarida aks ettiramiz. Buning uchun, BLOB tipidagi elementni textarea ga o'tkazamiz, son va satrlarni matnli satrlarni kiritish <input type=text> elementida aks ettiramiz, avtoincrement metkasi bor elementni esa chiqazmaymiz, chunki bu element qiymati avtomatik tarzda qo'yiladi.

Ushbu html-forma yordamida foydalanuvchi avtomatik tarzda univercity (universitet) va subject (o'quv fani) jadvallariga ma'lumotlarni kiritishi mumkin.

Bayroqlar ro'yxatidan auto_increment bayrog'ini ajratib olish kerak. Buning uchun explode funksiyasidan foydalanamiz.

Sintaksisi:

```

array explode( string ajratuvchi,
    string satr [, int limit])

```

Bu funksiya satr satrini ajratuvchi ajratuvchisi yordamida qismlarga ajratadi va satrdan hosil bo'lgan massivni chiqazadi.

Bizning misolda ajratuvchi sifatida probelni (« »), ajratish uchun satr sifatida esa maydon bayrog'i satrini olamiz.

Demak, univercity jadvaliga ma'lumotlar kiritish uchun forma yaratamiz:

11.8.1-listing. new_univ_subject.php fayli. uquv ma'lumotlar bazasining univercity jadvaliga ma'lumotlar kiritish formasi

```
<? $conn=mysql_connect("localhost","ilyosbey","135");
// bog'lanish o'rnatamiz
echo "<form method=post action=new_univ_subject.php>";
$j_names=array ("univercity", "subject");
$s = "Jadvalni tanlang: <select name=jadval >";
// jadvallar ro'yxatni yaratamiz
foreach ($j_names as $k => $name) {
    $s .= "<option value=$name>$name";
    // jadvallar ro'yxatga jadvallarni qo'shamiz
}
    $s .= "</select><br>";
    echo $s; //jadvallarga kiritiladigan ma'lumotlar soni ko'rsatiladi
    echo "Jadvallarga kiritilishi lozim bo'lgan satrlar soni: <input type=input
name='soni' size='5'>";
    echo "<input type=submit name='tan' value='Jadvalni tanlang'>";
    echo "</form>";
    if (isset($jadval) and isset($soni)) {
        $database = "uquv";
        $table_name = $jadval;
        ....mysql_select_db($database); // ishlash uchun MB ni tanlaymiz
        $list_f = mysql_list_fields($database,$table_name);
        // bazadagi maydonlar ro'yxatini olamiz
        $n = mysql_num_fields($list_f); //so'rov natijasidagi satrlar soni
        echo "<form method=post action=insert_univ_subject.php>";
        // ma'lumotlarni kiritish uchun forma yaratamiz
        echo "&nbsp;<TABLE BORDER=0 CELLSPACING=0 width=50% >
<tr> <td bgcolor='#005533' align=center><font color='#FFFFFF'>
<b> $table_name ga yangi satr qo'shish</b></font></td></tr>
<tr><td></td></tr></TABLE>";
        for($k=0; $k<$soni; $k++)
        {
            echo $k+1 . " - universitet";
```



```

switch ($type){
    case "string": $w = $len/5; echo "<td> <input type=text name=\"\$q_
value\" size = $w ></td>"; break;
    case "int": $w = $len/4; echo "<td><input type=text name=\"\$q_
value\" size = $w ></td>"; break;
    case "blob": echo "<td><textarea rows=6 cols=60 name=\"\$q_val-
ue\" ></textarea></td>"; break;
    case "date": $w = $len; echo "<td><input type=text name=\"\$q_
value\" size = $w >(YYYY-OO-KK)</td>"; break; } }
echo "</tr>"; }
echo "</table>"; }
echo '<input type=hidden name="jadval" value ='. $table_name.' >
<input type=hidden name="soni" value = '.$soni.' >';
echo "<input type=submit name='add' value='Qo'shish'>";
echo "</form>"; } ??

```

b) Ma'lumotlar bazasiga ma'lumotlarni yozish

Ma'lumotlarni ma'lumotlar bazasiga kiritish uchun forma yarattildi. Endi ushbu formadagi ma'lumotlar ma'lumotlar bazasiga jo'natiladi. SQL tilining INSERT buyrug'idan foydalanib, MB dagi jadvalga formadan olingan ma'lumotlarni yozamiz. Masalan:

```

mysql> INSERT INTO Student
SET name='Islomov';

```

Savol tug'iladi, bunday buyruqni (yoki SQL ning ixtiyoriy boshqa buyrug'ini) PHP skriptida qanday qo'llash mumkin. PHP da so'rovlarni ma'lumotlar bazasiga jo'natish uchun mysql_query() funksiyasidan foydalaniladi.

mysql_query() funksiyasi

Sintaksisi:

```

resource mysql_query ( string so'rov
[,resource bog'lanish_identifikatori])

```

Bu yerda, so'rov parametri SQL tilidagi so'rov matnlaridan tashkil topgan. mysql_query() funksiyasi shart bo'lmagan parametr bog'lanish_identifikatori yordamida ko'rsatilgan MySQL tizimining aktiv ma'lumotlar bazasiga yoki bu parametr tushirib qoldirilgan holda, oxirgi ochiq bog'lanishga ega ma'lumotlar bazasiga SQL-so'rovlarni jo'natadi. Agar ochiq bog'lanish mavjud bo'lmasa, funk-siya MBBT ga bog'lanishga urinib ko'radi. mysql_query() funksiyasi

ishining natijasi faqatgina bitta ifodadan (TRUE yoki FALSE dan) iborat bo'ladi.

Izoh: so'rov satri nuqta vergul bilan tugallanmasligi kerak.

mysql_query() funksiyasiga murojaat etganda, so'rov tipiga qarab funksiya turli natijalarni beradi. Faqatgina SELECT, SHOW, EXPLAIN, DESCRIBE so'rovlari uchun, agar so'rov bajarilmasa mysql_query() ko'rsatkichi so'rov natijasi sifatida FALSE ni qaytaradi. Agar foydalanuvchining so'rovda ko'rsatilgan jadval bilan ishlashi uchun huquqi yetarli bo'lmasa, mysql_query() bu holni ham xato hisoblab, FALSE ni qaytaradi.

Boshqa hollarda agar so'rov muvaffaqiyatli bajarilsa, mysql_query() TRUE ni qaytaradi. FALSE ga teng bo'lmagan qiymat so'rov muvaffaqiyatli bajarilganligini bildiradi. SELECT buyrug'i muvaffaqiyatli bajarilganda, funksiya natija identifikatorini qaytaradi va ushbu identifikator oxirgi tahrirlash va so'rov natijalarini tasvirlash uchun mysql_result() funksiyasiga uzatiladi.

Demak, endi biz ma'lumotlar bazasiga satrlarni qo'shish uchun qanday qilib so'rov jo'natishni bilamiz. Esga olamiz, oldingi misolda forma elementlarini jadval maydonlari nomlari bilan atadik. Shuning uchun insert_univ_subject.php skriptida forma ma'lumotlarini \$_POST['maydon_nomi'] o'zgaruvchisi sifatida qayta ishlash mumkin.

11.8.2-listing. insert_univ_subject.php fayli. uquv ma'lumotlar bazasining univcity jadvaliga yangi talaba ma'lumotlarini yozish.

```
<?
$conn=mysql_connect("localhost","ilyosbey", "135");
// bog'lanishni o'rnatamiz
$dbase = "uquv";
$table_name = $jadval;
$soni= $soni;
mysql_select_db($dbase); // MB ni tanlaymiz
$list_f = mysql_list_fields($dbase,$table_name);
// bazadagi jadvalning maydonlar ro'yxatini olamiz
$n = mysql_num_fields($list_f);
// oldingi so'rov natijasidagi so'rovlar soni
```

```

// jadvalning barcha maydonlari uchun bitta so'rov tuzamiz
for($k=0; $k<$soni; $k++)
{ $sql[$k] = "INSERT INTO $table_name SET ";
for($i=0;$i<$n; $i++){
    $name_f = mysql_field_name ($list_f,$i);
    // maydon nomini olamiz
    $flags_str = mysql_field_flags ($list_f, $i);
    $flags = explode(" ", $flags_str);
    foreach ($flags as $f){
        if ($f == 'auto_increment') $key = $name_f;
        // avtoinkrement nomini eslab qolamiz
    }
    if ($key <> $name_f){
        $z=$_POST[$name_f]; $kl=0;
        //massivdagi qiymatlarni olib so'rov tuzamiz
        foreach ($z as $qiymatlar)
        $kl=0; //massivdagi qiymatlarni olib so'rov tuzamiz
        foreach ($z as $qiymatlar) {
            if ($kl==$k) $svalue=$qiymatlar; $kl++;    }
            $j = $i + 1;
            $sql[$k] = $sql[$k] . ' ' . $name_f . "=" . $svalue . " ";
            // $sql satriga nomi-qiymati justligini yozamiz
            if ($j <> $n) $sql[$k] = $sql[$k] . ", ";
            // agar maydon ro'yxatda oxirgi bo'lmasa vergul qo'yamiz } }
            $sql[$k]=$sql[$k];
            $result[$k] = mysql_query($sql[$k],$conn); // so'rovni jo'natamiz
            // so'rov muvaffaqiyatli bajarilganligi haqida xabar chiqazamiz
            if (!$result[$k]) echo " ($table_name)qo'shilmadi <br> ";
            else echo "so'rov muvaffaqiyatli bajarildi!<br>";    }; ?>

```

Xuddi shu kabi, subject (o'quv fani) jadvaliga ham ma'lumotlarni kiritamiz. univercity (universitet) va subject (o'quv fani) jadvaliga ma'lumotlar kiritildi. Endi student (talaba), lecturer (o'qituvchi), exam_marks (imtihon bahosi), subj_lect (o'qituvchining o'quv disiplinalari) jadvallariga ma'lumotlar kiritish uchun alohida-alohida forma yaratamiz. Chunki bu jadvallarda boshqa jadvallardan olinadigan qiymatlar mavjud.

11.9.1-listing. new_talaba.php fayli. uquv ma'lumotlar bazasining student jadvaliga ma'lumotlar kiritish formasi

```
<? $conn=mysql_connect("localhost","ilyosbey", "135");// bog'lanishni
o'rnatamiz
$database = "uquv";
$table_name = $jadval;
$soni= $soni;
mysql_select_db($database); // MB ni tanlaymiz ?>
<html> <head>
<title>Talabalar portalining O'quv MB siga Talabalar haqidagi
ma'lumotlarni kiritish</title> </head>
<body>
<h3>O'quv MB siga Talabalar haqidagi ma'lumotlarni kiritish</h3>
<form action="insert_talaba.php" method="post">
<table border=0> <tr>
<td bgcolor='#C2E3B6'>Talabaning identifikator nomeri:</td>
<td><input type=text name=stud_id maxlength=13 size=13></td>
</tr>
<tr> <td bgcolor='#C2E3B6'>Familiyasi:</td>
<td> <input type=text name=surname maxlength=30 size=30></td>
</tr>
<tr> <td bgcolor='#C2E3B6'>Ismi:</td>
<td> <input type=text name=name maxlength=30 size=30></td>
</tr>
<tr> <td bgcolor='#C2E3B6'>Stipendiyasi:</td>
<td> <input type=text name=stipend maxlength=30 size=30></td>
</tr>
<tr> <td bgcolor='#C2E3B6'>Kursi:</td>
<td> <select name=jadval>
<option value="1">Birinchii kurs</option>
<option value="2">Ikkinchi kurs</option>
<option value="3">Uchinchi kurs</option>
<option value="4">To'rtinchi kurs</option>
</select> </td> </tr>
<tr> <td bgcolor='#C2E3B6'>Manzili:</td>
<td> <input type=text name=city maxlength=30 size=30></td>
```

```

</tr>
<tr> <td bgcolor='#C2E3B6'>Tug'ilgan sanasi:</td>
<td> <input type=text name=birthday maxlength=30 size=30>(YYYY-
OO-KK)</td> </tr>
<tr> <td bgcolor='#C2E3B6'>Universiteti:</td>
<td> <?
echo "<select name=univ_id>"; // universitetlar ro'yxatni yaratamiz
$univ=mysql_query("select univ_id, univ_name from univcity");
$univ_cnt=mysql_num_rows($univ);
for($i=0;$i<$univ_cnt;$i++)
{ $univ_result=mysql_fetch_array($univ);
echo "<option value=".$univ_result["univ_id"].">".$univ_result["univ_
name"]; }
echo "</select><br>"; ?> </td> </tr>
<tr><td colspan=2><input type=submit value="Kiritish"></td>
</tr> </table> </form>
</body> </html>

```

O'quv MIB siga Talabalar haqidagi ma'lumotlarni kiritish

Talabaning identifikator nomeri:	
Familiyasi:	
Ismi:	
Stipendiyasi:	
Kursi:	Birinchi kurs
Manzili:	
Tug'ilgan sanasi:	(YYYY-OO-KK)
Университети:	Toshkent davlat pedagogika universiteti Toshkent davlat pedagogika universiteti Texnika davlat universiteti Andijon davlat universiteti Termez davlat universiteti Samarqand davlat universiteti O'zbekiston Milliy universiteti Toshkent davlat axborot universiteti
Kiritish	

11.6-rasm. Student jadvaliga talabalar haqidagi ma'lumotlarni kiritish formasi

11.6-rasmdagi formada kiritilgan ma'lumotlarni ma'lumotlar bazasiga yozish 11.9.2-listingda ko'riladi.

11.9.2-listing. insert_talaba.php fayli. uquv ma'lumotlar bazasi-ning student jadvaliga yangi talaba ma'lumotlarini yozish

```
<html>
<head> <title>O'quv MB ga ma'lumotlarni kiritish</title> </head>
<body> <h3>O'quv MB ga ma'lumotlarni kiritish</h3>
<? if (!$stud_id || !$name || !$surname || !$kurs || !$univ_id)
{ echo "Ko'rsatilgan maydonlar to'ldirilmagan. To'ldirilishi zarur
bo'lgan maydonlarga ma'lumotlarni kiriting va qayta urinib ko'ring." ;
exit; }
@ $db = mysql_pconnect("localhost", "ilyosbey", "135");
if (!$db) { echo "Ma'lumotlar bazasiga bog'lana olmadi . " ; exit; }
mysql_select_db( "uquv" ) ;
$query = "insert into student values (". $stud_id. ", ". $surname. ", ". $sna
me. ", ". $stipend. ", ". $kurs. ", ". $city. ", ". $birthday. ", ". $univ_id. ")";
$result = mysql_query($query);
if ($result) echo mysql_affected_rows( ). "talaba ma'lumoti bazaga yozil-
di."; ?> </body> </html>
```

Xuddi shu kabi, boshqa jadvallarga ham ma'lumotlar kiritish interfeysini yaratish lozim.

Demak, ma'lumotlarni kiritish masalasini Web-interfeys yordamida hal etdik, lekin masalaning bunday hal etilishi hamma vaqt ham samara bermaydi. Masalan, ba'zi jadvallarda maydon qiymatlarini boshqa bir jadvallardan olishga to'g'ri keladi yoki ma'lumotlarni bu tarzda kiritishda jadvalga bo'sh satrlar ham kirishi mumkin. Shuning uchun ma'lumotlarni kiritish masalasini yechayotgan vaqtda masalaning barcha taraflariga e'tibor qaratish lozim. Masalani to'g'ri loyihalash va uni hal etish yo'llarini topishni o'quvchiga mustaqil ravishda qoldiramiz. Biz boshqa bir masalani – MySQL MBBT da saqlanayotgan ma'lumotlar bazasidagi ma'lumotlarni chiqazishni ko'rib o'tamiz.

c) MySQL da saqlanayotgan ma'lumotlarni olish

PHP texnologiyasi yordamida ma'lumotlarni brauzerga chiqazish uchun bazadan olingan ma'lumotlar o'zgaruvchiga tenglanadi. So'ng ushbu o'zgaruvchi brauzerga chiqaziladi.

MySQL da ma'lumotlarni jadvaldan olishni amalga oshirish uchun SQL tilining SELECT buyrug'idan foydalaniladi:

```
mysql> SELECT * FROM student;
```

Oldingi ma'ruzada aytganimizdek, `mysql_query()` funksiyasi ma'lumotlar bazasiga so'rovlar (ma'lumotlarni jo'natish, olish, o'chirish, o'zgartirish kabi) bilan murojaat etish interfeysini ta'minlaydi. Shuning uchun yuqoridagi kabi so'rovni ham serverga `mysql_query()` funksiyasi yordamida jo'natish mumkin. Endi so'rovlarda barcha maydonlarni tanlash, natijani esa brauzerda chiqazish talab etiladi. Bu so'rov natijasi sifatida — jadvalning barcha qiymatlari, aniqrog'i, bu jadvalga ko'rsatkich chiqaziladi.

Biz yuqorida `mysql_field_name()` funksiyasi yordamida so'rov natijasidan maydon nomini olishni ko'rib o'tgan edik. Endi bizga maydon nomi emas, balki maydon qiymatini olish kerak bo'ladi. PHP ning bunday funksiyalari mavjud, ulardan eng ko'p ishlatiladiganlari — `mysql_result()` va `mysql_fetch_array()`.

`mysql_result()` funksiyasi

`mysql_result()` funksiyasi `mysql_query()` funksiyasi bilan birgalikda ma'lumotlar to'plamini olish uchun ishlatiladi.

Sintaksisi:

`mysql_result (resource natija_identifikatori,
butun satr(yoki yozuv) [,maydon])`

`mysql_result()` so'rov natijasi sifatida bitta yacheyka qiymatini chiqazadi.

`natija_identifikatori` parametrda `mysql_query()` funksiyasi qaytargan identifikator — jadvalga ko'rsatkich ko'rsatiladi. `satr(yoki yozuv)` parametrda `natija_identifikatori` parametrda aniqlangan ma'lumotlar to'plamidagi satrlarning qaysi biri chiqazilishi lozimligi ko'rsatiladi.

Shart bo'lmagan parametr maydon sifatida:

- jadvaldagi aralash maydonlar;
- maydon nomi;
- maydon nomi `maydon_nomi_jadval_nomi` formatida berilishi mumkin.

Agar so'rovlarni qayta ishlashda juda ko'p xotira sarflanayotgan bo'lsa va xotira yetishmovchiligi yuzaga kelsa, u holda PHP ning standart funksiyasi `mysql_free_result` ga murojaat etiladi. Unga murojaat etishda `mysql_query()` qaytargan `natija_identifikatori` uzatiladi. `mysql_free_result()` funksiyasi berilgan so'rov bilan bog'liq barcha xotirani bo'shatadi.

Sintaksisi:

```
int mysql_free_result(int result);
```

Ko'pgina hollarda SQL so'rovida INSERT, UPDATE, REPLACE yoki DELETE buyruqlari ishtirok etgan vaqtda so'rov natijasidagi satrlar sonini bilish talab etiladi. Bu masalani mysql_affected_rows() funksiyasi hal etadi.

mysql_affected_rows() funksiyasi

mysql_affected_rows() funksiyasi so'rov natijasidagi satrlar sonini aniqlash uchun ishlatiladi. mysql_affected_rows() funksiyasi SELECT buyrug'i ishtirok etgan so'rovlarda ishlamaydi. DELETE buyrug'ini WHERE siz bajarishda mysql_affected_rows() funksiyasi har doim 0 ni chiqazadi.

Sintaksisi:

```
int mysql_affected_rows ([int bog'lanish_identifikatori])
```

bog'lanish_identifikatori parametri shart bo'lmagan parametr. Agar u ko'rsatilmasa, mysql_affected_rows() oxirgi ochiq bog'lanishni ishlatishga urinib ko'radi.

11.10-misol. mysql_affected_rows() funksiyasining ishlatilishi

```
<? // serverga bog'lanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga
bog'lanib bo'lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo'lmadi!");
$query = "UPDATE student SET surname = \"Fozilov\" WHERE stud_
id = '3'"; // So'rov yaratish
$result = mysql_query($query);
print mysql_affected_rows(). " ta satr almashtirildi.";
mysql_close(); ?>
```

mysql_num_rows() funksiyasi

SELECT buyrug'i ishlatilganda chiqadigan satrlar sonini aniqlash uchun mysql_num_rows() funksiyasi ishlatiladi.

Sintaksisi:

```
int mysql_num_rows(int natija)
```

11.11-misol. Subject jadvalidagi ma'lumotlarni brauzerda jadval shaklida chiqazamiz:

11.11.1-listing.

```
<? // serverga bog'lanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga
bog'lanib bo'lmadi!");
```

```

@mysql_select_db("uquv") or die("uquv bazasini tanlab bo'lmadi!");
// subject jadvalidagi barcha yozuvlarni olish
$query = "SELECT * FROM subject";
$result = mysql_query($query);
$x = 0; print "<table>\n";
print "<tr>\n<th>Subj_id</th><th>Subj_name</th><th>Hour</th><th>Semestr</th>\n</tr>\n";
while ($x < mysql_num_rows($result)) :
    $id = mysql_result($result, $x, 'subj_id');
    $name = mysql_result($result, $x, 'subj_name');
    $hou = mysql_result($result, $x, 'hour');
    $sem = mysql_result($result, $x, 'semestr');
    print "<tr>\n";
    print "<td>$id</td>\n<td>$name</td>\n<td>$hou</td>\n<td>$sem</td>\n"; print "</tr>\n";
    $x++; endwhile; print "</table>";
mysql_close(); ?>

```

Ko'pgina so'rov natijalari bilan ishlashda natijaning butun bir satrini bir vaqtning o'zida qayta ishlaydigan funksiyalarga amal qilishga to'g'ri keladi (masalan, `mysql_fetch_row()`, `mysql_fetch_array()` va h.k.).

Alohida ustunlarni olish uchun ko'p marotaba `mysql_result()` funksiyasiga murojaat etgandan ko'ra barcha ustun yozuvlarini indeksli (0 dan boshlanuvchi indeksli) massiv elementi qiymatlariga tenglashtirish anchagina qulaylikka olib keladi. Masala `mysql_fetch_row()` funksiya yordamida hal etiladi.

mysql_fetch_row() funksiyasi

`mysql_fetch_row()` funksiyasi so'rov natijasidagi ma'lumotlar to'plamini olish uchun ishlatiladi.

Sintaksisi:

`array mysql_fetch_row (int natija_identifikatori)`

`natija_identifikatori` parametrda `mysql_query()` funksiyasi qaytargan qiymat beriladi.

11.11.2-listing. `mysql_fetch_row()` funksiya yordamida ma'lumotlarni olish

```

<? @mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga bog'lanib bo'lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo'lmadi!");

```



```

$query = "SELECT * FROM subject";
$result =mysql_query($query);
print "<table>\n";
print "<tr>\n<th>subj_id</th><th>subj_name</th><th>hour</th><th>
semestr</th>\n</tr>\n";
while ($row = mysql_fetch_row($result)) : print "<tr>\n";
print"<td>".$row["0"]."</td>\n<td>".$row["1"]."
</td>\n<td>".$row["2"]."</td>\n<td>".$row["3"]."</td>\n";
print "</tr>\n";
endwhile; print "</table>";
mysql_close(); ?>

```

mysql_fetch_array() funksiyasi mysql_fetch_row() funksiyasiga o'xshash, biroq bitta farqli tarafi odatga ko'ra maydon satrlari qiymatlari assotsiativ massiv sifatida saqlanadi. Indeksash (assotsiativ, sonli yoki kombinatsiyalangan) tipini tanlash mumkin. Bu funksiya so'rov natijalari satrini qayta ishlaydi va qayta ishlangan so'rov natijalari satrini massiv (assotsiativ, sonli yoki ikkalasi) sifatida chiqazadi yoki agar satrlar bo'lmasa, FALSE ni qaytaradi. mysql_fetch_array() — bu mysql_fetch_row() funksiyasining kengaytirilgan versiyasi. mysql_fetch_array() funksiyasi mysql_fetch_row() ga nisbatan sekin ishlamasligini sezish mumkin va ma'lumotlarga birmuncha qulay bog'lanish imkonini yaratadi.

Sintaksisi:

```

array mysql_fetch_array (int natija_identifikatori
[, indekslash_tipi])

```

natija_identifikatori parametrda mysql_query() funksiyasi qaytargan qiymat beriladi. Shart bo'lmagan indekslash_tipi parametri ham konstanta bo'lishi mumkin va quyidagi qiymatlardan birini qabul qilishi mumkin:

- **MYSQL_ASSOC** —mysql_fetch_array() funksiyasi assotsiativ massivni chiqazadi. Agar parametr ko'rsatilmasa, bu qiymat odatga ko'ra ishlatiladi;
- **MYSQL_NUM** —mysql_fetch_array() funksiyasi sonli indeksli massivni chiqazadi;
- **MYSQL_BOTH** — olinadigan massiv ham sonli indeksli, ham assotsiativ indeksli bo'lishi mumkin.

Bu imkoniyat PHP 3.0.7 versiyasidan boshlab qo'shilgan.

11.11.3-listing. mysql_fetch_array() funksiya yordamida ma'lumotlarni olish

```
<? @mysql_connect("localhost", "ilyosbey", "135") or die("MySQL
serveriga bog'lanib bo'lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo'lmadi!");
$query = "SELECT * FROM subject";
$result = mysql_query($query);
print "<table>\n";
print "<tr>\n<th>subj_id</th><th>subj_name</th>
<th>hour</th><th>semester</th> \n</tr>\n";
while ($row = mysql_fetch_array($result)) { print "<tr>\n";
print "<td>". $row["subj_id"]. "</td>\n<td>". $row["subj_name"]. "</
td>\n <td>". $row["hour"] . "</td>\n<td>". $row["semester"]. "</td>\n";
print "</tr>\n"; } print "</table>";
mysql_close(); ?>
```

Izoh: bu funksiya yordamida chiqaziladigan maydonlar nomi registrga ahamiyatli.

11.11-misolni brauzerda boshqa ko'rinishda tasvirlash.

11.11.4-listing. 1-usul.

```
<?
$conn=mysql_connect("localhost","ilyosbey", "135");
$dbase = "uquv";
$table_name = "subject";
mysql_select_db($dbase);
$list_f = mysql_list_fields($dbase,$table_name);
$num = mysql_num_fields($list_f);
// $names massivida maydon nomlarini saqlaymiz
for($j=0;$j<$num; $j++){
    $names[] = mysql_field_name ($list_f,$j);
}
$sql = "SELECT * FROM $table_name"; // SQL so'rovini yaratamiz
$q = mysql_query($sql,$conn) or die(); // serverga so'rovni jo'natamiz
$num = mysql_num_rows($q); // natija satrlari sonini olamiz
// HTML-jadvallni yozamiz
echo "&nbsp;   <TABLE BORDER=0 CELLPACING=0 width=90%
align=center><tr><TD BGCOLOR='#005533' align=center> <font
color='#FFFFFF'><b>$table_name</b></font></td> </tr></TABLE>";
```



```

echo "<table cellpadding=0 cellspacing=1 border=1
width=90% align=center>";
// omaydon nomlarini chiqazamiz
echo "<tr>";
foreach ($names as $val){
echo "<th ALIGN=CENTER BGCOLOR='#C2E3B6'>
<font size=2>$val</font></th>"; }
// maydon qiymatlarini chiqazamiz
echo "</tr>";
for($i=0;$i<$n; $i++){ // tanlash sorovi natijasidagi barcha satrlarni nav-
bat bilan olamiz
echo "<tr>";
foreach ($names as $k => $val) { // barcha maydon nomlarini olamiz
$value = mysql_result($q,$i,$val); // maydon qiymatlarini olamiz
echo "<td><font size=2>&nbsp;$value</font></td>";
// maydon qiymatlarini chiqazamiz
}
echo "</tr>"; }
echo "</table>"; ?>

```

Xuddi shuni `mysql_fetch_array()` yordamida amalga oshiramiz:
11.11.5-listing. 2-usul.

```

<? $conn=mysql_connect("localhost","ilyosbey", "135");
$dbase = "uquv";
$table_name = "subject";
mysql_select_db($dbase);
$list_f = mysql_list_fields($dbase,$table_name);
$num = mysql_num_fields($list_f);
// $names massivida maydon nomlarini saqlaymiz
for($j=0;$j<$num; $j++){
    $names[] = mysql_field_name ($list_f,$j); }
$sql = "SELECT * FROM $table_name"; // SQL so'rovini yaratamiz
$q = mysql_query($sql,$conn) or die(); // serverga so'rovni jo'natamiz
$num = mysql_num_rows($q); // natija satrlari sonini olamiz
// HTML-jadvallni yozamiz
echo "&nbsp;<TABLE BORDER=0 CELLSPACING=0 width=90%
align=center><tr><TD BGCOLOR='#005533' align=center>

```



```

<font color='#FFFFFF'><b>$table_name</b></font></td>
</tr></TABLE>";
echo "<table cellpadding=0 cellspacing=1 border=1
width=90% align=center>";
// maydon nomlarini chiqazamiz
echo "<tr>";
foreach ($names as $val){
echo "<th ALIGN=CENTER BGCOLOR='#C2E3B6'>
<font size=2>$val</font></th>";
} // maydon qiymatlarini chiqazamiz
echo "</tr>";
// yuqorigi qismni avvalgi misol bilan bir xil
for($i=0;$i<$n; $i++){
// maydon qiymatlarini assotiativ massiv sifatida olamiz
while($row = mysql_fetch_array($q, MYSQL_ASSOC)) { echo "<tr>";
foreach ($names as $k => $val){
echo "<td><font size=2>&nbsp;$row[$val]</font></td>";
// maydon qiymatlarini chiqazamiz
}
echo "</tr>"; } }
echo "</table>"; ?>

```

Natija. Ikkala usulda ham natija bir xil chiqadi

Ma'lumotlar bazasini yaratish va o'chirish

PHP-sahifasini ishlatib MySQL da yangi ma'lumotlar bazasini yaratish uchun `mysql_create_db()` funksiyasi va ma'lumotlar bazasini o'chirish uchun — `mysql_drop_db()` funksiyasi qo'llaniladi.

Sintaksisi:

```

int mysql_create_db(string database, [int database_connection] );
int mysql_drop_db(string database, [int database_connection] );

```

Bu ikkala funksiyada parametr sifatida ma'lumotlar bazasi nomi va bog'lanish identifikatori ishlatiladi. Agar bog'lanish identifikatori bo'lmasa, oxirgi bog'lanilgan ma'lumotlar bazasi bilan ishlaydi. Bu funksiyalar ma'lumotlar bazasini yaratadi va o'chiradi, funksiya ishi muvaffaqiyatli yakunlansa, `true` qiymatini, aks holda — `false` ni qaytaradi.

3-qism. FOYDALANISH UCHUN RUXSAT

XII bob. SESSIYALAR BILAN ISHLASH

Bu bobda tarmoqdagi xavfsizlikni ta'minlash va bu uchun sessiya mexanizmlarining ishlatilishi haqida so'z boradi. Sessiyani initsializatsiyalash, foydalanuvchi identifikatorini uzatish, sessiya o'zgaruvchilarini ro'yxatdan o'tkazish, sessiyani o'chirish kabilar ko'rib o'tiladi. Hamda sessiyani php.ini, httpd.conf, .htaccess fayllarida sozlash ko'riladi. Bob so'ngida mexanizm yordamida foydalanuvchini ro'yxatdan o'tkazishga misol keltiriladi.

12.1. Foydalanish uchun ruxsat

“Foydalanish uchun ruxsat” nima? Bu tushunchani hayotiy misol yordamida tushuntirishga harakat qilamiz. Tasavvur qiling, kutubxonadan kitob olmoqchisiz. Lekin bu xizmatdan foydalanish uchun faqatgina kutubxona bileti borlargaagina ruxsat berilgan. Ushbu bilet yordamida kutubxona resurslariga “foydalanish uchun ruxsat” beriladi, deyish mumkin. Kutubxonachi kutubxona biletini ko'rgandan so'ng, kitobni kim olayotganligi haqida biladi va zarur bo'lgan hollarda (masalan, kitob anchadan beri qaytarilmagan) biror-bir cho'ra (kitobxonga telefon qiladi) ko'radi. Oddiy tashrif buyuruvchiga nisbatan kutubxonachi ko'proq huquqqa ega: u aniqlangan tashrif buyuruvchiga kitobni berishi yoki bermasligi mumkin, yangi kelgan kitoblarni ko'rsatish uchun qo'yishi va kam o'qiladigan kitoblarni arxivga olib qo'yishi mumkin va h.k.

Axborot texnologiyalarida ham hamma misollar xuddi shunday. Tarmoqda ko'p sondagi resurslar mavjud, hatto “kutubxona”dan ham ko'p. Ularning har birini o'zining «kutubxonachi»si hamda resurs tarkibiga javob beruvchi va ma'lumotni foydalanuvchiga taqdim etuvchi inson va guruhlar mavjud. Ular administratorlar deb ataladi. Administrator funksiyasi, qoidaga ko'ra yangi ma'lumotni qo'shish, mavjudlarini tahrirlash va o'chirish, foydalanuvchiga ma'lumotlarni tasvirlash usullarini sozlash kabilarni o'z ichiga oladi. Foydalanuvchi (resursga oddiy tashrif buyuruvchi) funksiyasiga esa ma'lumotlarni ko'rish va qidirish kiradi.

Foydalanuvchini administratoridan qanday ajratib olish mumkin? Haqiqiy kutubxonada bu ravshan, lekin agar kutubxonachi va foyda-

lanuvchi rolini virtual haqiqiylikka ko'chirilsa, bu aniqlik yo'qoladi. Kutubxonachi foydalanuvchi kabi kutubxona resurslariga Internet orqali kira oladi. HTTP protokoliga muvofiq barcha mijozlar mutlaqo teng huquqli. Saytga kim kirganligini qanday bilish mumkin? Oddiy foydalanuvchi (tashrif buyuruvchi)mi yoki administrator (kutubxonachi)? Agar bu oddiy foydalanuvchi bo'lsa, tashrif buyuruvchiga saytning yopiq arxividan foydalanishga ruxsat bermaslikni qanday ta'minlash mumkin? Shu o'rinda savol tug'iladi, so'rov jo'natgan mijozni qanday identifikatsiyalash kerak va u saytda ishlayotgan vaqtda u haqidagi ma'lumotlarni qanday saqlash mumkin?

Esga birinchi keladigan, eng oddiy varianti — bu sistemadagi insonni ro'yxatdan o'tkazish va unga kutubxona biletiga o'xshash sistemaning boshqaruv (administrativ) qismiga kirish uchun login va parol berish kerak. Bu ma'lumotlar kompyuter serverida saqlanadi va sistemaga kirishdan oldin foydalanuvchi kiritgan login va parolni sistemada saqlanayotgani bilan mosligi tekshiriladi. Haqiqatda, bu yerda haqiqiy kutubxona bilan taqqoslaganda vaziyat o'zgaradi: kutubxona bileti kutubxonachidan sistemaning yopiq qismiga kirish uchun talab etiladi, kitobxon esa saytga bimalol kirishi mumkin. Umuman olganda oddiy tashrif buyuruvchilarni ham ro'yxatdan o'tkazish mumkin. U holda barcha ro'yxatdan o'tgan foydalanuvchilarni guruhlariga ularni mos huquqlariga ko'ra: kutubxonachilar (administratorlar) va kitobxonlar (oddiy foydalanuvchilar) ga ajratish kerak. Biz bu nozik masalani ichiga kirmaymiz va eng oddiy saytning ba'zi bir sahifalariga kirish uchun talab etilgan login va parolni kiritish usulidan foydalanamiz.

12.1-misol. Bizda talabalar portalining Admin bo'limi mavjud bo'lib, ushbu bo'lim orqali portalga ma'lumotlar kiritish mumkin.

12.1a-listing. home.php fayli. Talabalar portalining Asosiy sahifasi

```
<html>
<head> <title>Talabalar portali</title>
<style> <!--
h1 {color: white; font-size :24p; text-align : center ; font-family : arial ,
sans-serif}
.menu {color: white; font-size: 12pt; text-align : center ; font-family : arial
, sans-serif; font-weight : bold}
```



```

    td {background: silver}
    p {color :darkblue; font-size: 12pt; text-align: justify;font-family: arial ,
sans-serif}
    p.foot {color: darkblue; font-size :9pt; text-align: center; font-family : arial
, sans-serif ; font-weight : bold}
    a: link , a : visited, a : active {color : white }
--> </style>
</head> <body>
<!-- page header -->
<table width="100%" cellpadding = 12 cellspacing =0 border = 0>
<tr bgcolor = silver>
<td align = left><img src = "logo.png "></td>
<td><h1>Talabalar portali</h1></td>
<td align = right><img src = "logo.png "></td>
</tr></table>
<!-- menu -->
<table width = "100%" bgcolor = white cellpadding = 4 cellspacing =
4>
<tr ><td width = "25%"><img src = "s-logo.png "><a href= "home.
php"> <span class=menu>Asosiy</span></a></td>
<td width = "25%"><img src = "s-logo.png "> <a href= "resurs.php">
<span class=menu>Ta'lim resurslari</span></a></td>
<td width = "25%"> <img src = "s-logo.png "> <a href= "talaba.php">
<span class=menu>Talabalar uchun</span></a></td>
<td width = "25%"> <img src = "s-logo.png "> <a href= "authorize.
php"> <span class=menu>Admin</span></a></td>
</tr></table>
<!-- page content -->
<p>Uzluksiz ta'lim kadrlar tayyorlash tizimining asosi, O'zbekiston Res-
publikasining ijtimoiy-iqtisodiy taraqqiyotini ta'minlovchi, shaxs, jamiyat
va davlatning iqtisodiy, ijtimoiy, ilmiy-texnikaviy va madaniy ehtiyojlarini
qondiruvchi ustuvor sohadir. </p>
<!-- page footer-->
<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr><td><p class=foot>&copy; Designed By SMD.</p>

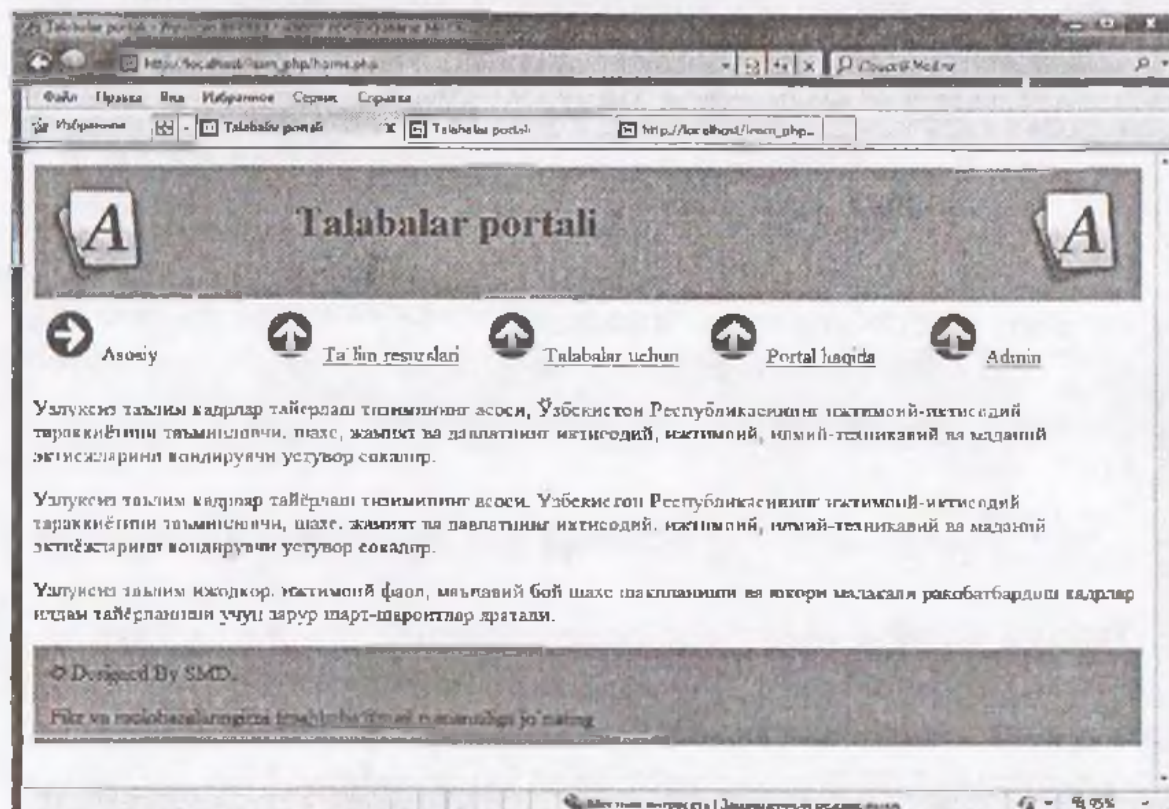
```



```

<p class=foot>Fikr va mulohazalaringizni <a href="mailto:fmahbuba@
mail.ru">fmahbuba@mail.ru</a>manziliga jo'nating </p>
</td></tr></table>
</body> </html>

```



12.1-rasm. home.php faylining brauzerdagi ko'rinishi

Administrator bo'limida faqatgina ma'lumotlar bazasiga ma'lumotlarni kiritish huquqiga ega bo'lgan shaxslargina ishlay olishadi. Bu bo'limga kirish uchun Admin menyusiga murojaat etiladi.

12.1b-listing. admin.php fayli. Talabalar portalining Admin bo'limi:

```

<html>
<head> <title>Talabalar portali</title>
<style> <!--
h1 {color: white; font-size :24p; text-align : center ; font-family : arial ,
sans-serif}
.menu {color: white; font-size: 12pt; text-align : center ; font-family : arial
, sans-serif; font-weight : bold}

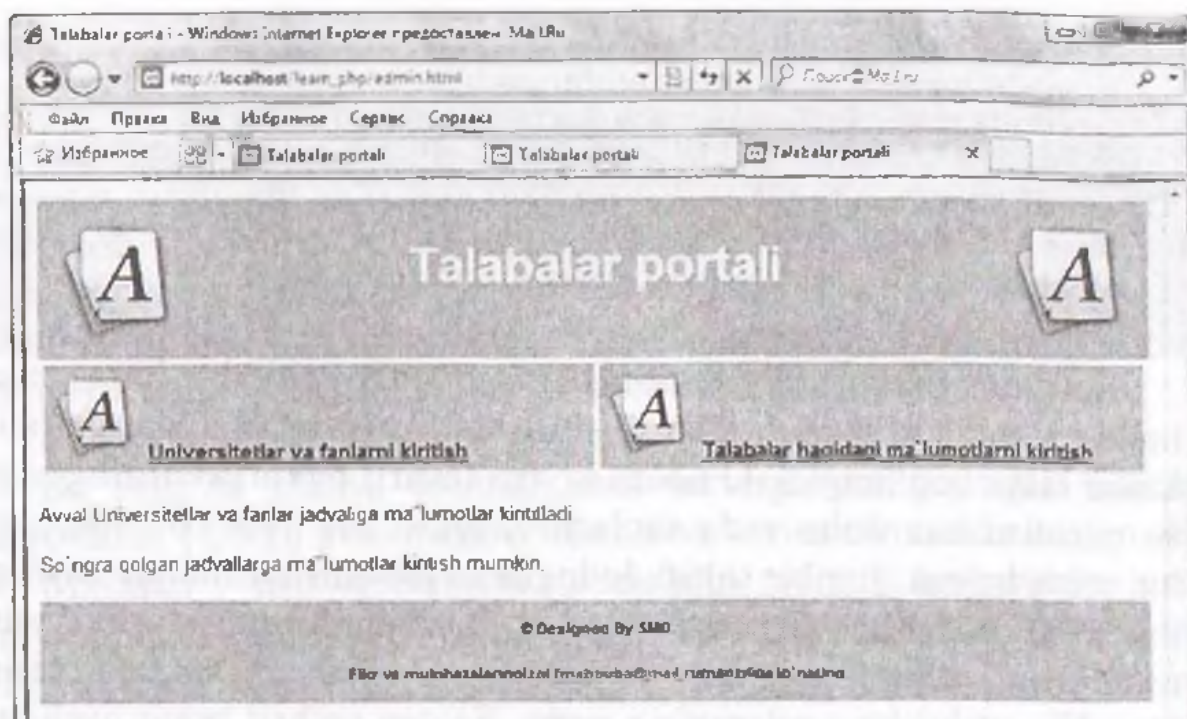
```



```

td {background: silver}
a: link , a : visited, a : active {color : white }
p {color :darkblue; font-size: 12pt; text-align: justify;font-family: arial ,
sans-serif}
p.foot {color: darkblue; font-size :9pt; text-align: center; font-family : arial
, sans-serif ; font-weight : bold}
--> </style>
</head>
<body> <!-- page header -->
<table width="100%" cellpadding = 12 cellspacing =0 border = 0>
<tr bgcolor = silver>
<td align = left><img src = "logo.png "></td>
<td> <h1>Talabalar portali</h1> </td>
<td align = right><img src = "logo.png "></td>
</tr> </table>
<!-- menu -->
<table width = "100%" bgcolor = white cellpadding = 4 cellspacing =
4>
<tr > <td width = "25%">
<img src = "s-logo.png "> <span class=menu><a href="home.
php">Asosiy</a></span></td>
<td width = "25%">
<img src = "s-logo.png "> <span class=menu><a href="new_univ_sub-
ject.php">Universitetlar va fanlarni kiritish</a></span></td>
<td width = "25%">
<img src = "s-logo.png "> <span class=menu><a href="new_talaba.
php">Talabalar haqidagi ma'lumotlarni kiritish</a></span></td>
</tr> </table>
<!-- page content -->
<p>Avval Universitetlar va fanlar jadvaliga ma'lumotlar kiritiladi. </p>
<p>So'ngra qolgan jadvallarga ma'lumotlar kiritish mumkin.</p><!--
page footer-->
<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr> <td> <p class=foot>&copy; Designed By SMD.</p>
<p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@
mail.ru">fmahbuba@mail.ru</a>manziliga jo'nating </p>
</td> </tr> </table>
</body>
</html>

```



12.2-rasm. admin.php faylining brauzerdagi ko'rinishi

Agar bu fayllarni shundayligicha qoldirilsa, u holda ixtiyoriy foydalanuvchi "Admin" gipermurojaat (ssilka)ini bosib administrator sahifasiga kirishi mumkin. Administrator sahifasidan faqatgina administratorlar foydalanishini ta'minlash uchun bu sahifaga murojaat etilganda foydalanuvchi shaxsini aniqlovchi oraliq skriptini qo'shish kerak. Administrator sahifasiga murojaat etilganda admin.html fayliga emas, balki avval authorize.php skriptiga yo'naltirilishini ta'minlash kerak.

Avtorizatsiya skripti o'zida login va parolni kirituvchi formani va ularni to'g'riligini tekshiruvchi kodni, agar tekshirish muvaffaqiyatli yakunlangan bo'lsa, uni maxfiy sahifaga yo'naltirishni, aks holda xatolik haqidagi xabarni chiqazishni jamlagan bo'lishi kerak.

12.1c-listing. authorize.php

```
<? if (!isset($_GET['go'])) {
// formadagi ma'lumotlar jo'natilganligini tekshiramiz
// avtorizatsiya uchun forma (login va parolni kiritish)
echo "<form> Login: <input type=text name=login>
Password: <input type=password name=passwd>
<input type=submit name=go value=Go> </form>";
```



```
}else { // agar forma to'ldirilgan bo'lsa, login parolni  
// to'g'ri login va parol bilan solishtiramiz  
if ($_GET['login']==="lola" && $_GET['passwd']==="123")  
{ Header("Location: admin.php");
```

```
//va maxfiy sahifaga yo'naltiramiz  
}else echo "Noto'g'ri kiritildi, qayta kiritib ko'ring<br>";  
} ?>
```

Hammasi juda oddiy. Lekin bizda maxfiy sahifalar bitta emas, balki bir nechta. Hamda ular bir-biri bilan kesishgan jo'natma (ssilka)lar bilan bog'langan. U holda saytga tashrif buyuruvchining login va parolini har doim esda saqlash zarurati tug'iladi. Bu masalani hal etish uchun, har bir sahifada login va parolni sahifadan sahifaga uzatuvchi yashirin parametrlar sifatida berilgan formani tashkil etish mumkin. Lekin bu usulda parametrlarni olish va soxtalashtirish mumkin, ya'ni bu usul xavfsiz emas. Saytga tashrif buyuruvchilarning ishlash jarayonida ularning ma'lumotlarini saqlash muammosi yechimi PHP da birmuncha qulay va xavfsiz usul bilan hal etiladi — bu sessiya mexanizmi.

12.2. Sessiya mexanizmi

Sessiya — o'zgaruvchilarni yaratish va ishlatish imkoniyatini yaratuvchi mexanizm. Bu o'zgaruvchilar foydalanuvchi sayt bilan ishlash vaqti jarayonida o'z qiymatlarini saqlaydilar.

Bu o'zgaruvchilar qiymatlari har bir foydalanuvchi uchun turlicha bo'lishi mumkin va undan foydalanuvchi sistemadan chiqib ketguncha saytning ixtiyoriy sahifasida foydalanish mumkin. Shuning uchun saytga har gal kirganda, foydalanuvchi ushbu seans davomida yoki sayt bilan ishlash sessiyasida uni identifikatsiyalash imkoniyatini beruvchi yangi o'zgaruvchi qiymatini oladi.

Foydalanuvchini identifikatsiyalash uchun har bir foydalanuvchiga sessiya identifikatori(SID, Session IDentifier) deb nomlanuvchi unikal nomer beriladi. U PHP da foydalanuvchi saytga kirgan vaqtning o'zida generatsiya qilinadi hamda o'zida 32 ta belgidan iborat satrni ifodalaydi (masalan, ac4f4a45bdc893434c95dcaffblcl811) va foydalanuvchi saytdan chiqqan vaqtda o'chiriladi. Bu identifikator serverga mijoz so'rovi bilan birgalikda uzatiladi va server javobi bilan birga ortga qaytariladi.

Sessiya identifikatorini uzatishni bir nechta yo'llari mavjud:

- cookies yordamida.

Cookies mijozni identifikatsiyalashni bir qiymatli metodi sifatida maxsus yaratilgan va o'zida http protokoli kengaytmasini ifodalaydi. Bu holda sessiya identifikatori so'rov jo'natayotgan mijoz kompyuterining vaqtinchalik faylida saqlanadi. Bu metod, shubhasiz yaxshi, lekin ko'pgina foydalanuvchilar xavfsizlik muammosi tufayli o'zlarini kompyuterida cookies yordamini o'chirib qo'yishadi.

- buyruqlar satri parametri yordamida.

Bu holatda sessiya identifikatori serverga uzatilayotgan barcha so'rovlarni (URL) avtomatik o'rnatadi va server tomonida saqlanadi.

Masalan : <http://green.nsu.ru/test.php> manzili <http://green.nsu.ru/test.php?PHPSESSID=ac4f4a45bdc893434c95dcaffb1c1811> manziliga aylantiriladi

Identifikatorni uzatishning bu usuli agar so'rov jo'natilayotgan brauzerda cookies o'chirilgan bo'lsa avtomatik tarzda ishlaydi. U har doim buyruqlar satrida parametrlarni uzatishda yetarli darajada ishonchli bo'lishi mumkin. Boshqa tomondan esa sessiya identifikatorini brauzerda saqlangan variantidan foydalanish yoki soxtalashtirish uchun ko'rish mumkin. Shunday bo'lsa ham, albatta bu muammolarni yechish mumkin. Masalan, kim 32 ta turli belgidan iborat satrni eslab qoladi? Agar sessiyalar bilan ishlashni to'g'ri tashkillashtirilsa (ularni o'chirish vaqtida), u holda hatto brauzerda saqlanayotgan sessiya nomeri ham hech narsa bermasligi mumkin.

12.3. Sessiyani sozlash

Sessiya bilan ishlashdan avval, ularni PHP interpretatori yordamida qayta ishlashni, qanday qilib to'g'ri sozlashni aniqlab olish kerak.

Odatda PHP ga sessiyalar bilan ishlash mexanizmi maxsus modullar yordamida biriktirilgan bo'ladi. Bu shuni anglatadiki, sessiya bilan ishni tashkil qilish uchun hech qanday qo'shimcha elementlarni o'rnatish kerak emas. Lekin, ushbu modulning sozlovlari qanday ekanligini bilish sessiya bilan ishlashda vujudga keluvchi xatolarni bartaraf qilishda foydalidir.

Sessiyalar bilan ishlash uchun PHP ni sozlash ham, `php.ini` fayliga yozish orqali amalga oshiriladi. Bu faylga murojaat etamiz.

Biz bilamizki, sessiya identifikatori server kompyuterida yoki mijoz kompyuterida yoki boshqa joyda saqlanadi.

php.ini dagi session.save_path parametri, sessiya ma'lumotlari serverning qayerida saqlanishi kerakligini aniqlaydi. U tufayli Windows-serverida boshqalardan ko'ra ko'proq muammolar paydo bo'ladi, chunki odatda session.save_path qiymati /tmp ga o'rnatilgan bo'ladi. Va agar serverning o'zak katalogida bunday papka bo'lmasa, u holda sessiyani ishga tushirishda xatolik yuzaga keladi.

Server ko'p miqdordagi sessiyalarni bir vaqtning o'zida qayta ishlashi mumkin va ularning barcha vaqtinchalik fayllari session.save_path parametrda berilgan katalogda saqlanadi. Agar tizim katta hajmdagi papkalar bilan yaxshi ishlamasa, u holda qism kataloglarni ishlatish qulay hisoblanadi. Buning uchun, papka nomidan tashqari parametr qiymatiga, bu papkada (N;/dir) joylashgan qism direktoriyalarni ichma-ichligini aniqlovchi son ham qo'shiladi. Modomiki, nuqta vergul PHP ni sozlash faylida izoh belgilaridan biri ekan, bu qiymatni albatta qo'shtirnoq ichiga olish kerak. Sessiya ma'lumotlarini saqlash uchun barcha direktoriya va qism direktoriyalar mustaqil ravishda yaratiladi.

Masalan: 2;/Temp shuni anglatadiki, sessiya o'zgaruvchilari c:\Temp\0\a\, c:\Temp\0\b\ va b. ko'rinishidagi papkada saqlanadi.

Mijoz tomonida ma'lumotlarni saqlash cookies yordamida amalga oshiriladi. PHP ni cookies bilan ishlashini session.use_cookies, session.cookie_lifetime parametrlari yordamida sozlanadi.

session.use_cookies parametri sessiya bilan ishlashda cookies ishlatilayotganmi yoki yo'qligini aniqlaydi. Odatda, bu opsiya ulangan (yoqilgan) bo'ladi ("1" qiymatini qabul qiladi).

session.cookie_lifetime parametrda cookies ning ishlash vaqti davomiyligi sekundlarda beriladi. Odatda, bu "0", h.k. ga teng bo'ladi. cookies dagi ma'lumotlar brauzer oynasi yopilguncha saqlanib turadi.

Bu parametrlardan tashqari, sessiya nomini aniqlovchi session.name, sessiyani avtomatik tarzda ishga tushirish imkoniyatini beruvchi session.auto_start, sessiya ma'lumotlari kodirovkasi usullarini beruvchi session.serialize_handler, va keshda hujjatlar necha minutdan so'ng iste'moldan chiqishini aniqlovchi session.cache_expire parametrlarini ko'rsatish mumkin.

Odatda, sessiya nomi (session.name) PHPSESSID kabi o'rnatilib, cookies da sessiya identifikatorida saqlanayotgan o'zgaruvchi sifatida ishlatiladi. Sessiyani avtomatik ishga tushirish o'chirilgan bo'ladi, lekin session.auto_start ni qiymatini "1" ga tenglab, uni yuklash mumkin. Sessiya ma'lumotlari kodirovkasi uchun php ishlatiladi. Keshda saqlanayotgan ma'lumotlarni iste'moldan chiqishi, 180 minutdan so'ng amalga oshiriladi.

php.ini sozlash faylida amalga oshirish mumkin bo'lgan bir nechta sozlashlar mavjud. Lekin, PHP sessiyalari bilan ishlash uchun ushbu ko'rib o'tgan sozlashlarimiz yetarli.

12.4. Sessiyalar bilan ishlash

Sessiya yaratish

Birinchidan, sessiyalar bilan ishlash uchun, ya'ni sessiya mexanizmini ishga tushirish uchun nima qilinadi (agar ular server administratori tomonidan sozlab bo'lingan bo'lsa). Agar server sozlovda session.auto_start o'zgaruvchisiga "0" qiymati o'rnatilgan bo'lsa (agar session.auto_start=1, u holda sessiya avtomatik tarzda ishga tushadi), u holda sessiya ma'lumoti ishlatilishi kerak bo'lgan ixtiyoriy skript quyidagi buyruq bilan boshlanadi:

```
session_start();
```

Bunday buyruqni olgandan so'ng, server yangi sessiya yaratadi yoki so'rov bilan uzatilgan sessiya identifikatoriga asoslangan joriy sessiyani tiklaydi. Bu qanday bajariladi? PHP interpretatori sessiya identifikatori (PHPSESSID)da saqlanayotgan o'zgaruvchini avval cookies dan, so'ng POST- va GET-so'rovlari yordamida uzatilgan o'zgaruvchidan qidiradi. Agar identifikator topilsa, u holda foydalanuvchi identifikatsiyalangan hisoblanadi, barcha URL almashtirishlar bajariladi va cookiesga yozib qo'yiladi. Aks holda foydalanuvchi yangi hisoblanadi va uning uchun yangi unikal identifikator generatsiya qilinadi, so'ng URL almashtirishlar bajariladi va cookies ga yozib qo'yiladi.

Biror-bir ma'lumotni brauzerga chiqazishda, sessiya o'zgaruvchilarini ishlatish lozim bo'lgan barcha skriptlarda session_start() buyrug'iga murojaat etadi. Bu shu bilan bog'liqki, cookies faqatgina ma'lumot ekranga chiqazguncha yozib qo'yiladi.

Joriy sessiya identifikatorini olish uchun session_id() funksiyasidan foydalaniladi.

Sessiyani ko'rsatish uchun `session_name([sessiya_nomi])` funksiyasi yordamida nom beriladi. Buni sessiyani initsializatsiya qilguncha bajariladi. Joriy sessiya nomini shu funksiya yordamida, parametrsiz murojaat etib olish mumkin: `session_name()`;

12.2-misol. Sessiya yaratish.

`home.php` fayliga o'zgartirishlar kiritamiz. Ushbu sahifada, sessiya yaratamiz va u qanday identifikator va nom olishini ko'ramiz.

```
<? session_start();  
    // yangi sessiya yaratamiz yoki joriysini tiklaymiz  
echo session_id(); // sessiya identifikatorini chiqazamiz  
// _____  
//home.php faylining 12.1a – listingdagi tarkibi  
// _____  
echo session_name(); // joriy sessiya nomini chiqazamiz. ?>
```

Agar bularni hammasini `authorize.php` faylida bajarib, administrator sahifasiga `home.php` orqali kirsak va bu oynani brauzerda yopmasak (yopilsa sessiya identifikatori o'zgaradi), o'zgaruvchi (sessiya id si va uning nomi) chiqazadigan qiymatlar bir xil bo'ladi.

Sessiya o'zgaruvchilarini ro'yxatdan o'tkazish

Sessiya jarayonida biz o'zimizning shaxsiy o'zgaruvchilarimizni (masalan, login va parolni) uzatishni va saqlashni xohlaymiz. Bunga erishish uchun o'zimizning o'zgaruvchilarni oddiygina ro'yxatdan o'tkazish kerak. Bu `session_register()` funksiyasi yordamida amalga oshiriladi.

Sintaksisi:

```
session_register(o'zgaruvchi_nomi1,  
    o'zgaruvchi_nomi2, ...);
```

E'tibor bering, qiymat emas, balki o'zgaruvchi_nomi ro'yxatga olinmoqda. O'zgaruvchilarni sessiya ishlatilayotgan ixtiyoriy sahifada bir marta ro'yxatdan o'tkazish yetarli. O'zgaruvchi nomi `session_register()` funksiyasiga \$ belgisiz uzatiladi. Barcha ro'yxatdan o'tkazilganlar shu o'rinda o'zgaruvchilar ham berilgan sessiya sayt bilan ishlash jarayonida global (ixtiyoriy sahifada foydalanish mumkin) sifatida shakllanadi.

O'zgaruvchilarni ularning qiymatlarini oddiygina assotsiativ massiv `$_SESSION` ga yozib ham ro'yxatdan o'tkazish mumkin.

```
$_SESSION['o'zgaruvchi_nomi'] =
```

'o'zgaruvchi_qiymati';

Bu massivda barcha ro'yxatdan o'tgan (global) sessiya o'zgaruvchilari saqlanadi.

Bunday o'zgaruvchilarga dostup \$_SESSION['o'zgaruvchi_nomi'] (yoki PHP 4.0.6 va undan avvalgi versiyalari uchun \$HTTP_SESSION_VARS['o'zgaruvchi_nomi']) massivi yordamida amalga oshiriladi. Agar php ni sozlashda register_globals opsiyasi yoqilgan bo'lsa, u holda sessiya o'zgaruvchilariga oddiy o'zgaruvchilarga murojaat etgan kabi murojaat etish mumkin. Masalan: \$o'zgaruvchi_nomi.

Agar register_globals=off (o'chirilgan) bo'lsa, u holda POST yoki GET metodi bilan uzatilgan o'zgaruvchilarni ro'yxatdan o'tkazish uchun session_register() dan foydalanish mumkin emas, chunki bu ishlamaydi. Umuman olganda, o'zgaruvchilarni \$_SESSION va session_register()lar yordamida ro'yxatdan o'tkazish mumkin, lekin bu ikkala metodni bir vaqtning o'zida ishlatish tavsiya etilmaydi.

12.3-misol. O'zgaruvchilarni ro'yxatdan o'tkazish.

Avtorizatsiya sahifasida foydalanuvchi tomonidan kiritilgan login va parolni ro'yxatdan o'tkazish.

12.3a-listing. authorize.php

```
<? session_start();
// yangi sessiya yaratamiz yoki joriysini tiklaymiz
if (!isset($_GET['go'])) { echo "<form>
Login: <input type=text name=login>
Password: <input type=password name=passwd>
<input type=submit name=go value=Go> </form>"; }
else { $_SESSION['login']=$_GET['login'];
// login o'zgaruvchisini ro'yxatdan o'tkazamiz
$_SESSION['passwd']=$_GET['passwd'];
/* passwd o'zgaruvchisini ro'yxatdan o'tkazamiz endi login va parol — bu
sessiya uchun global o'zgaruvchi. */
if ($_GET['login']== "lola" && $_GET['passwd']== "123") {
Header("Location: admin.php"); // admin.php sahifasiga yo'naltiramiz
} else echo "Noto'g'ri kiritildi, Qayta kiriting!<br>"; }
print_r($_SESSION); // sessiyaning barcha o'zgaruvchilarini chiqazish
?>
```


Endi biz admin.php sahifasiga va saytning ixtiyoriy boshqa sahifasiga kira olamiz, \$_SESSION massivida saqlanayotgan foydalanuvchi tomonidan kiritilgan login va parol bilan ishlay olamiz. Agar administrator sahifasini kodini quyidagicha o'zgartirsak (e'tibor bering, biz uni admin.php da o'zgartirdik):

12.3b-listing. admin.php

```
<? session_start();  
// yangi sessiya yaratamiz yoki joriysini tiklaymiz  
print_r($_SESSION);  
// sessiyaning barcha o'zgaruvchilarini chiqazamiz  
>
```

//admin.php faylining 12.1b – listingdagi tarkibi

Natijada authorize.php ro'yxatdan o'tkazilgan o'zgaruvchilar ro'yxatini va maxfiy sahifani o'zini olamiz.

Bu bizga nima beradi? Faraz qiling, xaker administrator sahifasiga kirib ma'lumotlar bazasidagi ma'lumotlarni o'zgartirmoqchi. Va u administrator sahifasi (yoki sahifalari)ni qanday nomlanishini bilib oldi. U holda u brauzer satrida uning manzilini kiritishga harakat qilib ko'radi. Bu holatlarga uchramaslik uchun, administrator sahifasida quyidagicha kod satrlarini yozish kerak:

12.3c-listing. admin.php ning 2-usuli

```
<?php session_start();  
// yangi sessiya yaratamiz yoki joriysini tiklaymiz  
print_r($_SESSION);  
// sessiyaning barcha o'zgaruvchilarini chiqazamiz  
if (!($_SESSION['login']==”lola” && $_SESSION['passwd']==123))  
// parol va loginni to'g'iligini tekshiramiz  
Header(“Location: authorize.php”); ?>  
//admin.php faylining 12.1b – listingdagi tarkibi
```

Sessiya o'zgaruvchilarini o'chirish

Sessiya o'zgaruvchilarini ro'yxatga olish (ishning barcha seanslari davomida uni global etish) dan tashqari, bunday o'zgaruvchilarni va butun sessiyaning o'chirish ham zarur.

session_unregister (o'zgaruvchi_nomi) funksiyasi joriy sessiyadagi global o'zgaruvchilarni o'chiradi. Agar ro'yxatga olish \$_SESSION (PHP 4.0.6 va undan avvalgi versiyalari uchun \$HTTP_SES-

SION_VARS) yordamida amalga oshirilgan bo'lsa, u holda unset() til konstruktsiyasi ishlatiladi. U ko'rsatilgan o'zgaruvchini o'chiradi, natijasi sifatida esa hech qanday qiymatni qaytarmaydi.

Bu qayerlarda qo'l keladi (as qotadi)? Masalan, maxfiy sahifadan chiqqandan so'ng tashrif buyuruvchi haqidagi ma'lumotlar (login va parol)ni o'chirish uchun ishlatiladi. Agar to'g'ri login va parol saqlanayotgan bo'lsa va saytga tashrifdan so'ng brauzer oynasi yopil-magan bo'lsa, u holda boshqa ixtiyoriy foydalanuvchi bu kompyu-terda yopiq axborotni o'qishi mumkin.

12.4-misol. Sessiya o'zgaruvchilarini o'chirish.

12.4a-listing. home.php fayli. home.php da sessiyada saqlanib turgan, avval kiritilgan login va parolni o'chirish uchun skript yo-zamiz:

```
<? session_start();  
session_unregister('passwd'); // parolni o'chiramiz  
unset($_SESSION['login']); // loginni o'chiramiz  
print_r($_SESSION); // sessiyaning global o'zgaruvchilarini chiqazamiz  
//home.php faylining 12.1a-listingdagi tarkibi
```

Asosiy sahifaga qaytganda sessiyada saqlanib turgan login va pa-rol o'chiriladi. Endi maxfiy sahifaga kirish uchun yana qaytadan login va parolni kiritiladi.

Sessiyaning barcha o'zgaruvchilarini tashlab yuborish uchun ses-sion_unset() funksiyasidan foydalanish mumkin.

Joriy sessiyani butunlay o'chirish uchun session_destroy() buyrug'idan foydalaniladi. Bu funksiya sessiyaning global o'z-garuvchilarining qiymatlarini tashlab yubormaydi va cookies ni ham o'chirmaydi, biroq joriy sessiya bilan bog'liq barcha ma'lumotlarni o'chiradi.

12.5-misol. Sessiya va global o'zgaruvchilarni o'chirish

```
<? session_start(); // sessiyani initsiatalash  
$test = "Sessiya o'zgaruvchisi";  
$_SESSION['test'] = $test; // test o'zgaruvchisini ro'yxatdan o'tkazamiz.  
/* agar register_globals=on bo'lsa, u holda session_register('test');  
ni ishlatish mumkin */  
print_r($_SESSION); // barcha global o'zgaruvchilarni chiqazamiz  
echo session_id(); // sessiya identifikatorini chiqazamiz
```



```
echo "<hr>";  
session_unset();//sessiyaning barcha global o'zgaruvchilarini o'chiramiz  
print_r($_SESSION);  
echo session_id();  
echo "<hr>";  
session_destroy(); // sessiyaning o'chirishini  
print_r($_SESSION);  
echo session_id(); ?>
```

Bu skript ishining natijasi sifatida uchta satr chiqaziladi: birinchisi – qiymati mavjud test elementiga ega massiv hamda sessiya identifikatori, ikkinchisi – bo'sh massiv va sessiya identifikatori, uchinchi – bo'sh massiv. Bundan ko'rinib turibdiki, sessiyaning o'chirishidan so'ng uning identifikatori ham o'chiriladi va biz o'zgaruvchilarni boshqa ro'yxatdan o'tkaza olmaymiz hamda sessiya ustida hech qanday ishni amalga oshirib bo'lmaydi.

12.5. Xavfsizlik

Sessiya mexanizmini ishlatish sistemaning to'liq xavfsizligiga javob bermaydi. Buning uchun qo'shimcha choralar ko'rilishi kerak. Sessiya bilan ishlash jarayonida yuzaga kelishi mumkin bo'lgan xavfsizlik muammosiga e'tiborimizni qaratamiz, jumladan, biz yozgan dasturlash mavzusiga.

Birinchidan, parolni turli sahifalarga uzatish xavfli, chunki uni bilib olishlari mumkin. Bundan tashqari, biz uni sessiyaning global o'zgaruvchisi sifatida ro'yxatdan o'tkazdik, demak, u mijoz kompyuterining cookies da saqlanadi. Bu esa yomon. Umuman olganda, parol va loginni yaxshisi ma'lumotlar bazasida saqlash kerak. Foydalanuvchi haqidagi ma'lumotlar "test" ma'lumotlar bazasining "users" jadvalida saqlansa ham, biz unga my_user logini va my_passwd paroli bilan kira olamiz.

Ikkinchidan, agar kimdir maxfiy sahifa uchun parollarni saralash skriptini yozsa nima qilish kerak? Bu holda avtorizatsiya sahifasida birorta begona skript bir necha marta eshik qoqib turadi. Shuning uchun bizning saytning avtorizatsiyasiga so'rov kelganligini tekshirish kerak, agar kelmasa, uni davom ettirishga yo'l qo'ymaslik kerak. So'rov qabul qilgan sahifa manzilini \$_SERVER['HTTP_REFERER'] global o'zgaruvchisi yordamida olish mumkin. Shun-

day bo'lsa ham, albatta, agar saytga kirish uchun uni buzishga jiddiy yondoshilgan bo'linsa, u holda bu o'zgaruvchi qiymatini o'zgartirish kerak. Shunga ko'ra uning qiymatini tekshirishni, saytning xavfsizligini ta'minlashning asosiy qadamlaridan biri deb hisoblash mumkin.

Uquv ma'lumotlar bazasida phpmyadmin yordamida users jadvalini yaratamiz. Bu jadval uchta maydondan iborat bo'lib, birinchi id maydoni, name maydonida foydalanuvchilar nomi va password maydonida parollari saqlanadi.

```
CREATE TABLE `users` (
  `id` INT( 6 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `name` VARCHAR( 20 ) NOT NULL ,
  `password` VARCHAR( 20 ) NOT NULL , UNIQUE ( `name` ))
ENGINE=MYISAM;
```

Va ushbu users jadvaliga bitta foydalanuvchi logini va parolini kiritamiz.

```
INSERT INTO `users` ( `id` , `login` , `password` )
VALUES ( NULL , 'lola' , '123');
```

12.6-listing. authorize.php

```
<? session_start(); // yangi sessiya yaratamiz yoki joriysini tiklaymiz
$conn = mysql_connect("localhost", "ilyosbey", "135");
mysql_select_db("uquv"); // ishchi MB sini tanlaymiz
$SERVER_ROOT = "http://localhost/learn_php/";
/* skriptlar qayerda joylashgan bo'lsa ^$SERVER_ROOT doimiy ifoda va
eregi funksiyasi yordamida jo'natilayotgan skript $_SERVER['HTTP_REFERER'] satri $SERVER_ROOT satri bilan boshlanayaptimi tekshiramiz*/
if(eregi("^$SERVER_ROOT", $_SERVER['HTTP_REFERER']))
{ if (!isset($_POST['go'])) { echo "<form method=POST >
  Login: <input type=text name=login>
  Password: <input type=password name=passwd>
  <input type=submit name=go value=Go>
  </form>"; }
else {
```

```
$sql = "SELECT login FROM users WHERE login='" . $_POST['login']
. "' AND password='" . $_POST['passwd'] . "'";
$q = mysql_query($sql,$conn); // MB ga so'rovni jo'natamiz
```



```

    $n = mysql_num_rows($q); // so'rov javobidagi satrlar soni
    if (!$n==0){
        $_SESSION['user_login']=$_POST['login'];
        // login o'zgaruvchisini ro'yxatdan o'tkazamiz
        Header("Location: admin.php"); } else
    echo "Kiritilgan login va parol noto'g'ri, qayta urunib ko'ring<br>"; }
    print_r($_SESSION); } ?>

```

Birinchi ikkita muammo hal etildi. Lekin bittasi qoldi. Agar xaker so'rov satriga biror-bir global o'zgaruvchining qiymatini (masalan, loginni) yozsa, nima qilish kerak? Umuman.olganda, agar register_globals=On bo'lsagina bu mumkin. Har holda biz global o'zgaruvchilar bilan ishlash uchun \$_SESSION massivni ishlatamiz, unga esa bunday hiylalar o'tmaydi. Shunday bo'lsa ham bu muammoni yechishga harakat qilamiz. Buning uchun parametr qiymatini solishtirishdan oldin so'rov satrini tozalash kerak. U holda avval \$user_login qiymatini tashlab yuboramiz. Keyin ushbu o'zgaruvchini yana ro'yxatdan o'tkazamiz, faqatgina yangi emas, balki avval mavjud o'zgaruvchi sifatida. Buning uchun ro'yxatdan o'tkazishda dollar belgisi tushirib qoldirilmaydi.

12.7-listing. admin.php.

```

<?php
unset($user_login); // o'zgaruvchini o'chiramiz
session_start(); // yangi sessiya yaratamiz yoki joriysini tiklaymiz
session_register($user_login);
if (!($user_login=="lola")) // loginni tekshiramiz
    Header("Location: authorize.php");
    // agar xato bo'lsa, avtorizasiya sahifasiga yo'naltiramiz
?>
//admin.php faylining 15.1b-listingdagi tarkibi

```

ADABIYOTLAR RO'YXATI

1. Aripov M., Muxammadiyev J. Informatika, informatsion texnologiyalar. (Huquqshunoslik mutaxassisliklari uchun darslik) – T., 2004.
2. Aripov M. va boshqalar. “Informatika, informatsion texnologiyalar” (Bakalavr uchun o‘quv qo‘llanma). – T., TDTU. 1–2 qism. 2003. 320–430 b.
3. Aripov M. va boshqalar “Informatika, informatsioniy texnologii” (Bakalavr uchun o‘quv qo‘llanma). – T., Rus tilida. 2003. 300 b.
4. Арипов М. и др. “Основы Internet”. – Т., Университет 2002. 194 б.
5. Aripov M. Internet va elektron pochta asoslari. – T., Universitet 2000. 126 b
6. Арипов М. Англо-русско-узбекский словарь сокращенных слов по информатике. – Т., Университет-2001. 145 б.
7. Begimqulov U.SH., Mamarajabov M.E., Tursunov S . FLASH MX dasturi va undan ta’limda foydalanish imkoniyatlari. – T., TDPU. 2006.
8. Орлов Л.В. “Web сайт без секретов”. – М., ЗАО “Новый издательский дом” – 2004.
9. А. Матросов, А. Сергеев, М. Чаунин. HTML 4.0. “БХВ–Санкт-Петербург”, – 2003.
10. Кришномурти Б., Рексфорд Дж. “Web протоколы. теория и практика”. – М., ЗАО “Издательство БИНОМ”, 2002.
11. Дрогов В.А. “JAVA SCRIPT в Web дизайне”. СПб. “БХВ-Петербург”, 2002.
12. Голиш Л.В. “Технология обучения на лекциях и семинарах в экономическом ВУЗе”. Учебное пособие, – Ташкент, ТГЭУ, 2005.
13. Максимова О.В., Невзорова В.И. “Информационные технологии для экономистов”. Учебное пособия. Ростов на Дону: – Феникс, 2004.
14. Шафран Э. Создание Web страниц: самоучитель СПб: – Питер. 2001.
15. И.Ф. Астахова, Т.В. Курченкова, Р.А. Дураков, И.С. Битюцких, Д.В. Комаров. “Web технологии с базами данных”, – Воронеж, ИПЦ ВГУ, 2008.
16. Л. Томсон, Л. Веллинг. Разработка Web приложений на PHP и MySQL: пер. с англ. – 2-е изд., испр. – М., DiaSoft, 2003.
17. Гилмор Г. PHP 4. Учебный курс. – СПб.: Питер, 2001.

18. Гончаров А. Самоучитель HTML. — СПб.: Питер, 2002.
19. Дейт К. Дж. Введение в системы баз данных. — 7-е изд. — М., Вильямс, 2001.
20. Т. Коннолли, К. Бегг. Базы данных. Проектирование. Реализация. Сопровождение. — М., Вильямс, 2003.
21. А. Мазуркевич, Д. Еловой. PHP: настольная книга программиста. — М., Новое знание, 2004.
22. Д. Мелони, П. Вайнберг. PHP 4 в действии : пер. с англ. — М., Лучшие книги, 2002.
23. Дж. Кастаньетто, Х. Рафат, С. Шуман, и др. Профессиональное PHP программирование : пер. с англ. — СПб.: Символ-Плюс, 2001.
24. Петюшкин А.В. HTML. Экспресс-курс. — СПб. : БХБ-Петербург, 2003.
25. Харрис Э. PHP/MySQL: пер с англ. — М., КУДИЦ-ОБРАЗ, 2005.
26. А.В. Олишук, А. Н. Чаплыгин. Разработка WEB приложений на PHP 5. Профессиональная работа. — М., Вильямс, 2006.

Internet Web saytlari

<http://www.cgi.ru>
<http://www.woweb.ru>
<http://www.ru>
<http://www.vanta.ru/script/>
<http://www.vbnet.ru>
<http://www.scriptic.ru/>
<http://www.webacademy.com/>
[http:// www .intuit.ru/](http://www.intuit.ru/)

MUNDARIJA

KIRISH	3
I-qism. PHP TEXNOLOGIYASI	4
I bob. PHP VA MYSQL	4
1.1. PHP tarixi va uning imkoniyatlari	4
1.2. MySQL: tarix va imkoniyatlar	5
1.3. XAMPP dasturiy ta'minotini o'rnatish va sozlash	7
II bob. O'ZGARUVCHILAR VA MA'LUMOTLAR TIPI	16
2.1. PHP texnologiyasi haqida qisqacha ma'lumot	16
2.2. PHP deskriptorlarining ishlatilishi	16
2.3. PHP operatorlari	17
2.4. Izohlar.	17
2.5. O'zgaruvchi tiplari	18
2.6. Identifikatorlar	28
2.7. O'zgaruvchilar bilan ishlovchi funksiyalar	29
III bob. AMALLAR	32
3.1. Satrlar bilan ishlash amallari	32
3.2. Arifmetik amallar	34
3.3. Mantiqiy binar amallar	36
3.4. Tenglik amali	37
3.5. Munosabat (taqqoslash) amali	38
3.6. Ternar amali.	38
IV bob. PHP YORDAMIDA SO'ROVLARNI QAYTA ISHLASH	39
4.1. Mijoz-server texnologiyasi asoslari	39
4.2. HTTP protokoli va ma'lumotlarni serverga jo'natish metodlari	42
4.3. Mijoz so'rovi shakli.	43
4.4. Metodlar	44
4.5. HTML formasining serverga ma'lumotlarni jo'natish uchun qo'llanilishi	45
4.6. So'rovlarni PHP yordamida qayta ishlash	53
4.7. Formalar bilan ishlashga misollar	56
V bob. BOSHQARUVCHI KONSTRUKSIYALAR	63
5.1. Tanlash operatorlari.	63
5.2. Takrorlash (sikl) operatorlari	70
5.3. O'tish operatorlari.	73

5.4. Qamrab olish operatorlari	75
VI bob. FUNKSIYALAR.	84
6.1. Foydalanuvchi tomonidan aniqlangan funksiyalar.	84
6.2. Registr va funksiya nomi	86
6.3. Funksiyaga murojaat etish	87
6.4. Parametrlar (yoki Argumentlar)	88
6.5. Harakatlar maydoni.	94
6.6. Funksiya ichida o'zgaruvchilarning ishlatilishi.	95
6.7. Qaytariluvchi mazmunlar.	97
6.8. Qaytariluvchi jo'natmalar.	97
6.9. O'zgaruvchili funksiyalar	98
6.10. Standart funksiyalar.	99
VII bob. FAYLLAR BILAN ISHLASH	105
7.1. Ma'lumotlarni saqlash	105
7.2. Faylni ochish va faylga bog'lanish.	105
7.3. Fayl bilan bog'lanishni yopish	110
7.4. Fayl mavjudligini tekshirish	111
7.5. Faylga ma'lumot yozish	112
7.6. Fayldan ma'lumotlarni o'qish	117
7.7. Fayllar bilan ishlovchi funksiyalar	123
7.8. Faylni muhofazalash (blokirovkalash)	128
7.9. Serverga faylni yuklash	129
7.10. Qayta ishlashning ratsional usuli. Ma'lumotlar bazasini boshqarish tizimi.	133
VIII bob. MASSIVLAR	135
8.1. Massivlar	135
8.2. Massivlar ustida amallar	138
8.3. Massivlar bilan ishlovchi funksiyalar	139
8.4. Massivlarni saralash	146
8.5. Ko'p o'lchovli massivlarni saralash	150
8.6. Massivning har bir elementi uchun ixtiyoriy funksiyani qo'llash.	154
8.7. Qism massivni belgilash.	155
8.8. Massiv elementlari yig'indisini hisoblash.	156
8.9. Massiv elementlarining joylashgan tartibini o'zgartirish.	157
8.10. Ma'lumotlarni fayldan massivga yuklab olish	159
8.11. Massivlar ustida boshqa amallarni bajarish	162
IX bob. SATRLAR VA DOIMIY IFODALAR.	167
9.1. Satrlar bilan ishlash.	167

9.2. Satrlarni tahrirlash	169
9.3. Satrlarni taqqoslash	176
9.4. Satrlar bilan ishlovchi funksiyalar yordamida satrlarni ajratish va birlashtirish	178
9.5. Qism satrlarni qidirish va almashtirish	185
9.6. html-kodlardan tarkib topgan satrlar	191
9.7. Satrlar bilan ishlashga misollar	192

2-qism. MA'LUMOTLAR BAZASI BILAN ISHLASH 197

X bob. MA'LUMOTLAR BAZASI 197

10.1. Ma'lumotlar bazasi haqida asosiy tushunchalar	197
10.2. WEB ma'lumotlar bazasini tashkil etish	206
10.3. MySQL ma'lumotlar bazasini boshqarish tizimi	207
10.4. MySQL MBBT ning asosiy operatorlari	213

XI bob. PHP VA MYSQL NING O'ZARO BOG'LANISHI 233

11.1. PhpMyadmin yordamida MySQL ma'lumotlar bazasi bilan ishlash	233
11.2. MySQL tizimi bilan bog'lanishni o'rnatish va tizimda ishlash uchun ma'lumotlar bazasini tanlash	236
11.3. Tanlangan bazada so'rovlarni qayta ishlash	240

3-qism. FOYDALANISH UCHUN RUXSAT 258

XII bob. SESSIYALAR BILAN ISHLASH 258

12.1. Foydalanish uchun ruxsat	258
12.2. Sessiya mexanizmi	264
12.3. Sessiyani sozlash	265
12.4. Sessiyalar bilan ishlash	267
12.5. Xavfsizlik	272

ADABIYOTLAR RO'YXATI 275

Internet Web saytlari 276

Aripov Mersaid,
Dottoyev Sayfulla Xamidullayevich,
Fayziyeva Mahbubaxon Rahimjonovna

WEB TEKNOLOGIYALARI

o'quv qo'llanma

Muharrir *M. Akbarov*
Musahhih *H. Zokirova*
Dizayner sahifalovchi *F. Rahimov*

O'zbekiston faylasuflari milliy jamiyati nashriyoti.
100029, Toshkent shahri, Matbuotchilar ko'chasi, 32-uy.
Tel.: 236-55-79; Faks: 239-88-61.

Nashriyot litsenziyasi: AI №216, 03.08.2012.
Bosishga ruxsat etildi 12.10.2013. «Uz-Times» garniturası. Ofset usulida chop etildi. Qog'oz bichimi 60x84 $\frac{1}{16}$. Bosma tabog'i 17,5. Nashr hisob tabog'i 18,0. Adadi 500 nusxa. Buyurtma № 41.

«START-TRACK PRINT» MCHJ bosmaxonasida chop etildi.
Manzil: Toshkent shahri, 8-mart ko'chasi, 57-uy.