PHP MySQL

WEB DASTURLASH

fanidan (amaliy va laborotoriya) mashg'ulotlarini bajarish bo'yicha

USLUBIY-QO'LLANMA

JIZZAX-2020

Ushbu uslubiy qo'llanma «Web dasturlash» fani bo'yicha 5-semestr uchun mo'ljallangan bo'lib, unda amaliy-laboratoriya mashg'ulotlarini bajarish bo'yicha nazariy ma'lumotlar, uslubiy ko'rsatmalar, variant topshiriqlari berilgan. Har bir amaliy-laboratoriya hisobot shaklida rasmiylashtirilishi keltirilgan. Undagi mavzular o'quv reja va fan dasturi asosida tuzilgan.

Tuzuvchilar:

Baratov J.R. JizPI, " Informatsion texnologiyalar "kafedrasi assistenti.

Taqrizchilar:

XaitovF.N. A.Qodiriy nomli Jizzax davlat pedagogika instituti "Informatika o'qitish metodikasi" kafedrasi mudiri, t.f.n., dotsent.

Yusupov R.M. A.Qodiriy nomli Jizzax davlat pedagogika instituti "Informatika o'qitish metodikasi" kafedrasi dotsent, t.f.n.

Ushbu uslubiy qo'llanma "Informatsion texnologiyalar " kafedrasining 20__ 2020 yil 14 apreldagi № — -sonli yig'ilishida muhokamadan o'tgan va fakultet kengashida muhokama qilish uchun tavsiya etildi.

Kafedra mudiri: ________________ **dos. U.O'.Turopov**

Ushbu uslubiy qo'llanma "Elektromexanika va radioelektronika " fakulteti kengashida 2020 yil 19-maydagi № 8-sonli yig'ilishida muhokamadan o'tgan va institut ilmiy-uslubiy kengashida muhokama qilish uchun tavsiya etildi.

Fakultet kengashi raisi: ____________ **pfd. U.M.Yarlaqabov**

Ushbu uslubiy qo'llanma institut ilmiy-uslubiy kengashining 2020– yil 31-maydagi bo'lib o'tgan 10-sonli majlisida ko'rib chiqildi va chop etishga tavsiya etildi.

© «Web dasturlash» fanidan amaliy-tajriba mashg'ulotlari uchun uslubiy qo'llanma. Jizzax Politexnika instituti®,–Jizzax : JizPI, 2020 yil. 220-bet.

MUNDARIJA

Fanning na'munaviy dasturi 4

Fanning ishchi o'quv dasturi. 6

Ma'ruzalar matni. 27

Amaliy mashg'ulot materiallari 89

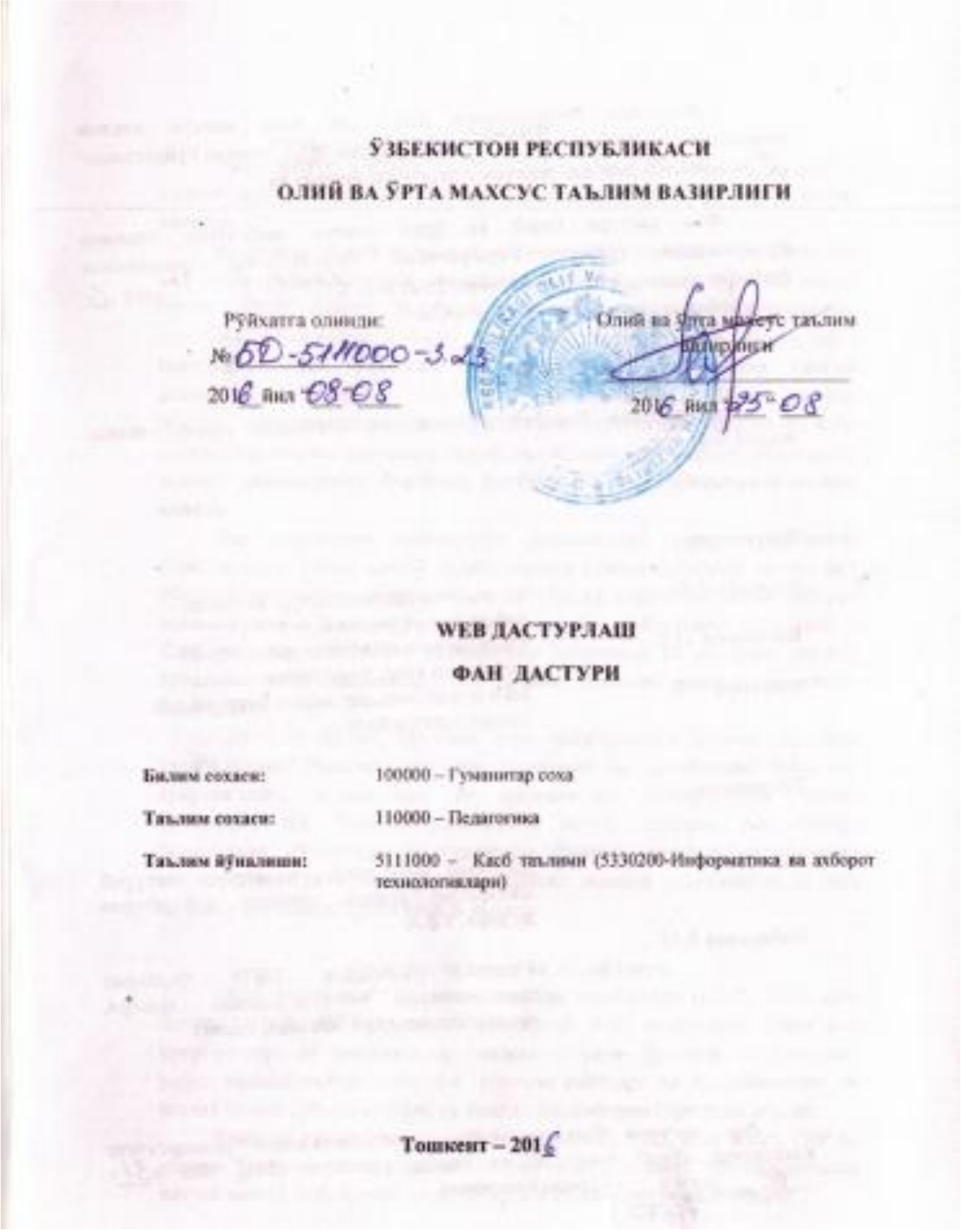
Laboratoriya mashg'ulot materiallari 128

Glossariy. 218

Xulosa. 219

Adabiyotlar ro'yxati 220

1.1. Namunaviy fan dasturi



ЎЗБЕКИСТОН РЕСПУБЛИКАСИ

ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ

Рўйхатга олинди:
№ БД-5111000-3.23
2016 йил 08-08

Олий ва ўрта махсус таълим вазирлиги
2016 йил 25-08

WEB ДАСТУРЛАШ

ФАН ДАСТУРИ

Билим соҳаси:	100000 – Гуманитар соҳа
Таълим соҳаси:	110000 – Педагогика
Таълим йўналиши:	5111000 – Касб таълими (5330200-Информатика ва ахборот технологиялари)

Тошкент – 2016

Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлигининг 2016 йил "25" 08 даги "355"-сонли буйругининг 2 -иловаси билан фан дастури рўйхати тасдикланган.

Фан дастури Олий ва ўрта махсус, касб-хунар таълими йўналишлари бўйича Ўкув-услубий бирлашмалар фаолиятини Мувофиклаштирувчи Кенгашининг 2016 йил "08" 08 даги 3 - сонли баённомаси билан маъкулланган.

Фан дастури Тошкент ахборот технологиялари университетида ишлаб чикилди.

Тузувчилар:

Керимов К.Ф. - Тизимли ва амалий дастурлаштириш кафедраси мудири т.ф.н.

Бабажанов М.Р. - Тизимли ва амалий дастурлаштириш кафедраси катта ўкитувчиси

Файзиева М.Р. - Бош илмий методик марказ катта илмий ходим изланувчиси

Такризчилар:

Равшанов Н - Дастурий махсулотлар ва аппарат-дастурий мажмуалар яратиш маркази лаборатория мудири, т.ф.д.

Набиулина Л.М. - Низомий номидаги ТДПУ "Ахборот технологиялари" кафедраси мудири, педагогика фанлари номзоди, доцент.

Фан дастури Тошкент ахборот технологиялари университети Кенгашида кўриб чикилган ва тавсия килинган (2016 йил «31» 05 даги «__» - сонли баённома). 7(657)

1.2. Fanning ishchi-o’quv dasturi

O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA’LIM VAZIRLIGI

JIZZAX POLITEXNIKA INSTITUTI

O‘quv-uslubiy bo‘lim tomonidan ro‘yxatga olindi

№ ____________________

2017 y. «____» ____________

“TASDIQLAYMAN”
O‘quv ishlari bo‘yicha prorektor

____

2017y. «____» ____________

“Informatsion texnologiyalar”kafedrasi

“Web dasturlash ” fanining

ISHCHI O‘QUV DASTURI

OLIY TA’LIMNING

Bilim sohasi:	100000-Gumanitar soha
Ta’lim sohasi:	110000-Pedagogika
Ta’lim yo‘nalishi:	5111000 - Kasb ta’limi (5330200-Informatika va axborot texnologiyalari)

JIZZAX – 2017

1

Mazkur ishchi dastur Jizzax politexnika instituti “Informatsion texnologiyalari” kafedrasida 5330200-Informatika va Axborot texnologiyalari bakalavriat ta’lim yo‘nalishlarining o‘quv rejasidagi «Web dasturlash» fani bo‘yicha O‘zR Oliy va o‘rta maxsus ta’lim Vazirligi tomonidan tasdiqlangan Namunaviy o‘quv dasturi asosida ishlab chiqilgan.

Tuzuvchilar:

Baratov J.R. JizPI, “Informatsion texnologiyalar” kafedrasi assistenti

Taqrizchilar:

Xaitov F.N. A.Qodiriy nomli Jizzax davlat pedagogika instituti “Informatika va AT” kafedrasi mudiri, t.f.n., dotsent

Fanning ishchi o‘quv dasturi “Informatsion texnologiyalar” kafedrasining 201_ yil ___ _________dagi № __-sonli yig’ilishida muhokamadan o’tgan va fakultet kengashida muhokama qilish uchun tavsiya etildi.

Kafedra mudiri: _______________ dots. S.A.Tavboyev

Ushbu uslubiy qo’llanma “Elektromexanika va radioelektronika ” fakulteti kengashida 201_ yil ___ _________dagi № __-sonli yig’ilishida muhokamadan o’tgan va institut ilmiy-uslubiy kengashida muhokama qilish uchun tavsiya etildi.

Fakultet kengashi raisi: __________ To’raqulov O.X.

© “web dasturlash ” fani (Ishchi o‘quv dasturi).-Jizzax: JizPI, 2017. 20–bet.

Kirish

Ushbu dastur Uzbekiston Respublikasining "Ta'lim to'g'risida"gi qonuni hamda "Kadrlar tayyorlash Milliy dasturi"ga muvofik ishlab chiqildi.

Zamonaviy ilmiy-texnika revolyusiyasining o'ziga xos xususiyati har xil ko'rinishdagi axborot oqimlari xajmining jadal suratlar bilan ortib borishi. axborotlarga bulgan talab kurinishlarining shakllanishi, xar xil ko'rinishli axborot kadrining ortishi kabi faktorlarga bog'liqdir. Real borlikni vizuallashtirish uch o'lchovli grafik axborotlarni paydo bulishiga va ularni qayta ishlash texnologiyalarining takomillashuviga asos bulib xizmat kiladi. Bu kabi axborotlar jamiyat rivojiga, ilm-fanning yangi kirralarini ochilishiga, jamiyat a'zolarining borlikni butunligicha xis qilishlariga xizmat kiladi.

Har tomonlama shakllangan, ma'naviyatli shaxsni tarbiyalab yetishtirishda, uning ilmiy dunyokarashini shakllantirishda muxim rol uynovchi kompyuter fafikasi va dizayn va informatika va axborot texnologiyalari soxalar uchun pedagog kadrlar tayyorlovchi «5111000 -Kasb ta'limi (5330400 - Kompyuter grafikasi va dizayn)» ta'lim yunalishi uchun zamonaviy web dizayn fanining urni va axamiyati bekiyosdir.

Har bir fanni o'rganish uni maksadiariga boglik. Bu fan talabalarning Internet asoslari, zamonaviy veb dizaynning uziga xos xususiyatlari, muammolari va rivojlanish istikbollari xakida tasavvurga ega bulishi, dizaynning paydo bulishi, rivojlanish boskichlari, obyektlari va turlarini bilishi, dizaynerlar ijodini taxdil etishi va yetakchi dizaynerlarning ijodiy uslublarini kulday olishi uchun vosita bulib xizmat kiladi.

O'quv fanining maqsadi va vazifalari

"Web dasturlash" fanining maksadi talabalarga HTML, CSS, Java Script, MySQL, PHP yordamida zamonaviy Web dizaynning uziga xos xususiyatlari va rivojlanish tendensiyalarini urgagish va Internet resurslarini(reklama saytlari, tijorat saytlari va x.) loyixalash va ishlab chiqish buyicha nazariy va amaliy bilimlarni berishdan iborat.

Fanning vazifasi esa talabalarda HTML, CSS, Java Script, PHP va ularning zamonaviy asoslarini urgatish va ular asosida Web resurslarni ishlab chiqish, loyixalash kunikmalarini shakllantirishdan iborat.

Fan bo'yicha talabalar bilim, ko'nikma va malakalariga qo'yiladigan talablar

Talabalar mazkur fanni uzdashtirish natijasida:

HTML, CSS, Java Script, JQuery, MySQL. PHP imkoniyatlari *haqida tasavvurga ega bo'lishlari;*

HTML, CSS, Java Script, JQuery, MySQL, PHP larning zamonaviy asoslarini ***bilish*** hamda Internet saytlarini shakllantirishda Web dizaynning zamonaviy usul va metodlardan samarali ***foydalana olishlari;***

Web loyixalar strukturasini shakllantirish, Web loyixalarni yaratishda moye dasturiy-texnik vositalarni kullay olish, internetda uzining Web loyixa variantini takdim eta olish ***ko'nikmasiga ega bo'lishlari lozim.***

Fanning o'quv rejadagi boshqa fanlar bilan o'zaro bog'liqligi va uslubiy jihatdan uzviy ketma - ketligi

"Web dasturlash" fanini uzlashtirish kuyidagi fanlardan yetarli bilim va kunikmalarni talab kiladi: "Dasturlash texnologiyasi», «Kompyuter grafikasi va dizaynning maxsus effektlari», «Kompyuter tarmoklari», «Animasion texnologiyalar», «Ma'lumotlar banki va bazalari». Bundan tashkari ukuvchi talabalar biror bir dasturlash tilida dastur yaratish tajribasiga ega deb karaladi.

Mazkur fan buyicha olingan bilim va kunikmalar "Web dasturlash", "Kompyuter fafikasi va Web dizayn" fanlarini o'rganishda xamda kurs va bitiruv ishlarini tayyorlashda katta yordam beradi.

Fanning ishlab chiqarishdagi o'rni

"Web dizayndasturlash" fani yordamida yangi zamonaviy Web loyixalarni ishlab chiqish, tadbik etish buyicha olgan bilim va kunikmalari natijasida axborot texnologiyalaridan foydalangan xolda ishlab chikarishni rivojlantirish uchun lozim buladigan dasturiy majmua va ta'minotlarni samarali, rasional va sifatli loyixalash va ishlab chiqishda ishtirok etishi mumkin.

Fanni o'qitishda zamonaviy axborot va pedagogik texnologiyalar

Talabalarning "Web dasturlash" fanini uzlashtirishlari *uchun,* ukitishning ilgor va zamonaviy usullaridan *xamda* yangi pedagogik, informasion texnologiyalardan foydalanish muxim axamiyatga *ega.* Ukuv fanini talab darajasida uzlashtirish uchun uni ukitish jarayonida ukitishning interfaol usullaridan; kichik guruxlaria ishlash, akliy xujum, muammoli ukitish xamda jumbok usullar kabn innovasion metodlardan foydalanib talabalarning erkin va mustakil ishlashini samarali tashkil etish maksadida ularga kuyilgan ukuv vazifalarini mustakil bajarishni tashkil etish maksadida axborot texnolog iyalari xamda axborot ta'lim muxitidan foydalanish kuzda tutiladi.

Fanni uzlashtnrishda darelik, internet va intranst tarmoklari, ukuv va uslubiy qo'llanmalar, ma'ruza matnlari, videoma'ruzalar, tarqatma materiallar, elektron ma'ruzalar, elektron qo'llanmalardan foydalaniladi.

ASOSIY QISM

2.1 MA’RUZA MASHG’ULOTLAR

2.1.1. *"Web dasturlash" faniga kirish*

Web dizaynning asosiy tushunchalari. Web texnologiya: HTML; Web brauzerlar; Web-saxifa, Web-sayt, Web-server; razmetkali va ssenariyli dasturlash tillari; "kliyent-server" texnologiyasi, server struktuasi. Web xosting. Web dizaynning paydo bo’lishi va rivojlanish bosqichlari. O’zbekistonda dizaynning *rivojlani1i* tendensiyalari. Web dizaynning asosiy koidalari. Web dizayn texnologiyalari: saytning mantikiy va fizik sgr)kt)rasi, saytning statik va dinamik komponovkasi, Web-saxifa elementlari.

Qo’llaniladigan ta’lim texnologiyalari:*dialogik yondashuv, muammoli ta’lim, bingo, blis-so’rov, munozara, o’z-o’zini nazorat.*

Adabiyotlar:A1; A2; A3; A5; A6;A9; Q1; Q2; Q3.

2.1.2. *HTML asoslari*

Xujjat tuzilishi. HTML ning asosiy teglari. Matnlarni bezash. Ruyxatlar. Jadvallar. Freymlar. Grafika va multimedia. Gipermurojaat teglari. Formalar. HTML ning yangi imkoniyatlari va eskirgan teglar.

Qo’llaniladigan ta’lim texnologiyalari:*dialogik yondashuv, muammoli ta’lim, bingo, blis-so’rov, munozara, o’z-o’zini nazorat.*

Adabiyotlar:A1; A2; A3; Q5; Q10 Q2; Q3.

2.1.3. *CSS asoslari*

CSS (Cascading Style Sheets) asoslari. Selektor, klass, identifikatorlar va ularning CSS da ishlash prinsiplari. Selektor guruxlari. PSD selektorlari va ulardan foydalanish. Ssылka selektorlari. CSS da fon, rang va shriftlar bilan ishlash. CSS da matn va ruyxatlar bilan ishlash. Blok elementlari umumiy konsepsiyalari. Sloylar bilan ishlash. Cross brauzerlar.

Qo’llaniladigan ta’lim texnologiyalari:*dialogik yondashuv, muammoli ta’lim, bingo, blis-so’rov, munozara, o’z-o’zini nazorat.*

Adabiyotlar:A1; A2; A3; A5; A6;A9; Q1; Q5; Q10 Q2; Q3.

2.1.4. *JavaScriptna dasturlash*

JavaScriptning asosiy tushunchalari: xossalar, metodlar, xodisalar. JavaScript da malumot tiplari, O’zgaruvchilar, obyektlar, ifodalar, arifmetik amallar. JavaScript da jarayonlarni boshqarish elementlari(shartli va sikl operatorlari). JavaScript da OID asoslari(standart, brauzer va dasturchi tomonidan

yaratilgan obyektlar). JavaScript da xodisalar. Interaktiv formalar ishlab . chiqish. Multimediali veb ilova ishlab chiqish (kliyent uchun). Uyin, karkulyator, animasiyali ilovalar ishlab chiqish. Web saxifani chop qilish. Interneta ishlaydigan ilovalar ishlab chiqish (sosial saytlar, pochta, izlash tizimlari)

Qo'llaniladigan ta'lim texnologiyalari:*dialogik yondashuv, muammoli ta'lim, bingo, blis-so'rov, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A5; A6;A8; Q1; Q2; Q3.

2.1. PHP asoslari

Web serverda dasturlash. Web server dasturiy ta'minoti(XAMRR, LAMP, WAMP, **DENWER)** o'rnatish va sozlash. PHP asosiy tushunchalari. konstanta, o'zgaruyachilar. ma'lumot tiplari. PHP da amallar bilan ishlash. PHP da jarayonlarni boshqaruvchi elementlar(shartli va sikl operatorlar). PHP da funksiyalar. PHP da massiv va satrlar va fayllar. MySQL ma'lumotlar bazasida ishlash. Ma'lumotlar bazasini tashkil qilish va undan foydalanish. PHP va MySQL integrasiyasi. Sessiyalar bilan ishlash. Veb resurs xavfsizligi. Mobil vositalar uchun saytlar qilish imkoniyatlari. An'anaviy mobil saytlar taraqqiyoti va farqlari.

Qo'llaniladigan ta'lim texnologiyalari:*dialogik yondashuv, muammoli ta'lim, bingo, blis-so'rov, munozara, o'z-o'zini nazorat.*

Adabiyotlar:A1; A2; A3; A5; A6;A7; Q1; Q4; Q2; Q3.

"Web dasturlash" fani bo'yicha ma'ruza mashg'ulotlarining kalendar tematik rejasi

Ma'ruza mavzulari

(5 – semestr uchun – 10 soat)

№	*Mashg'ulot turi*	Mavzu nomi va nazoratlar turlari	*Ajratilgan soatlar*	*Bajarilganligi haqida ma'lumot*	*O'qituvchi imzosi*
				Oy va kun	
1.	Ma'ruza	PHP. Server tomondan dasturlash. PHP ga kirish. PHP ni o'rnatish va testlash.	2		
2.	Ma'ruza	PHP asosiy tuzilishi. Ma'lumotlar tiplari. Ifodalar. Jarayonlarni boshqarish	2		
3.	Ma'ruza	PHP da satr va massivlar bilan ishlash	2		
4.	Ma'ruza	Funksiyalar. Klasslar va obyektlar. Xatoliklar bilan ishlash. PHP-funksiyalar	2		
5.	Ma'ruza	PHPda ma'lumotlar bazalari bilan ishlash. MySQL ma'lumotlar bazasi	2		
	Jami		10s		

“Web dasturlash” fani bo’yicha amaliy mashg’ulotlarning kalendar tematik rejasi

Amaliy mashg‘ulot soatlari
(5 – semestr uchun - 14 soat)

№	*Mashg’ ulot turi*	Mavzu nomi va nazoratlar turlari	*Ajratil gan soatla r*	*Bajarilganli gi haqida ma’lumot*	*O’qituvch i imzosi*
				Oy va kun	
1	Amaliy	**PHP sintaksisi**	2		
2	Amaliy	**Matematik funksiyalar**	2		
3	Amaliy	**O'zgarmaslar**	2		
4	Amaliy	**SQL so'rovlar tili**	2		
5	Amaliy	**MySQL da ma'lumot turlari**	2		
6	Amaliy	**MySQL ga ma'lumot qo'shishni rejalashtirish**	2		
7	Amaliy	**MySQL ga bir nechta ma'lumot qo'shish**	2		
	Jami		14s		

“Web dasturlash” fani bo’yicha tajriba mashg’ulotlarning kalendar tematik rejasi

Tajriba mashg‘ulot soatlari

(5 – semestr uchun - 52 soat)

№	*Mashg’ulot turi*	**Mavzu nomi va nazoratlar turlari**	*Ajratilgan soatlar*	*Bajarilganligi haqida ma’lumot* (Oy va kun)	*O’qituvchi imzosi*
1	Tajriba	Shart operatori	2		
2	Tajriba	S_GET o'zgaruvchilari PHP da toifalar	2		
3	Tajriba	PHP da satrlar bilan ishlash	2		
4	Tajriba	Tanlash operatori	2		
5	Tajriba	for sikl operatori	2		
6	Tajriba	do-while, while sikl operatorlari	2		
7	Tajriba	Funksiyalar bilan ishlash	2		
8	Tajriba	PHP da massivlar bilan ishlash	2		
9	Tajriba	PHPda ko'p o'lchamli massivlar	2		
10	Tajriba	PHP da massivlar bilan ishlovchi funksiyalar	2		
11	Tajriba	PHP da superglobal o'zgaruvchilar	2		
12	Tajriba	GET va POST metodlari	2		
13	Tajriba	PHP da forma elementlarini tekshirish	2		
14	Tajriba	Majburiy maydonlarni to'ldirish	2		
15	Tajriba	Email va URLni tekshirish	2		
16	Tajriba	Forma elementari qiymati	2		
17	Tajriba	PHPda fayllarni yuklash	2		
18	Tajriba	PHP da include va require funksiyalari	2		
19	Tajriba	MySQL ga ulanish	2		
20	Tajriba	MySQL da ma'lumotlar bazasini hosil qilish	2		
21	Tajriba	MySQL da jadval hosil qilish	2		
22	Tajriba	MySQL dan ma'lumotlarni o'qish	2		
23	Tajriba	MySQL ga ma'lumot qo'shish	2		
24	Tajriba	MySQL ma' lumotlarni o’chirish	2		
25	Tajriba	PHPda sessiyalar bilan ishlash	2		
26	Tajriba	MySQL ma’lumotlarni yangilash (tahrirlash)	2		

	Jami		52s		

Laboratoriya ishlarini tashkil etish bo'yicha ko'rsatma va tavsiyalar

Laboratoriya ishlarini tashkil qilish, olib borish va bajarish buyicha kafedra professor-ukituvchilari tomonidan tashkil i y-uslubiy kursatmalar ishlab chiqilishi, xar bir laboratoriya tajriba ishini bajarishdan oldin, tayyorlanishi lozim bulgan laboratoriya ishiga oid mavzular buyicha maslaxatlar utkazilishi, talabalar laboratoriya kurilmalarida ishlash uchun texnika xavfsizligini o'rganishlari, talaba laboratoriya ishiga tayyorgarlik kurish davrida mazkur kursatma va tavsiya etilayotgan adabiyotlardan foydalangan xolda kerakli nazariy materiallarni o'rganishlari, zaruriy xisoblashlarni amalga oshirishlari va nazorat savollariga javob berishlari lozim buladi. Bundan tashkari xar bir laboratoriya ishi talaba tomonidan mustakil ravishda tayyorlanishi va shaxsiy tarzda xisobot topshirilishi xam lozim buladi. Xisobotni elektron xujjat kurinishda topshirishga ruxsat etilib, talabalar bilimlari laboratoriya mashguloti va xisobot topshirish mobaynida ukituvchi tomonidan tekshiriladi.

Laboratoriya ishlariga tavsiya qiladigan taxminiy mavzular ro'yxati

Internet ilovalar ishlab chiqish uchun zarur bo'ladigan dasturiy ta'minotlar. va vositalar (Operapion sistemalar. veb serverlar, matn muxarirlar. CMS lar)

Internet asoslari: protokollar HTTP, URL, HTML va boshqa standartlar.

Razmetkali tillar: HTML. XML. XHTML. HTML-5 yangiliklar. Eskirgan teglar.

Web loyihalar strukturasini ishlab chiqish.

HTML da matnlar bilan ishlovchi teglar va ularga doir masalalar yechish.

HTML da gipermurojaat teglari bilan ishlash. Ularga oid masalalar yechish.

HTML da multimedia va obyekt teglari bilan ishlash. Ularga oid masalalar yechish.

HTML da jadval va freym teglari bilan ishlash. Ularga oid masalalar yechish.

HTML da turli formalarni yaratish. Ularga oid masalalar yechish.

HTML da rasmlarni vizual bir nechta kismlarga bulish teglari vaularga oid masalalar yechish.

HTML tili yordamida Veb saxifa yaratish.

Selektor, klass, identifikatorlar va CSS da ishlash prinsipi.

CSS da fon, rang va shriftlar bilan ishlash. Ularga oid masalalaryechish.

CSS da matn va ruyxatlar bilan ishlash. Ularga oid masalalar yechish.

Blok maketiga asoslangan sodda modellar kurish asoslari.

Sloylar bilan ishlash. Ularga oid masalalar yechish.

Dinamik kontentlar hosil qilish. Ularga oid masalalar yechish.

Blok maketi asosida CSS va HTML dan foydalanib internetmagazinlarni yaratish asoslari.

HTML va CSS yordamida sodda saytlar yaratish.

Tablisa maketidan foydalanish. Konteyner maketidan foydalanish.

Query da ishlash asoslari.

JavaScript da obyekt va metodlar bilan ishlash. Ularga oid masalalar yechish

JavaScript da forma elementlari bilan ishlash. Ularga oid masalalar yechish.

Mijoz ilovalar obyektlari. Xodisalarni qayta ishlash. Ularga oid masalalar yechish.

HTML, CSS, JavaScript, DOM lardan foydalanib dasturlar ishlab chiqish.

JavaScript va CSS ni o'zaro bog'lash. Ularga oid masalalar yechish. Malumotlarni kiritish formalarini gekshiruvchi dastur ishlab chiqish.

Multimediali kontentli veb saxifa ishlab chiqish.

Xarakatlanuvchi elementlar va sloylarni hosil qilish.

Web server dasturiy ta'minoti (XAMRR. LAMP, WAMP. DENWER)ni o'rnatish va sozlash.

PHP da forma ma'lumotlarini qayta ishlash. Ularga oid masalalar yechish.

Formalarga GET va POST metodlaridan foydalanib ma'lumot kirituvchi saxifalar ishlab chiqish.

PHP da boshqaruvchi konstruksiyalar bilan ishlash.

PHP da funksiyalar bilan ishlash. Ularga oid masalalar yechish.

PHP da massivlar bilan ishlash. Ularga oid masalalar yechish.

PHP da satrlar bilan ishlash. Ularga oid masalalar yechish.

PHP da internet magazinlar uchun buyurtma formasini tuzish

PHP da fayllar bilan ishlash. Malumotlarni faylga yozish va fayldan ma'lumotlarni o'qish

Serverga faylni yuklash.

PHP da klass va obyektlar yordamida veb saxifa yaratish.

PHP dinamik dasturlash kontenti xususiyatlari.

MySQL ma'lumotlar bazasi bilan ishlash va loyixalashtirish.

PhpMyadmin yordamida MySQL ma'lumotlar bazasi bilan ishlash.

PHP va MySQL ma'lumotlar bazasi integrasiyasi. Ularga oid masalalar yechish.

PHP sahifasi orqali ma'lumotlar bazasiga so'rovlar tashkil etish.

PHP da sessiyalar bilan ishlash

PHP da grafika bilan ishlash.

PHP va JQuery bilan ishlash . (Forms, Pop-ups).

Sodda sayt yaratish asoslari.

- Internet magazin uchun ma’lumotlar bazasini yaragish.
- Internet magazin saxifa va ma’lumotlar bazasini bog’lash
- PHP va MySQL asosida sayt yaratish.
- Saytni PHP da himoya qilish.

Kurs ishini tashkil etish buyicha uslubiy ko’rsatmalar

Kurs ishi mavzuning dolzarbligi va erishilgan natijalarning amaliyotga tadbiki, uning uslubiy darajasi xamda rasmiylashtirilishiga qarab baxolanadi. Kurs ishini bajarishning muxim boskichi rejada belgilangan savollar yoritilishida u^zaR° nazariy va amaliy alokadorlikni ta’minlashdir. Kurs ishining tarkibiy tuzilishini tugri shakllantirish talabaga uning maksadi va vazifalarini anik belgilab olish xamda kuzlangan natijaga erishish yullarini, shuningdek, kurs ishini tayyorlash boskichlarini ketma-ket bajarishga imkon yaratadi. Kurs ishining tarkibiy tuzilishini tugri ishlab chiqilishi talabaga izlanishning oldiga kuyilgan maksad xamda natijalarini anik ifodalash, fikrlarni mantikiy bayon etish uslublari va kurs ishini tayyorlash boskichlarini ajratish imkonini beradi.

Kurs ishi talabaning saloxiyati xamda fan buyicha chukur bilim va amaliy kunikmaga egaligini namoyon kiladi. Kurs ishida urganiladigan masalalarning nazariy jixatlari, aniklangan muammolarini, kuyilgan vazifalarni taxlil qilish xamda muammolarni xal etishni kuzlagan xolda reja tuziladi va shu asosida bajariladi. Talaba kurs ishi buyicha izlanish natijalarini yoritishda Web dasturlarva ilovalarni ishlab chiqishni tashkil qilishning uziga xosligi, xorijiy davlatlarning tajribalari va orttirgan amaliy yutuklarni umumlashtirgan xolda belgilangan muammolarning yechimi yuzasidan yul-y$fiklarni ishlab chikadi va ularni bartaraf etish yullarini tavsiya etadi.

Kurs ishlarining namunaviy mavzulari

- Internet ilovalar ishlab chiqish uchun zarur buladigan dasturiy ta’minotlar va vositalar.
- Word Press imkoniyatlari .
- PHP imkoniyatlari.
- JavaScript imkoniyatlarini taxlil qilish.
- Vizual)ffsktlar.
- Vizual formatlash modeli.
- S55da adreslash modeli.
- CSS2 da konteynerlar.
- PHP da JQuery bilan ishlash imkoniyatlari
- Bootstrap imkoniyalarini taxlil qilish.

Mustaqil ta’limning shakli va mazmuni

Talaba mustakil ishni tayyorlashda fanning xususiyatlarini xisobga olgan xolda kuyidagi shakllardan foydalanish tavsiya etiladi:

- darslik va ukuv kullanmalar buyicha fan boblari va mavzularini o'rganish;
- tarkatma materiallar buyicha ma'ruzalar kismini uzlashtirish;
- avtomatlashtirilgan, urgatuvchi va nazorat kiluvchi dasturlar bilan ishlash;
- maxsus adabiyetlar buyicha fan bulimlari Yoki mavzulari ustida ishlash;
- lokal va global kompyuter tarmogidagi ma'lumotlarni taxlil qilish;
- masofali ta'lim texnologiyalardan foydalanib fan mavzularini O'rganish.

Mustakil ishning asosiy shakllari bulib referat yozish, konspekt tayyorlash, nazorat ishi, ma'ruza matnini yozish, internetdan foydalanib, fanga oid bulgan yangi ma'lumotlarni olish *va* boshkalar kiradi.

Tavsiya etilayotgan mustaqil ishlarning mavzulari

- HTML tili imkoniyatlari va yangi teglarning vazifalari.
- CSS texnologiyasila ishlash.
- PHP va JQuery imkoniyatlari.
- PHP da dasturlashda freymvorklarning urni
- Shablonlar.
- CSS konteynerlar.
- JavaScript da dasturlash.
- Bootstrap imkoniyalarini taxlil qilish.
- CSS da konformlilik.
- PHP da JQuery bilan ishlash.
- Klass selektorlari.

Dasturning informatsion-uslubiy ta'minoti

Mazkur fanni ukitish jarayonida ta'limning zamonaviy metodlari, pedagogik va axborot-kommunikasiya texnologiyalarni kullash nazarda tutilgan:ma'ruza darslarida zamonaviy kompyuter texnologiyalari yordamida prezentasion va elektron-didaktik materiallardan;amaliy va laboratoriya mashgulotlarda zamonaviy kompyuter sinflaridan foydalanish kuzda tutilgan.

Shuningdek bugungi kunning asosiy ma'lumotlar olish bazasi sifatida Intranet va Internet tizimiga ulangan bulishi va universitet portaliga ma'lumotlarni joylashtirish imkoniyati mavjud bulishi lozim.

"Web dasturlash" fanidan talabalar bilimini reyting tizimi asosida baholash mezoni

"Web dasturlash" fani bo'yicha reyting jadvallari, nazorat turi, shakli, soni hamda har bir nazoratga ajratilgan maksinial ball, shuningdek, joriy va oraliq nazoratlarining saralash ballari haqidagi ma'lumotlar fan bo'yicha birinchi mashg'ulotda talabalarga e'lon qilinadi.

Fan bo'yicha talabalarning bilim saviyasi va o'zlashtirish darajasining Davlat ta'lim standartlariga muvofiqligini ta'minlash uchun quyidagi nazorat turlari o'tkaziladi:

- **joriy nazorat (JN)** - talabaning fan mavzulari bo'yicha bilim va amaliy ko'nikma darajasini aniqlash va baholash usuli. Joriy nazorat fanning xususiyatidan kelib chiqqan holda ma'ruza va amaliy mashg'ulotlarda og'zaki so'rov, test o'tkazish, suhbat, nazorat ishi, kollekvium, uy vazifalarini tekshirish va shu kabi boshqa shakllarda o'tkazilishi mumkin;
- **oraliq nazorat (ON)** - semestr davomida o'quv dasturining tegishli (fanlarning bir necha mavzularini o'z ichiga olgan) bo'limi tugallangandan keyin talabaning nazariy bilim va amaliy ko'nikma darajasini aniqlash va baholash usuli. Oraliq nazorat bir semestrda ikki marta o'tkaziladi va shakli (yozma, og'zaki, test va hokazo) o'quv faniga ajratilgan umumiy soatlar hajmidan kelib chiqqan holda belgilanadi;
- **yakuniy nazorat (YaN)** - semestr yakunida muayyan fan bo'yicha nazariy bilim va amaliy ko'nikmalarni talabalar tomonidan o'zlashtirish darajasini baholash usuli. Yakuniy nazorat asosan tayanch tushuncha va iboralarga asoslangan "Yozma ish" shaklida o'tkaziladi.

ON o'tkazish jarayoni kafedra mudiri tomonidan tuzilgan komissiya ishtirokida muntazam ravishda o'rganib boriladi va uni o'tkazish tartiblari buzilgan hollarda **ON** natijalari bekor qilinishi mumkin. Bunday hollarda **ON** qayta o'tkaziladi.

Oliy ta'lim muassasasi rahbarining buyrug'i bilan monitoring va ichki nazorat bo'limi rahbarligida tuzilgan komissiya ishtirokida **YaN** ni o'tkazish jarayoni muntazam ravishda o'rganib boriladi va uni o'tkazish tartiblari buzilgan hollarda **YaN** natijalari bekor qilinishi mumkin. Bunday hollarda **YaN** qayta o'tkaziladi.Talabaning bilim saviyasi, ko'nikma va malakalarini nazorat qilishning reyting tizimi asosida talabaning fan bo'yicha o'zlashtirish darajasi ballar orqali ifodalanadi.

Fan bo'yicha talabalarning semestr davomidagi o'zlashtirish ko'rsatkichi 100 ballik tizimda baholanadi, shundan: Ya.N.- 30 ball, J.N.- 35 ball va O.N.-35 ball qilib taqsinilanadi.

Ball	Baho	Talabalarning bilim darajasi
86-100	A'lo	Xulosa va qaror qabul qilish. Ijodiy fikrlay olish. Mustaqil mushohada yurita olish. Olgan bilimlarini amalda qo'llay olish. Mohiyatini tushuntirish. Bilish, aytib berish. Tasavvurga ega bo'lish
71-85	Yaxshi	Mustaqil mushohada qilish. Olgan bilimlarini amalda

		qo'llay olish. Mohiyatini tushuntirish. Bilish, aytib berish. Tasavvurga ega bo'lish
55-70	Qoniqarli	Mohiyatini tushuntirish. Bilish, aytib berish Tasavvurga ega bo'lish
0-54	Qoniqarsiz	Aniq tasavvurga ega bo'lmaslik. Bilmaslik

• Fan bo'yicha saralash bali 55 ballni tashkil etadi. Talabaning saralash balldan past bo'lgan o'zlashtirishi reyting daftarchasida qayd etilmaydi.

• Talabalarning o'quv fani bo'yicha mustaqil ishi joriy, oraliq va yakuniy nazoratlar jarayonida tegishli topshiriqlarni bajarishi va unga ajratilgan ballardan kelib chiqqan holda baholanadi.

• Talabaning fan bo'yicha reytingi quyidagicha aniqlanadi:

R=*V*O'/100,*

bu yerda: V - semestrda fanga ajratilgan umumiy o'quv yuklamasi(soatlarda); *O'* -fan bo'yicha o'zlashtirish darajasi (ballarda).

Fan bo'yicha joriy va oraliq nazoratlarga ajratilgan umumiy ballning 55 foizi saralash ball hisoblanib, ushbu foizdan kam ball to'plagan talaba yakuniy nazoratga kiritilmaydi.

• Joriy **JN** va oraliq **ON** turlari bo'yicha 55ball va undan yuqori balni to'plagan talaba fanni o'zlashtirgan deb hisoblanadi va ushbu fan bo'yicha yakuniy nazoratga kirmasligiga yo'l qo'yiladi.

• Talabaning semestr davomida fan bo'yicha to'plagan umumiy bali har bir nazorat turidan belgilangan qoidalarga muvofiq to'plagan ballari yig'indisiga teng.

• **ON** va **YaN** turlari kalendar tematik rejaga muvofiq dekanat tomonidan tuzilgan reyting nazorat jadvallari asosida o'tkaziladi. **YaN** semestrning oxirgi 2 haftasi mobaynida o'tkaziladi. **JN** va **ON** nazoratlarda saralash balidan kam ball to'plagan va uzrli sabablarga ko'ra nazoratlarda qatnasha olmagan talabaga qayta topshirish uchun, navbatdagi shu nazorat turigacha, so'nggi joriy va oraliq nazoratlar uchun esa yakuniy nazoratgacha bo'lgan muddat beriladi.

• Talabaning semestrda **JN** va **ON** turlari bo'yicha to'plagan ballari ushbu nazorat turlari umumiy balining 55 foizidan kam bo'lsa Yoki semestr davomida joriy, oraliq va yakuniy nazorat turlari bo'yicha to'plagan ballari yig'indisi 55 baldan kam bo'lsa, u akademik qarzdor deb hisoblanadi.

• Talaba nazorat natijalaridan norozi bo'lsa, fan bo'yicha nazorat turi natijalari e'lon qilingan vaqtdan boshlab bir kun mobaynida fakultet dekaniga ariza bilan murojaat etishi mumkin. Bunday holda fakultet dekanining taqdimnomasiga ko'ra rektor buyrug'i bilan 3 (uch) nafardan kam bo'lmagan tarkibda appelyasiya komissiyasi tashkil etiladi.

• Appelyasiya komissiyasi talabalarning arizalarini ko’rib chiqib, shu kunning o’zida xulosasini bildiradi.

Baholashning o’rnatilgan talablar asosida belgilangan muddatlarda o’tkazilishi hamda rasmiylashtirilishi fakultet dekani, kafedra mudiri, o’quv-uslubiy boshqarma hamda monitoring va ichki nazorat bo’limi tomonidan nazorat qilinadi.

Talabalar JN va ONdan to’playdigan ballarning namunaviy mezonlari

/r	Nazorat turidagi topshiriqlarning nomlanishi	Maksinial yig’ish mumkin bo’lgan ball	JN va ON ballar taqsinioti		
I. Joriy nazoratdagi ballar taqsinioti		**35 ball**	*1-JN 15 b)*	*N (14*	*N (6 b)*
Ma’ruza mashg’ulotlarda					
.	Talabaning ma’ruza mashg’ulotlarda faolligi, muntazam ravishda konspekt yuritishi uchun	6	3		
.	Mustaqil ta’lim topshiriqlarining o’z vaqtida va sifatli bajarilishi (keys-stadilar, esse, referat, taqdimot va boshqa turdagi mustaqil ta’lim topshiriqlari)	9	3	3	3
Amaliy mashg’ulotlarda					
	Talabaning mashg’ulotlarda faolligi, savollarga to’g’ri javob berganligi, amaliy topshiriq (masala, misol) larni bajarganligi uchun	20	9	9	2
II. Oraliq nazorat		**35 ball**			
.	Birinchi oraliq nazorat (amaliy mashg’ulot o’qituvchisi tomonidan qabul qilinadi)	15	Semestrning 10-haftasi		
.	Ikkinchi oraliq nazorat (ma’ruzachi va amaliy mashg’ulot o’qituvchisi tomonidan qabul qilinadi). Ikkinchi oraliq nazorat 2 bosqichda amalga oshiriladi. Birinchi bosqich, 10 ball-talaba yakka tartibda topshiriqlar	20	Semestrning 11-17-haftalar oralig’ida		

	oladi va himoya qiladi. Ikkinchi bosqich, 10 ball-talabalar kichik guruhlarga bo'linadi (har bir guruhda talabalar soni 5-7 tagacha bo'lishi mumkin), har bir guruhga alohida topshiriqlar beriladi va himoya qabul qilinadi. Topshiriqlar 2-3-haftalar oralig'ida talabalarga biriktiriladi. Guruhning faolligi, berilgan topshiriqni nazariy va amaliy jihatdan yoritilishi, xulosalarning mantiqiy bog'liqligi, kreativ mulohazalarning mavjudligi, huquqiy-normativ hujjatlarni bilishi va boshqa talablarga mosligi hisobga olinadi. Guruhdagi har bir talabaga 0-10 oralig'ida bir xil ball qo'yiladi. Himoya kafedra mudiri tomonidan tasdiqlangan grafik asosida dars mashg'ulotlaridan so'ng tashkil etiladi		
III. Yakuniy nazorat		**30 ball**	Semestrning oxirgi ikki haftasida
Jami:		**100 ball**	

Izoh.J.N. va O.N. bo'yicha talabalar reyting ballarining taqsinioti shu fan bo'yicha ishlab chiqilgan sillabusda batafsil keltiriladi.

Yakuniy nazorat "Yozma ish" shaklida belgilangan bo'lsa, u holda yakuniy nazorat 30 ballik "Yozma ish" variantlari asosida o'tkaziladi. Agar yakuniy nazorat markazlashgan test asosida tashkil etilgan bo'lib fan bo'yicha yakuniy nazorat "Test" shaklida belgilangan bo'lsa, u holda yakuniy nazorat quyidagi jadval asosida amalga oshiriladi:

	Ko'rsatkichlar	**YaN ballari**	
		maksinial ball	**o'zgarish oralig'i**
.	Fan bo'yicha yakuniy yozma ish nazorati	30	0-30
.	Fan bo'yicha yakuniy test nazorati	30	0-30

	Jami	30	0-30

Yakuniy nazoratni “Yozma ish” asosida o’tkazish
M E Z O N I

Talabalar fandan yakuniy nazoratni test topshiriqlari Yoki “Yozma ish” usulida topshiradilar va ularni o’zlashtirish ko’rsatkichi 0 dan 30 ballgacha baholanadi.Agar yakuniy nazorat “Yozma ish” usulida amalga oshirilsa, sinov ko’p variantli usulda o’tkaziladi. Har bir variantda mavzular yuzasidan beshtadan savol bo’lib, har bir savolga tegishli javoblar yoziladi. Har bir savolga yozilgan javoblar bo’yicha o’zlashtirish ko’rsatkichi 0-6 ball oralig’ida baholanadi. Berilgan har bir savolga talaba tomonidan fanning nazariy va uslubiy asoslari to’g’ri va to’liq yoritilsa, fan doirasida mustaqil fikrga ega bo’lsa va uni yozma shaklda bayon eta olsa, javobda mantiqiy yaxlitlikka erishilsa, o’zlashtirish umumiy ko’rsatkichi 0-6 ball bilan baholanadi.

Berilgan har bir savolga javob yozilmasa, noto’g’ri javob yozilsa Yoki o’quv adabiyotidan so’zma-so’z ko’chirib yozilsa, amaliy topshiriq shartida belgilangan amallar bajarilmasa, iqtisodiy hodisa va jarayonlarga ta’sir etuvchi omillar aniqlanmasa, xulosa yozilmasa o’zlashtirish ko’rsatkichi 0-1 ball bilan baholanadi. Yozma sinov bo’yicha umumiy o’zlashtirish ko’rsatkichini aniqlash uchun variantda berilgan savollarning har biri uchun yozilgan javoblarga qo’yilgan o’zlashtirish ballari qo’shiladi va yig’indi talabaning yakuniy nazorat bo’yicha o’zlashtirish bali hisoblanadi.

Yakuniy nazoratni “Test topshiriqlari” asosida o’tkazish
MEZONI

Agar yakuniy nazorat “Tyest topshiriqlari” asosida amalga oshirilsa, sinov ko’p variantli usulda kompyuter sinflarida o’tkaziladi. Har bir variantda 30 tadan test topshirig’i bo’lib, talabaga savollarning javobini belgilash uchun 30 daqiqa vaqt beriladi. Har bir test topshirig’iga talaba tomonidan berilgan javoblar 1 ball bilan baholanadi.

Mustaqil ish (__ soat)

Ushbu o‘quv fani bo‘yicha talabaning mustaqil ishi ma’ruzalar konspekti va tavsiya etilgan adabiyotlar hamda davriy jurnallar va Internet materiallari bilan ishlashni, laboratoriya ishlarini o‘tishga tayyorgarlik ko‘rishni , referatlar yozishni, kurs loyihasiga ijodiy yondashib, standart talablariga mos ravishda va hisoblash texnikasidan foydalanib mustaqil ish bajarishni o‘z ichiga oladi.

Mustaqil ishni bajarishdan asosiy maqsad — professor-o‘qituvchilarning bevosita rahbarligi va nazorati ostida talabalarni semestr davomida fanlarni uzluksiz o‘rganishini tashkil etish, olingan bilim, ko‘nikmalarini chuqur o‘rganib

mustahkamlash, kelgusidagi darslarga tayyorgarlik ko‘rish, aqliy mehnat madaniyatini, yangi bilimlarni mustaqil ravishda izlab topish va qabul qilishini shakllantirishdan iborat.

Talaba mustaqil ishini tashkil etishda «Informatika» fanining xususiyatlarini, shuningdek, har bir talabaning akademik o‘zlashtirish darajasi va qobiliyatini hisobga olgan holda quyidagi shakllardan foydalaniladi: ayrim nazariy mavzularni turli xil o‘quv adabiyotlari yordamida mustaqil o‘zlashtirish, berilgan mavzu bo‘yicha referat tayyorlash, laboratoriya va amaliy mashg‘ulotlarga tayyorgarlik ko‘rish, kurs ishini bajarish, bitiruv malakaviy ishini tayyorlash, anjumanlarga ma’ruza tezislarni tayyorlash, fan olimpiadalarida ishtirok etish, ilmiy jamiyatlar va to‘garaklarda ishtirok etish, kafedralarning ilmiy ishlarida ishtirok etish va hokazolar. Mustaqil ish uchun namunaviy mavzular:

XHTML murakkab formalari

JavaScriptda hodisalar bilan ishlash

VBScript elementlari

PHP da obyektga yo’naltirilgan dasturlashtirish imkoniyatlari

MySQL funksiyalari bilan ishlash

PHP da XML ning ishlatilishi.

ActiveX texnologiyasi.

ASP.NET va web xizmatlari**.**

Izoh: Mustaqil ta’lim soatlari hajmlaridan kelib chiqqan holda ishchi dasturda ma’ruza va amaliy mashg’ulotlarga tayyorgarlik ko’rish, fan bo’yicha talaba dunyoqarashini kengaytiradigan mustaqil ta’lim mavzulari shakllantiriladi.

Web dasturlash fanidan nazorat turlari, ularni o‘tkazish tartibi va baholash mezoni

Web dasturlashfanidan talabalarning bilim saviyasi va o‘zlashtirish darajasining Davlat ta’lim standartlariga muvofiqligini ta’minlash uchun quyidagi nazorat turlari o‘tkaziladi:

joriy nazorat – talabaning fan mavzulari bo‘yicha bilim va amaliy ko‘nikma darajasini aniqlash va baholash usuli bo‘lib, joriy nazorat “Web dasturlash” fanining xususiyatidan kelib chiqqan holda, laboratoriya va amaliy mashg‘ulotlarida og‘zaki so‘rov va uy vazifalarini tekshirish shakllarda o‘tkaziladi;

oraliq nazorat – semestr davomida o‘quv dasturining tegishli (fanning bir necha mavzularini o‘z ichiga olgan) bo‘limi tugallangandan keyin talabaning bilim va amaliy ko‘nikma darajasini aniqlash va baholash usuli. “Web dasturlash” fanidan oraliq nazorat 2 marta test shaklida belgilanadi;

yakuniy nazorat – semestr yakunida muayyan fan bo‘yicha nazariy bilim va amaliy ko‘nikmalarni talabalar tomonidan o‘zlashtirish darajasini baholash usuli.

Yakuniy nazorat asosan tayanch tushuncha va iboralarga asoslangan “Yozma ish” shaklida o‘tkaziladi.

“Web dasturlash” fani bo‘yicha talabaning semestr davomidagi o‘zlashtirish ko‘rsatkichi 100 ballik tizimda baholanadi.

Ushbu 100 ball nazorat turlari bo‘yicha quyidagicha taqsinilanadi:

yakuniy nazoratga – 30 ball, shundan talabalarning o‘quv fani bo‘yicha mustaqil ishiga 9 ball;

joriy va oraliq nazoratlarga – 70 ball, shundan talabalarning o‘quv fani bo‘yicha mustaqil ishiga 20 ball. Oraliq nazoratga 30 ball ajratiladi. Joriy nazoratga 40 ball ajratiladi.

Joriy nazoratda talaba 40 ballni yig‘ishi uchun amaliy mashg‘ulotlarda topshiriqlarni bajaradi. Har bir Amaliy mashg‘ulotda talaba 11 mavzu bo‘yicha berilgan topshiriqlarni bajaradi. Bitta mavzu bo‘yicha kamida 4 topshiriq beriladi. Har bir topshiriq uchun 0,5 ball ajratiladi, ya’ni 11x3 = 33 ball. SHu bilan birga talaba mustaqil ishida 21 ball to‘plashi kerak. Buning uchun talaba fan bo‘yicha 4 mustaqil ish topshiriqlarini bajaradi. Har bir mustaqil ish topshiriqlari uchun 5,25 ball beriladi. SHunday qilib, talaba joriy nazoratda amaliy mashg‘ulotda 33 ball va talaba mustaqil ishi bo‘yicha 21 ball to‘playdi.

Oraliq nazoratda talaba 30 ball to‘plashi kerak, shundan 12 balini mustaqil ish orqali to‘playdi. Oraliq nazoratda 30 ball yig‘ish uchun 2 marta yozma ish o‘tkaziladi. Web dasturlashfani bo‘yicha ma’ruzaga 36 soat ajratilgan bo‘lib, ma’ruza 18 mavzuni o‘z ichiga olgan. 10 mavzu o‘tib bo‘lingandan keyin birinchi oraliq nazorat o‘tkaziladi, buning uchun 10 ball beriladi, So‘ngra 8 mavzu o‘tiladi va yakuniy ma’ruzada ikkinchi oraliq nazorat o‘tkaziladi, bunda talaba 10 ball yig‘ish imkoniyatiga ega bo‘ladi. Talaba mustaqil ishda esa 10 mustaqil ish topshiriqlarini bajaradi. Har bir topshiriq uchun 2 ball ajratiladi.

Yakuniy nazoratga esa 30 ball ajratilgan, talaba shundan 21 balini “Yozma ish”da, 9 balini mustaqil ishda to‘playdi. Talaba 9 ballni to‘plashi uchun mustaqil ish topshiriqlarin to‘liq bajargan bo‘lishi, mustaqil olgan bilimlarini amalda qo‘llay olishi, mustaqil o‘rgangan mavzulari mohiyatini tushunishi kerak.

Talabaning “Web dasturlash” fani bo‘yicha o‘zlashtirish ko‘rsatkichi quyidagi mezonlar asosida aniqlanadi:

a) 86-100 ball uchun talabaning bilim darajasi quyidagilarga javob berishi lozim:

laboratoriya va amaliy mashg‘ulotlarda olingan natijalarga asosan xulosa va qaror qabul qilish;

laboratoriya ishini takomillashtirish, uni dasturiy ta’minotini yaratish bo‘yicha ijodiy fikrlay olish;

amaliy mashg‘ulotlarda berilgan topshiriqlarni yechimni mustaqil mushohada yuritish orqali topish;

ma’ruza darslarida olgan bilimlarini amalda laboratoriya va amaliy mashg‘ulotlarni bajarishda qo‘llay olish;

Web dasturlashfanidan berilgan tayanch tushuncha va iboralarni mazmun mohiyatini to‘liq tushunish;

Hisoblash texnikasi tarixi, axborot texnologiyasi, ishlash vositalari, axborot resurslari, axborot texnologiyasining iqtisodiy samaradorligi, jadvalli prosessorlardan foydalanish, elektron jadval tuzish, ma’lumotlar omborini tashkil qilish, lokal va global tarmoqlar, internetda va elektron pochtalarda ishlash

b) 71-85 ball uchun talabaning bilim darajasi quyidagilarga javob berishi lozim:

laboratoriya ishini takomillashtirish, uni dasturiy ta’minotini yaratish bo‘yicha ijodiy fikrlay olish;

amaliy mashg‘ulotlarda berilgan topshiriqlarni yechimni mustaqil mushohada yuritish orqali topish;

ma’ruza darslarida olgan bilimlarini amalda laboratoriya va amaliy mashg‘ulotlarni bajarishda qo‘llay olish;

Web dasturlashfanidan berilgan tayanch tushuncha va iboralarni mazmun mohiyatini to‘liq tushunish;

axborot va uning turlari, ularni yig‘ish,saqlash, uzatish va qayta ishlash usullari xaqidagi tushunchani;

kompyuterlar imkoniyatlari va ular bilan muloqotda bo‘lishni;

algoritmlash asoslari va dastur tuzish usullarini;

kompyuterda sonli tajribalar o‘tkazish va ular natijasini tahlil qilish;

matn muharrirlari yordamida matnlar ustida ishlashni;

ma’lumotlar jamg‘armasi va uni boshqarish tizimlari bilan ishlash haqida tasavvurga ega bo‘lish.

v) 56-70 ball uchun talabaning bilim darajasi quyidagilarga javob berishi lozim:

Web dasturlashfanidan berilgan tayanch tushuncha va iboralarni mazmun mohiyatini to‘liq tushunish;

Web dasturlashning hisoblash texnikasi tarixi, axborot texnologiyasi, ishlash vositalari, axborot resurslari, axborot texnologiyasining iqtisodiy samaradorligi, jadvalli prosessorlardan foydalanish, elektron jadval tuzish, ma’lumotlar omborini tashkil qilish, lokal va global tarmoqlar, internetda elektron pochtalarda ishlash haqida tasavvurga ega bo‘lish.

Foydalaniladigan adabiyotlar ro'yxati

Asosiy adabiyotlar

1. Karimov I.A. Yuksak ma'naviyat - yengilmas kuch. - T.: Ma'naviyat, 2008.

2. M.Aripov, M.Fayziyeva, C.*floTToeB.*Web dasturlash. Uslubiy qo'llanma .T.; "Faylasuflar jamiyati". 2013.

3. R.Xamdamov va boshkalar. Ta'limda ax borot texnologiyalari. Uslubiy kullanma.Toshkent. Uzbekistan milliy ensiklopediyasi" 2010

4. Robin Nixon. Learning PHP, MySQL&JavaScript width JQuery, CSS & HTML5, 4th Edition. O'Reilly Media. United States of America. 2015.

5. Lebedov S V..Web-dizayn..Ucheb.pos.Moskva, ZAO «Izdatelskiy dom Alyans press», 2004

6. Jamsil K. Lalani, S., S. Weekly Programming Web professionals Trans, from Engiish. A.I.Panasyuk - Mn. LLC "Potpourri", 1997. - 632 p.: III.

7. Malchuk EV. HTML and CSS. self-teacher

Qo'shimcha adabiyotlar:

1. M.Mamarajabov, S.Tursunov. Kompyuter grafikasi va Web dizayn. Darslik.T.:"Cho'lpon"\ 2013

2. U. Yuldashev, M.Mamarajabov, S.Tursunov. Pedagogik Web dizayn. Ukuv kullanma.T: "Voris"\ 2013

3. VG Olifer, N. A.OIifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.

4. R. Darnell. "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.

5. Dmitriev M. JavaScript. Quick start. - St. Petersburg.: BHV St. Pereburg, 2002.

Internet saytlar:

1. http://www.intuit.ru
2. http://etuit.uz/
3. www.tuit.uz
4. www.ziyonet.uz

5-Semestr uchun

1-ma'ruza. PHP. Server tomondan dasturlash. PHP ga kirish. PHP ni o'rnatish va testlash

Reja:

1.1. PHP ga kirish.

1.2. PHP imkoniyatlari.

1.3. Dasturiy vositani sozlash va o'rnatish.

Kalit so'zlar: PHP dasturlash tili, server tomonda dasturlash, denver paketi

1.1. PHP ga kirish

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

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

1995 yili Rasmus PHPni o'zining HTML formalari bilan ishlaydigan boshqa dastur bilan umumlashtirib PHP/FI Version 2 (”Form Interpretator“) hosil qildi. 1997 yilga borib PHP dan foydalanuvchi saytlar 50 mingdan oshdi. SHundan so'ng veb texnologiya ustalari PHP g'oyasi asosida mukammal til yaratishga Ziva Suraski va Endi Gutmans asoschiligida kirishildi. PHPni samarali deb hisoblanmagani uchun deyarli noldan boshlab, mavjud S va Perl tillaridan ibrat

olib PHP3 talqinini yaratildi. 1999 yilga kelib PHP asosida qurilgan saytlar milliondan oshib ketdi. 2000 yilda esa Zend Technologies shirkati yangi ko'pgina funksiyalarni qo'shgan holda PHP4 sharhlovchisini yaratdi.

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

1.2. PHP imkoniyatlari

«PHP da har qanday dastur bajarsa bo'ladi», – degan edi uning yaratuvchisi. Birinchi navbatda PHP tili server tomonidan bajariladigan skriptlar yaratish uchun foydalaniladi va aynan shuning uchun u yaratilgan. PHP tili ixtiyoriy CGI-skriptlari masalalarini yechishga va bundan tashqari html formali ma'lumotlarni qayta ishlashga hamda dinamik ravishda html sahifalarni ishlab chiqishga qodir. Biroq PHP tili foydalaniladigan boshqa sohalar ham mavjud. Bu sohalarni biz uchta asosiy qismga bo'larniz:

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

kompyuterimizga qanday o’rnatamiz, biz bu haqida keyinroq ko’rib o’tamiz.

Ikkinchi soha – buyruqlar satrida bajariladigan skriptlarni yaratish. Ya’ni PHP tili yordamida biror-bir kompyuterda brauzer va web-serverlardan mustaqil ravishda o’zi bajariladigan skriptlarni ham yaratish mumkin. Bu ishlarni bajarish uchun hyech bo’lmaganda PHP-parser (bu holatda biz uni buyruqlar satri interpretatori (CLI, command line interpreter) deb ataymiz) talab etiladi. Bunday ishlash uslubi turli masalalarni rejalashtirish yordamida bajarilishi uchun kerak bo’lgan skriptlar Yoki oddiy matnni qayta ishlash uchun kerak bo’lgan masalaga o’xshash ishlaydi.

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

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

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

PHP dasturlash tilida dasturlashning ikkita hammabop paradigmalari ishlatiladi, bular prosedurali va obyektli dasturlash. PHP4 dasturlash tili prosedurali dasturlashni butunlay qo’llab quvvatlaydi, biroq obyektli stildagi dasturlarni ham qo’llasa bo’ladi. PHP5 dasturlash tilining birinchi testlash

versiyasida PHP4 dasturlash tilida uchraydigan obyektga yo'naltirilgan dasturlash modellarining kamchiliklari to'ldirilgan. SHunday qilib, hozirda tanish bo'lib ulgurgan ishlash prinsipini tanlash kerak.

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

PHP dasturlash tilini turli dasturlash tillari o'rtasidagi o'zaro aloqasiga diqqatni qaratsak, bunga Java dasturlash tilini aytib o'tish kerakki, Java dasturlash tili obyektlarini PHP tili o'z obyektlari sifatida qaraydi. Obyektlarga murojaat sifatida CORBA kengaytmasidan foydalaniladi.

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

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

1.3. Dasturiy vositani sozlash va o'rnatish

Yuqorida PHP tili imkoniyatlarini, qo'llanilish sohalarini muhokama qildik va tarixini o'rgandik. Endi dasturiy vositani o'rnatishga kerak bo'lgan uskunalar majmuini ko'rib o'tsak. Modomiki, asosiy kursning amaliyoti sifatida biz quyidagi

masalalarni ko’rib chiqamiz: kliyent-server texnologiyasi sifatida ishlanadigan masalalar, mos ravishda skriptlar yaratilishida qo’llanilishi, serverlarni qayta ishlash. Bular uchun bizga web-server hamda PHP tili interpretatori kerak bo’ladi. Web-server sifatida, masalan, web-mutaxassislar o’rtasida mashhur bo’lgan Apache serverni olarniz. Dastur natijasini ko’rish uchun web-brauzer kerak bo’ladi, bunga misol Internet Explorer.

Denver Distributivi

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

PHP dasturlash tili, server tomonda dasturlash, denver paketi

Takrorlash uchun savollar:

1. Qanaqa dasturlash tillarini bilasiz?
2. PHP dasturlash tili imkoniyatlari haqida gapiring.
3. Server tomonda dasturlash deganda nimani tushunasiz?
4. PHP ni ishga tushirish, Denver paketi va undan foydalanish.

2-ma’ruza. PHP asosiy tuzilishi. Ma’lumotlar tiplari. Ifodalar. Jarayonlarni boshqarish

Reja:

2.1. PHP asosiy tuzilishi.

2.2. Asosiy sintaksislar.

2.3. Ma’lumotlar tiplari.

2.4. Alternativ sintaksislar.

Kalit so’zlar: PHP kod sintaksisi, ma’lumot tiplari, ifodalar, jarayonlarni boshqarish

2.1. PHP asosiy tuzilishi

Ko’p hollarda *PHP* tilini interpretatori ishlayotganligini tekshirib ko’rish uchun tuziladigan dastur eng sodda dastur deb ataladi. Hozir biz *PHP* tilidagi ushbu dasturni chuqur o’rganamiz hamda uni boshqa dasturlash tillari Ci, Perl va JavaScript lardan farqli tomonini tekshiramiz. Ushbu **misolni** ko’ramiz:

```
<html>
  <head>
  <title> Misol </title>
  </head>
  <body>
  <?php
  echo "<p> Salom, bu men – PHP skript! </p>";
  ?>
</body>
</html>
```

Bu *PHP* dasturlash tilining maxsus kodli teglari yordamida tuzilgan sodda html-fayldir.

Yuqorida aytib o’tganimizdek, *PHP* dasturlash tili Ci va Perl dasturlash tiliga o’xshash. Biroq keltirilgan dastur Ci va Perl dasturlash tilidagi dasturdan ancha katta farq qiladi. Bu yerda HTML sahifaga chiqarish uchun bir qator maxsus buyruqlarni yozish shart emas. Bevosita *PHP*-kod asosida qurilgan biror vazifani bajaradigan HTML-*skript* yoziladi (bizni misolda ekranda chiqarilgan matn). *PHP* dasturlash tilining Ci va Perl dasturlash tillaridan kamchiligi shuki, murakkab skriptlarni *PHP* dasturlash tili ancha sekin bajaradi.

PHP-skriptlar – bu serverda bajariladigan va qayta ishlanadigan dasturlardir. Bu skriptlarni JavaScript tipidagi skriptlar bilan taqqoslash mumkin emas, chunki JavaScript tilidagi skriptlarda yozilgan buyruqlar faqat kliyent kompyuteridagina bajariladi. Kliyent kompyuterida va server kompyuterida bajariladigan skriptlarning farqi nimada? Agarda skript serverda qayta ishlansa, mijoz kompyuteriga faqatgina natija yuboriladi. Masalan, agarda serverda skript bajarilayotgan bo’lsa, yuqorida keltirilganga o’xshab mijoz HTML-sahifa ko’rinishdagi natijani oladi:

```
<html>
   <head>
   <title> Misol </title>
   </head>
   <body>
   <p> Salom, bu men – PHP skript! </p>
   </body>
</html>
```

Bu holatda mijoz qanday kod bajarilayotganini bilmaydi. O’z serveringizni HTML-fayllarni *PHP* prosessori qayta ishlaydigan qilib sozlab olishingiz ham mumkin. Ya’ni kliyentlar oddiy HTML-faylni qabul qildimi Yoki skript natijasini ko’rdimi buni bila olmaydi. Agarda skript kliyent kompyuterida qayta ishlansa (masalan, JavaScript tilidagi dastur), u holda kliyent skript kodidan iborat HTML-sahifani ko’radi.

Biz yuqorida aytib o’tgandikki, *PHP-skriptlar HTML*-kod ichida yoziladi. Qanday qilib degan savol tug’iladi. Buning bir nechta usullari mavjud. Bulardan biri birinchi misolda keltirilganidek, *<?php* tegi bilan boshlanib *?>* tegi bilan tugagan sintaksis. Bunday ko’rinishdagi maxsus teglar HTML va *PHP* rejimidagina ishlatiladi. Bu sintaksis *PHP* tilini *XML* hujjatlari bilan birgalikda ishlaydigan dasturlarida juda ma’qul ko’riladi (masalan, XHTML tilida yozilgan dasturlarda). Biroq bazan quyidagi alternativ variantdan foydalansa ham bo’ladi(echo "Some text" buyrug’i «Some text» matnini ekranga chiqaradi.):

```
<? echo "Bu PHP tilida
        oddiy qayta ishlashning
        instruksiyasi"; ?>

<script language="php">
 echo "Bir nechta redaktorlar
      (FrontPage) quyidagicha
      qabul qilishadi";
</script>

<% echo " ASP texnologiyasidagi tegdan
        ham foydalansa bo’ladi"; %>

```

Bu keltirilgan usullardan birinchisi har doim ham bajarilavermaydi. Undan foydalanish uchun qisqa teglarni ishlatish kerak, Yoki *PHP*3 uchun *short_tags*() funksiyani ishlatish kerak, Yoki *PHP* tilining konfigurasion fayliga short_open_tag buyruqni o’rnatish kerak, Yoki *PHP* dasturlash tilida *enable-short-tags* parametr bilan kompilyasiya qilish kerak. Agarda php.ini-dist buyruqqa yuqoridagilar avtomatik qo’shilgan bo’lsa, u holda qisqa teglardan foydalanish tavsiya etilmaydi. Ikkinchi usul xuddi o’rniga qo’yishga o’xshaydi, masalan, JavaScript kodlari va uning uchun mos html teglar. SHuning uchun undan har doim foydalanish mumkin, lekin bu noqulayligi uchun kamdan-kam ishlatiladi. Uchinchi usuldan

faqat ASP texnologiyasidagi teglar *asp_tags* konfigurasiyasida ishlatilgandagina foydalaniladi.

PHP dasturlash tili faylni qayta ishlayotganda u oddiy matnni *PHP* kod interpretasiya qilishi kerak bo'lgan maxsus teglarni uchratmaguncha qaytarib beradi. Interpretator haqida gapirganda u topilgan barcha kodni yopiladigan teggacha bajaradi, so'ng yana oddiy matn qaytariladi. Bu mexanizm *PHP*-kodni HTML sahifaga aylantiradi, ya'ni barcha *PHP* teglardan tashqari barcha matnlarni o'zgarishsiz saqlaydi va ichkaridagilarni esa interpretasiyalaydi. Yana shuni aytish kerakki, php-fayl *CGI*-skriptga o'xshamaydi. php-fayl bajarilishi shart emas, Yoki yana qandaydir belgilanadi.

php-faylni serverda qayta ishlash uchun jo'natishda server tomonidan brauzer satrida bu faylni yo'lini ko'rsatish shart. *PHP* skriptlar www orqali kirishga ruxsat etilgan joyda joylashishi shart. Agarda php-fayl lokal kompyuterda mavjud bo'lsa, u holda uni buyruqlar satri interpretatori yordamida qayta ishlash mumkin.

Xulosa.

SHunday qilib, biz *PHP* dasturlash tili haqida ma'lumotga ega bo'ldik, u qanday dunyoga kelgan va tarqalgan, uni qanday va qayerda foydalanilishini o'rgandik, dasturiy vositani o'rnatdik hamda uni ishlashi uchun barcha sozlashlarni bajardik va *php*-dastur nimalardan tashkil topishini angiadik. Keyingi bo'limlarda biz *PHP* dasturlash tilining asosiy sintaksislarini ko'rib chiqamiz hamda bir qancha amaliy masalalarni hal etamiz.

2.2. Asosiy sintaksislar

Instruksiyani bir nechta qismga bo'lib ko'rib chiqamiz, ya'ni kommentariylar yaratish, o'zgaruvchilar, o'zgarmaslar va ma'lumot tiplari, operatorlarga.

Biz endi *PHP* dasturlash tilining asosiy sintaksis elementlarini o'rganishga o'tamiz. Misol sifatida elektron maktub tayyorlash masalasini ko'rib o'taylik.

Uning ma’nosi quyidagidan iborat.

Faraz qilarnizki, sizda qandaydir e’lon va e’lonni jo’natishingiz kerak bo’lgan bir nechta odamlar mavjud bo’lsin. Buning uchun siz e’lonni ichida o’zgaradigan (qabul qiluvchi bilan bog’liq bo’lmagan) bir nechta parametrlari mundarijasi bilan tayyorlaysiz.

Birinchi navbatda PHP dasturlash tili sintaksisiga nisbatan nimalarni bilish kerak. Bu HTML-kod ichiga o’rnatilgan va PHP dasturlash tilidagi koddir, uni interpretator farqlay biladi. Avvalgi bo’limlarda bular haqida aytib o’tgandik. Hammasini qaytarib o’tmaymiz, faqat biz ko’p hollarda misollarda <?php ?> variant o’rniga qisqartirilgan <? ?> teglardan foydalanishni aytib o’tamiz.

Instruksiyalarni ajratilishi.

PHP dasturlash tilidagi dastur(ixtiyoriy dasturlash tilidagi) – bu buyruqlar (instruksiyalar) to’plarnidir. Dasturni qayta ishlash uchun bir buyruqni boshqa buyruqdan farqini bilish kerak. Buning uchun maxsus sinivollar – ajratgichlardan foydalaniladi. PHP dasturlash tilida instruksiyalarni xuddi Ci Yoki Perl dasturlash tillari kabi ajratiladi, ya’ni har bir ifoda nuqtali vergul (“;”) bilan tugaydi.

«?>» yopiladigan teg ham instruksiyani tugashini angiatadi, shuning uchun undan oldin nuqtali vergul qo’yilmaydi. Masalan, quyidagi ikki fragmentlar ekvivalentdir:

```
<?php
echo "Hello, world!"; // buyruqlar oxirida nuqtali vergul qo’yish shart
?>
<?php
echo "Hello, world!" ?>
<!-- "?>" borligi uchun
   nuqtali vergul tashlab ketildi -->
```

Kommentariylar.

Ko’p hollarda dastur tuzganda kodni tushunarli bo’lishi uchun unga qandaydir izoh-***kommentariylar*** qo’yish kerak bo’lib qoladi. Bu holat katta hajmdagi dasturlar yaratganda hamda agarda bitta dastur ustida bir nechta

dasturchi ishlayotganda juda muhim. Kommentariylar dasturning kodi tushunarli bo'lishi uchun yoziladi. Bundan tashqari masalani qismlarga ajratib hal qilinganda ishning kamchiligi bor joyida keyinchalik esdan chiqmasligi uchun kommentariya yozib qo'yiladi. Barcha dasturlash tillarida dastur ichiga kommentariya qo'shish imkoniyati mavjud. *PHP* dasturlash tili bir qancha ko'rinishdagi kommentariylarni qo'llab quvvatlaydi: Ci, C++ dasturlash tillari stilidagi hamda Unix qobig'idagi kommentariylar. // va # belgilar bir satrli kommentariylarni angiatsa, /* va */ belgilar esa mos ravishda ko'p satrli kommentariylarning boshlanish va tugashini angiatadi.

Misol:PHP dasturlash tilida kommentariyning qo'llanilishi

```
<?php
echo "Meni ismim Alisher";
  // Bu bir satrli kommentariy
  // S++ dasturlash tili stilidagi
echo "Meni familiyam Boliyev";
/* Bu ko'p satrli kommentariy. Bu yerga bir qancha satr yozish mumkin. Dastur bajarilish jarayonida bu yerdagi barcha yozuvlar (kommentariylangan), o'qilmaydi. */
echo "Men PHP dasturlash tilini INTUIT.ru dan o'rganyapman";
  # Bu kommentariy
  # Unix qobig'idagi kommentariy.
?>
```

o'zgaruvchilar, o'zgarmaslar va operatorlar

Har bir dasturlash tilida muhim elementlardan biri bu *o'zgaruvchilar*, *o'zgarmaslar* va ular qo'llaniladigan *operatorlar*dir. PHP dasturlash tili bu elementlarni qanday belgilashi va qayta ishlashini ko'rib chiqamiz.

O'zgaruvchilar

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

O’zgaruvchilar nomlari registrlarni farqlaydi, ya’ni $my_var hamda bosh harfli $My_var o’zgaruvchilari turli xil o’zgaruvchilardir.

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

PHP4 dasturlash tilida bulardan tashqari o’zgaruvchiga qiymat o’zlashtirishning yana bir usuli mavjud: ssылka bo’yicha *o’zlashtirish*. Ssылka bo’yicha o’zgaruvchiga qiymat o’zlashtirish uchun uni nomi bo’lishi shart, ya’ni u qandaydir o’zgaruvchini taqdim etishi kerak. Bir o’zgaruvchi qiymatini boshqa o’zgaruvchiga Ssылka bo’yicha *o’zlashtirish* uchun birinchi o’zgaruvchi oldiga ampersand & belgisi qo’yish shart.Bunga yuqoridagi misolni ko’rib chiqamiz, faqat first o’zgaruvchi second o’zgaruvchiga ssылka bo’yicha o’zlashtiriladi:

Misol. Ssылkalar bo’yicha o’zlashtirish.

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

Bu skriptni natijasi esa quyidagicha bo’ladi:

- first nomli o’zgaruvchi qiymati New text ga teng.
- second nomli o’zgaruvchi qiymati New text ga teng.

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

O'zgarmaslar

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

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

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

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

Misol. PHP dasturlash tilida o'zgarmaslar.

```
<?php
// o'zgarmasni aniqlaymiz PASSWORD
define("PASSWORD","qwerty");
// registrlanmagan PI o'zgarmasni qiymatini aniqlaymiz 3.14
define("PI","3.14", True);
// PASSWORD o'zgarmas qiymatini olarniz,  ya'ni qwerty
echo (PASSWORD);
// bu ham qwerty ni chiqaradi
```

echo constant("PASSWORD");

echo (password);

/* password ni chiqaradi va biz registrlangan o’zgarmas PASSWORD ni kutgandik.*/

echo pi;

// 3.14 ni chiqaradi, chunki o’zgarmas PI registrlanmagan va aniqlangan.

?>

Dasturchi tomonidan o’zgaruvchilardan tashqari yuqorida aytib o’tganimizdek *PHP* dasturlash tilida mavjud o’zgarmaslar ham interpretator tomonidan aniqlanadi. Masalan, __FILE__ o’zgarmas dastur bajarilish jarayonida fayl nomini (va fayl yo’lini), __FUNCTION__ funksiya nomidan tashkil topadi, __CLASS__ - sinf nomi, PHP_VERSION – *PHP* dasturlash tili interpretatori versiyasini o’zida saqlaydi. Bunday o’zgarmaslarning barcha ro’yxatini *PHP* dasturlash tili uchun mo’ljallangan qo’llanmalardan topish mumkin.

Amallar.

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

2.1-jadval. Arifmetik amallar.		
Belgilanishi	**Nomlanishi**	**Misol**
+	Qo’shish	$a + $b
-	Ayirish	$a - $b
*	Ko’paytirish	$a * $b
/	Bo’lish	$a / $b
%	Bo’lishdagi qoldiq	$a % $b

2.3-jadval. O’zlashtirish amallari.			
Belgilanishi	**Nomlanishi**	**Izox**	**Misol**
=	O’zlashtirish	*Operatordan* o’ng tomonda turgan o’zgaruvchilar ustida bajarilgan amallardan hosil bo’lgan natija qiymati o’zlashtiriladi.	$a = ($b = 4) +5; ($a 9 ga teng, $b 4 ga teng)
+=		Qisqartirish. O’zgaruvchiga son qo’shiladi va keyin natija o’zlashtiriladi.	$a += 5; ($a = $a + 5

2.2-jadval. Satrli amallar.		
Belgilanishi	**Nomlanishi**	**Misol**
	Konkatenasiya (*satrlarni qo’shish*)	$c = $a . $b (bu $c satr $a va $b satrlardan iborat)

			ifodaga ekvivalent;)
.=		O’zlashtirish va konkatenasiya amallari kombinasiyasini qisqartirilgan shakli(dastavval satrlar qo’shiladi, so’ngra hosil bo’lgan satr o’zgaruvchiga o’zlashadi).	$b = "Hammaga "; $b .= "salom"; ($b = $b . "salom" ifodaga ekvivalent;) Natijasi: $b="Hammaga salom"

2.4-jadval. Matiqiy amallar.

Belgilanishi	**Nomlanishi**	**Izox**	**Misol**
and	VA	$a va $b rost (True)	$a and $b $a && $b $a or $b $a || $b $a xor $b ! $a
&&	VA		
or	Yoki	$a Yoki $b o’zgaruvchilardan hyech bo’lmaganda bittasi rost bo’lsa (ikkalasi ham rost bo’lishish mumkin).	
||	Yoki		
xor	Inversiya Yoki	O’zgaruvchilardan bittasi rost bo’lsa. Agarda ikkalasi ham rost bo’lsa inversiyalanadi.	
!	Inversiya (NOT)	Agarda $a=True, u holda !$a=False va aks holda teskarisi bo’ladi.	

2.5-jadval. Taqqoslash amallari.

Belgilanishi	**Nomlanishi**	**Izox**	
==	Tenglik	O’zgaruvchilar qiymatlari teng	$a == $b
===	Ekvivalentlik	O’zgaruvchilar qiymatlari va tiplari	$a === $b

		teng	
!=	Tengsizlik	O’zgaruvchilar qiymatlari teng emas	$a != $b
<>	Tengsizlik		$a <> $b
!==	Noekvivalentlik	O’zgaruvchilar ekvivalent emas	$a !== $b
<	Kichik		$a < $b
>	Katta		$a > $b
<=	Kichik Yoki teng		$a <= $b
>=	Katta Yoki teng		$a >= $b

2.6-jadval. Inkrement va dekrement amallari.

Belgilanishi	Nomlanishi	Izox	Misol
++$a	Pre-*inkrement*	$a qiymati birga oshiriladi va $a qiymati qaytariladi	<? $a=4; echo "4 bo’lishi shart:" .$a++; echo "6 bo’lishi shart:" .++$a; ?>
$a++	Post-*inkrement*	$a qiymati qaytariladi va so’ngra $a qiymati birga oshiriladi	
--$a	Pre-*dekrement*	$a qiymati birga kamaytiriladi va $a qiymati qaytariladi	
$a--	Post-*dekrement*	$a qiymati qaytariladi va so’ngra $a qiymati birga kamaytiriladi	

2.3. Ma’lumotlar tiplari

PHP dasturlash tili sakkizta sodda *ma’lumot tiplar*ini qo’llab quvvatlaydi:

To’rttasi skalyar *tiplar*:

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

Ikkitasi aralish *tiplar*:

- *array (massiv)*;

- *object (obyekt).*

Ikkitasi maxsus *tiplar*:

- *resource (resurs)*;
- *NULL.*

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

Boolean tipi(Bul Yoki mantiqiy tip).

Bu sodda tip qiymatni rost ekanligini ifodalaydi, ya'ni o'zgaruvchi faqat ikkita qiymat qabul qiladi – rost TRUE Yoki yolg'on FALSE.

Mantiqiy tiplarni aniqlash uchun TRUE Yoki FALSE kalit so'zlaridan foydalanamiz. Bu ikkala tiplar registrlanmagan.

Misol. Mantiqiy tip.

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

Mantiqiy tiplar turli *boshqariladigan konstruksiyalar*da (sikllar, shartlar va shunga o'xshash, bular haqida keyinroq aytib o'tamiz) qo'llaniladi. Bir qancha amallar (masalan, tengiik amali) ham mantiqiy tip qabul qilishi mumkin, ya'ni faqat ikki qiymat rost Yoki yolg'on qiymatni qabul qiladi. Ular *boshqariladigan konstruksiyalar*da shartlarni tekshirish uchun qo'llaniladi. Masalan, shartli konstruktorda amallar Yoki o'zgaruvchilar qiymati haqiqiyligini tekshiradi va natijadan qat'iy nazar shu Yoki boshqa amallarni bajarilishini tekshiradi. Bu yerda shart rost Yoki yolg'on bo'lishi mumkin, chunki *mantiqiy tip amallari* va *o'zgaruvchilar* ko'rsatilgan.

Misol. Mantiqiy tiplarning qo'llanilishi.

```
<?php
// '==' amal tengiikka tekshiradi mantiqiy qiymatni qaytaradi
```

```
if ($know == False) { // agar $know qiymat false bo’lsa
echo "PHP dasturlash tilini o’rgan!";
}
if (!$know) { // xuddi yuqoridagidek $know qiymati false bo’ladi
echo " PHP dasturlash tilini o’rgan!";
}
```

/* == amal $action o’zgaruvchi qiymati bilan "PHP dasturlash tilini o’rganish!" satrni ustma-ust tushishini tekshiradi. Agar ustma-ust tushsa true qiymatni qaytaradi, boshqa holda false ni qaytaradi. Agar true ni qaytarsa figurali qavs ichidagi amallar bajariladi. */

```
if ($action == " PHP dasturlash tilini o’rganish ")
{ echo "O’rganishni boshladim";}
?>
```

Integer (butun) tipi.

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

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

```
<?php
# o’nlik raqam
$a = 1234;
# manfiy son
$a = -123;
# sakkizlik son (o’nlik sistemasidagi
# 83 songa ekvivalent)
$a = 0123;
# o’n oltilik son (o’nlik sistemasidagi
```

26 songa ekvivalent)

$a = 0x1A;

?>

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

Agar siz *butun son* chegarasidan tashqari biror qiymat bersangiz interpretator bu sonni *qo’zg’aluvchan vergulli son*ga o’zgartiradi. Xuddi shunday *butun son* chegarasidan tashqari chiqib ketadigan biror amal bajarsangiz ham bu sonni *qo’zg’aluvchan vergulli son*ga o’zgartiriladi.

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

Float (qo’zg’aluvchan vergulli son) tipi.

*Qo’zg’aluvchan vergulli son*lar (ular ikki karra aniqlik Yoki haqiqiy sonlardir) quyidagi sintaksislar yordamida aniqlanishi mumkin:

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

?>

*Qo'zg'aluvchan vergulli son*ni o'lchami ham platformaga bog'liq, lekin qoidaga ko'ra maksinial qiymati ~1.8e308 aniqlik bilan 14 xonali raqam atrofida bo'ladi.

Resource (resurslar) tipi.

Resurs – bu tashqi resursga (masalan, ma'lumotlar bazasi bilan bog'lanish) ссылka orqali bog'langan maxsus o'zgaruvchidir. Resurslar maxsus funksiyalar (masalan, mysql_connect(), pdf_new() va shunga o'xshashlar) yordamida yaratiladi va foydalaniladi.

Null tipi.

Maxsus *NULL* qiymati *o'zgaruvchi*ni qiymatga ega emasligi haqida ogohlantiradi.

O'zgaruvchi NULL qiymat qabul qiladi, agarda:

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

*NULL tip*li faqat bitta qiymati mavjud – registrga sezgirligi kichik NULL kalit so'zidir.

Masalaning yechilishi.

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

```
<?
// bizning yozuvimiz
// o’zgarmas bo’lsin.
define("SIGN","Hurmat bilan, Azamat");
// odamlar va hodisalar massivini beramiz
$names = array("Ivan Ivanovich",
               "Petr Petrovich",
               "Semen Semenovich");
$events = array(
    "f" => "ochiq eshiklar kuni",
    "o" => "ko’rgazmaning ochilishi",
    "p" => "bitiruvchilar bali");
// taklifnoma matnini tuzamiz.
$str = "Hurmatli, $names[0]";
$str .= "<br> Sizni taklif etamiz ".
        $events["f"];
$str .= "<br>" . SIGN;
echo $str; // matnni ekranga chiqaramiz.
?>
```

Xulosa.

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

2.4. Alternativ sintaksislar

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

Masalan, *if* shart operatori sintaksisini quyidagicha tfodalash mumkin:

if (ifoda) : *bajariladigan_blok* endif;

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

Misol. Alternativ sintaksisdan foydalanish.

```
<?php
$names = array("Karim","Salim","Sodiq");
if ($names[0]=="Karim"):
?>
Salom, Karim!
<?php endif ?>
```

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

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

```
endif;
?>
```

Takrorlash uchun savollar

1. PHP kodni tuzilishini tushuntirib bering.
2. PHP da qanday ma’lumot tiplaridan foydalaniladi?
3. PHP da ifodalar qanday e’lon qilinadi?
4. Alternativ sintaksislar deganda nimani tushunasiz?

3-ma'ruza. PHP da satr va massivlar bilan ishlash

Reja:

3.1. PHP da satr tipi

3.2. PHP da massivlar tipi

Kalit so'zlar: PHP da satr tipi, massiv tipi

3.1. PHP da satr tipi

String (satr) tipi

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

PHP dasturlash tilida satrlar uchta turli xil usullarda aniqlanadi:

- *bittalik qo'shtirnoqlar* yordamida (‘’);
- *qo'shtirnoqlar* yordamida (“”);
- *heredoc-sintaksisi* yordamida.

Bittali tirnoqlar

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

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

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

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

3.2. PHP da massivlar tipi

Array (massiv) tipi.

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

*Massivlar*ni *array*() konstruksiyasi yordamida aniqlanadi Yoki elementlariga qiymat berish bilan aniqlanadi.

array() konstruksiyasi yordamida aniqlash.

```
array ([key] => value,
      [key1] => value1, ... )
```

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

Misol. PHP dasturlash tilida massivlar.

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

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

Misol. PHP dasturlash tilida massivlar.

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

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

Kvadrat qavs sintaksisi yordamida aniqlash.

Massivga qiymat berish orqali massiv yaratish mumkin. Biz yuqorida aytib o’tganimizdek, massiv elementi qiymatiga ega bo’lish uchun kvadrat qavs ichiga uning kaliti ko’rsatilishi kerak, masalan, $book["php"]. Agarda yangi kalit va yangi qiymat ko’rsatsangiz quyidagicha bo’ladi: $book["new_key"]="new_value" hamda massivga yangi element qo’shiladi. Agarda kalitni ko’rsatmay faqat qiymatni o’zlashtirsak, ya’ni $book[]="new_value", u holda massivga yangi element qo’shiladi va uni kaliti mavjud maksinial qiymatga bir qo’shiladi. Agarda biz qiymat bergan *massiv* yaratilmagan bo’lsa, u holda biz qiymat bergandan keyin u *yaratiladi*.

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

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

```
<?php
$books = array ("php" =>"PHP users guide",12 => true);
$books[] = "Book about Perl"; /* 13-kalit(indeks) bilan yangi element qo’shildi,
 bu quyidagiga ekvivalent $books[13] = "Book about Perl";
$books["lisp"] = 123456; /* Bu massivga yangi "lisp" kalitli 123456 qiymali yangi
 element qo’shish*/
unset($books[12]); // Bu 12-kalitli elementni massivdan o’chirish
unset ($books); // massivni butunlay o’chirish
?>
```

Bo’sh kvadrat qavsdan foydalanganda kalitning maksinial qiymati massivda mavjud oxirgi qayta indekslangan kalitlar orasidan qidiriladi. Massivni ***array_values***() funksiyasi yordamida ***qayta indekslash*** mumkin.

Misol. Massivni qayta indekslaymiz.

```
<?php
$arr =
 array ("a","b","c");
/* "a", "b" va "c" qiymatli massivni yaratamiz. Bu yerda kalit ko’rsatilmagan biroq mos ravishda ular 0,1,2 bo’ladi. */
print_r($arr); // massivni ekranga chiqaramiz (kaliti va qiymatini)
unset($arr[0]);
unset($arr[1]);
unset($arr[2]);
// massivdan hamma elementini o’chiramiz
print_r($arr);  // massivni ekranga chiqaramiz (kaliti va qiymatini)
```

$arr[] = "aa"; // massivga yangi element qo’shamiz. Uni indeksi(kaliti) 3 bo’ladi, 0 emas.

print_r($arr);

$arr =

array_values($arr); // massivni qayta indekslaymiz.

$arr[] = "bb"; // bu elementni kaliti 1 bo’ladi.

print_r($arr);

?>

Bu skriptning natijasi quyidagicha bo’ladi:

Array ([0] => a [1] => b [2] => c)

Array ( )

Array ([3] => aa)

Array ([0] => aa [1] => bb)

Takrorlash uchun savollar

1. PHP da qanday tiplarni bilasiz?
2. PHP da satr tipini tushuntirib bering?
3. PHP da massivlar tushuntirib bering?

4-ma’ruza. Funksiyalar. Klasslar va obyektlar. Xatoliklar bilan ishlash.

PHP-funksiyalar

Reja:

4.1. Funksiyalar

4.2. Funksiyalarning argumentlari

4.3. O’zgaruvchan uzunlik argumentlari ro’yxatlari

4.4. Funksiyalar ichida o’zgaruvchilardan foydalanish

4.5. Funksiyaning o’zgaruvchilari

4.6. Ichki joylashgan (ichma-ich) funksiyalar

4.7. Sinflar va obyektlar

4.8. O’zgaruvchilarni inisiallashtirish

4.9. Obyektlar

Kalit so’zlar: funksiyalar, klass va obyektlar, xatoliklar.

4.1. Funksiyalar

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

Natural son faktorialini hisoblash funksiyasi

Misol:

```
<?php
```

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

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

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

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

Agar php-dasturda to'g'ridan-to'g'ri yozilsa, hyech narsani ishlab bo'lmaydi. Birinchidan, funksiya nomi funksiya parametrlari nomlari (1-parametr, 2-parametr va b.) PHP da nomlanish qoidalariga muvofiq kelishi kerak (unda yaxshisi kirillcha sinivollarni ham ishlatmagan ma'qul). Funksiya nomlari registrga nisbatan sezuvchan bo'ladi. Ikkinchidan, funksiya parametrlari – tilning o'zgaruvchan qismlari, shuning uchun ularning har birining nomlari oldidan $ belgisi turishi lozim bo'ladi. Parametrlar ro'yxatida hyech qanday ko'p nuqtalarni qo'yish mumkin emas. Uchinchidan, amallar bloki so'zi bilan birga funksiya

tanasida istalgan to’g’ri PHP-kod mavjud bo’lishi kerak (parametrlarga muvofiq bo’lishi majburiy emas). Va nihoyat, return kalit so’zidan so’ng tartibli php-ifoda kelishi lozim (ma’noga ega bo’lgan qandaydir sinivollar). Bundan tashqari, funksiyada qaytariluvchi ma’no kabi parametrlar bo’lmasligi ham mumkin. Funksiyani to’g’ri e’lon qilishga misol – yuqorida keltirilgan faktorialni hisoblash funksiyasi.

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

Misol:

```
<?php
Funksiya_nomi ("1-parametr_uchun_ma’no", "2-parametr_uchun_ma’no
",...);
/* Funksiyani chaqirishga misol – funksiyani chaqirish faktorialni hisoblash yuqorida bor, 3 soni faktorialini hisoblash uchun u yerda biz fact(3) deb yozganmiz;  u yerda fact – chaqiriluvchi funksiya nomi,  a 3 –$n nomli uning parametri ma’nosi */
?>
```

Funksiyani qachon chaqirish mumkin? Bu g’alati savol bo’lib tuyulishi mumkin. Funksiyani uni aniqlangandan keyin chaqirish mumkin, ya’ni function f_name(){...} blokidan pastda istalgan dastur qatorida. PHP3 da bu aynan shunday. Lekin PHP4 da bunday talab yo’q. Hamma gap interpretator olingan kodni qanday qayta ishlashida. Birgina istisno shartli ravishda aniqlanadigan funksiyadan tashkil topadi (shartli operatorlar Yoki boshqa funksiyalar ichida). Funksiya shu tarzda aniqlangan taqdirda, uni aniqlash uni chaqirishdan oldin bajariladi.

Misol. SHartli funksiya ichida funksiyani aniqlash

```
<?
$make = true;
/* bu yerda Make_event() ni chaqirish mumkin emas; Chunki u hali mavjud emas, lekin Save_info() ni chaqirish mumkin */
```

```
Save_info("Sobir","Sodiqov", "Men PHP kursini tanladim");
if ($make){
// Make_event() funksiyasini aniqlash
function Make_event(){
echo "<p> Python<br> ni o’rganmoqchiman";
}
}
// endi Make_event() ni chaqirish mumkin
Make_event();
// Save_info funksiyasini aniqlanadi
function Save_info($first, $last, $message){
echo "<br>$message<br>";
echo "Ism: ". $first . " ". $last . "<br>";
}
Save_info("Murod","Yoqubov", "Men Lisp ni tanladim");
// Save_info ni bu yerda ham chaqirish mumkin
?>
```

Agar funksiya dastur ichida aniqlangan bo’lsa, uni keyin qayta aniqlash Yoki o’chirib tashlash mumkin emas. Funksiya nomlariga registr ta’sir qilmasligiga qaramasdan, yaxshisi funksiyani aniqlash paytida berilgan nom bilan chaqirish mumkin bo’ladi.

Misol. Funksiya ichidagi funksiyani aniqlash

```
<?php
/* ma’lumotlarni saqlash, ya’ni DataSave() funksiyasini chaqirish mumkin emas. Uning to’g’riligi tekshirilmasdan oldin, ya’ni DataCheck() funksiyasi chaqirilmasdan oldin bu mumkin emas.*/
DataCheck();
DataSave();
function DataCheck(){
// ma’lumotlar to’g’riligini tekshirish
```

```
  function DataSave(){
  // ma'lumotlarni saqlaymiz
} } ?>
```

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

4.2. Funksiyalarning argumentlari

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

Argumentlar yordamida ma'lumotlar funksiyaga uch xil turli usullar bilan o'tkazilishi mumkin. Bu argumentlarni ma'nosiga ko'ra (o'zgarmas holatda foydalaniladi), ilovalarga ko'ra va o'zgarmas holatda argumentlarga ma'no berishga ko'ra o'tkazish . Bu usullarni atroflicha ko'rib chiqamiz.

Argument funksiyaga ma'nosiga ko'ra o'tkazilsa, funksiya ichidagi argument ma'nosining o'zgarishi uning funksiya tashqarisidagi ma'nosiga ta'sir qilmaydi. Funksiyaga uning argumentlarini o'zgartirishga yo'l qo'yish uchun ularni havolalarga ko'ra o'tkazish kerak. Buning uchun argument nomi oldidan funksiyani aniqlashda ampensand "&" belgisini yozish kerak.

Misol. Argumentlarni havolasiga ko'ra o'tkazish

```
<?php
// qo'shimcha qilishi mumkin bo'lgan funksiyani yozamiz checked so'zi qatoriga
function add_label(&$data_str){
  $data_str .= "checked";
}
$str = "<input type=radio name=article ";
  // bunday qator mavjud bo'lsin
echo $str .."><br>";
```

```
// forma elementini keltiradi – belgilanmagan radio knopkasini
add_label($str);
   // funksiyani chaqiramiz
echo $str ."><br>";
   // bu endi belgilangan  radio knopkani keltiradi
?>
```

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

Bizda informasion xabar tuzuvchi funksiya, unga berilgan parametr ma’nosiga muvofiq tarzda o’zgaruvchi imzo bor. Agar parametr ma’nosi berilmagan bo’lsa, “Tashkiliy qo’mita” imzosidan foydalaniladi.

Misol. Tinch holatdagi argument ma’nosi

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

Bu skript ishining natijasi quyidagicha:

Keyingi yig’ilish ertaga bo’lib o’tadi.

Tashkiliy qo’mita.

Keyingi yig’ilish ertaga bo’lib o’tadi.

Hurmat bilan Kamoliddin.

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

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

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

Skript ishi natijasi sifatida quyidagilarni olarniz

Katalogga maqola kiritamiz:

Informatika va biz,

Muallif Murod Yoqubov.

Qisqa tavsif:

Bu maqola informatikaga oid...

Katalogga maqola kiritamiz:

Xarakterlar kim,

Muallif Odil Ziyayev.

Qisqa tavsif:

Bu maqola xarakterlar haqida...

Agar biz quyidagicha yozsak:

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

Natija quyidagicha bo'ladi:

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

4.3. O'zgaruvchan uzunlik argumentlari ro'yxatlari

PHP4 da argumentlarning o'zgaruvchan soni bilan funksiya tuzish mumkin. Ya'ni biz uni necha argumentlar bilan chaqirilishini bilmasdan, funksiya tuzamiz. Bu kabi funksiya yozish uchun hyech qanday maxsus sintaksis kerak bo'lmaydi. Hammasi uning ichiga o'rnatilgan funksiyalar func_num_args(), func_get_arg(), func_get_args() yordami bilan qilinadi.

func_num_args() funksiyasi argumentlar sonini qaytaradi. Bu funksiya faqat foydalanuvchi funksiyasini aniqlash mobaynida foydalanishi mumkin. Agar u funksiyadan tashqarida paydo bo'lsa, interpretator ogohlantirish beradi.

Misol. func_num_args() funksiyasidan foydalanish

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

```
// qatorni keltiradi "3-funksiya argumentlari soni"
?>
```

func_get_arg funksiyasi (argument_raqami to’laligicha) argumentni o’zgarishlar ro’yxatidan argumentlar funksiyasiga qaytaradi, uning tartib raqami func_get_arg parametri bilan beriladi. Funksiya argumentlari noldan boshlab hisoblanadi. **func_num_args**() kabi bu funksiya faqat biron-bir funksiyani aniqlashda foydalaniladi.

Argument raqami funksiyaga o’zgargan argumentlar sonidan ortib ketishi mumkin emas. Aks holda ogohlantirish umumlashtiriladi va func_num_args() funksiyasiga False qiymat qaytadi.

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

Misol. Ma’lumotlar tipini, uning argumentlarini tekshirish

```
<?
function DataCheck(){
  $check =true;
  $n = func_num_args();
/* funksiyaga o’zgargan argumentlar sonini tekshiramiz, birinchi o’zgargan argument butun sonmi-yo’qmi */
  if ($n>=1) if (!is_int(func_get_arg(0)))
      $check = false;
/* tekshiramiz, ikkinchi o’zgargan argument qatormi-yo’qmi */
  if ($n>=2)
     if (!is_string(func_get_arg(1)))
        $check = false;
return $check;
}
if (DataCheck(43,"text"))
   echo "Tekshiruv to’g’ri o’tdi<br>";
```

```
else echo "ma'lumotlar shartlarni qondirmaydi <br>";
if (DataCheck(324))
echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "ma'lumotlar shartlarni qondirmaydi <br>";
?>
```

Dastur natijasi quyidagicha bo'ladi.

Ma'lumotlar shartlarni qoniqtirmaydi. Tekshiruv to'g'ri o'tdi.

func_get_args() funsiyasi argumentlar ro'yxatidan tashkil topgan massiv qaytaradi. Massivning har bir elementi argumentga, funksiya o'zgarishiga to'g'ri keladi. Agar funksiya foydalanuvchi funksiyasi aniqligidan tashqarida foydalanilsa ogohlantirish umumlashtiriladi.

Avvalgi misolni ko'chiramiz, bu funksiyadan foydalanamiz. Funksiyani harakatlantiruvchi juft argument butun son ekanligini tekshiramiz:

Misol:

```
<?
function DataCheck(){
  $check =true;
  $n = func_num_args();
  // funksiyaga o'zgargan argumentlar soni
  $args = func_get_args();
  // funksiya argumentlari massivi
  for ($i=0;$i<$n;$i++){
    $v = $args[$i];
    if ($i % 2 == 0){
      if (!is_int($v)) $check = false;
// tekshiramiz, juft argument butunmi-yo'qmi
}
}
return $check;
}
```

```
if (DataCheck(array("text", 324)))
   echo "Tekshiruv to’g’ri o’tdi<br>";
else echo "Ma’lumotlar shartlarni qoniqtirmaydi <br>";
?>
```

Bundan, func_num_args(), func_get_arg() va func_get_args() funksiya kombinasiyasi funksiyalar o’zgaruvchan argumentlar ro’yxatiga ega bo’la olishi uchun foydalaniladi. Bu funksiyalar faqat PHP4 ga kiritilgan. PHP3 da shunday natijaga erishish uchun, argument sifatida massiv funksiyasidan foydalanish mumkin bo’ladi. **Masalan**, har bir toq funksiyalar parametri butun sonligini tekshiruvchi skriptni quyidagicha yozish mumkin:

```
<?
function DataCheck($params){
  $check =true;
$n = count($params);
// funksiyaga o’zgargan argumentlar soni
  for ($i=0;$i<$n;$i++){
    $v = $params[$i];
    if ($i % 2 !== 0){
// tekshiramiz, toq argument butunmi-yo’qmi
if (!is_int($v)) $check = false;
    }
  }
return $check;
}
if (DataCheck("text", 324))
    echo "Tekshiruv to’g’ri o’tdi<br>";
else echo "Ma’lumotlar shartlarni qoniqtirmaydi<br>";
?>
```

4.4. Funksiyalar ichida o’zgaruvchilardan foydalanish

1. Global o’zgaruvchilar
2. Statistik o’zgaruvchilar
3. Qaytariluvchan ma’nolar
4. Havolani qaytarish

Global o’zgaruvchilar

Funksiyalar ichida undan tashqarida berilgan o’zgaruvchilardan foydalanish uchun, bu o’zgaruvchilarni global deb e’lon qilish kerak. Buning uchun funksiya tanasida uning nomlarini global kalit so’zidan keyin keltirish lozim bo’ladi:

```
global $var1, $var2;
<?
$a=1;
function Test_g(){
global $a;
  $a = $a*2;
  echo ' $a=',$a funksiya ishida natija;
}
echo 'funksiyadan tashqarida $a=',$a,', ';
Test_g();
echo "<br>";
echo funksiyadan tashqarida $a=',$a,', ';
Test_g();
?>
```

misol. Global o’zgaruvchilar

Bu skript ishidan quyidagi natijalarni olarniz:

$a=2 funksiyadan tashqarida, $=2 funksiya ishida natija

$a=2 funksiyadan tashqarida, $=4 funksiya ishida natija

O’zgaruvchi global deb e’lon qilinganda, aniq global o’zgaruvchi uchun havola tuziladi. Buning uchun bunday yozuv quyidagiga ekvivalent (GLOBALS massivi mavjud ko’rinish sohalariga muvofiq barcha global o’zgaruvchilarni o’z

ichiga oladi):

$var1 = & $GLOBALS["var1"];

$var2 = & $GLOBALS["var2"];

Bundan kelib chiqadiki, $var1 o’zgaruvchini o’chirish $_GLOBALS["var1"] global o’zgaruvchiini o’chirib tashlamaydi.

Statistik o’zgaruvchilar

O’zgaruvchilardan faqat funksiya ichida foydalanish uchun bunda uning ma’nosini saqlagan holda va funksiyadan chiqqandan so’ng, bu o’zgaruvchilarni statistik deb e’lon qilish kerak. Statistik o’zgaruvchilar faqat funksiyalar ichida ko’rinadi va dasturni yuklash funksiya doirasidan tashqariga chiqsa o’z ma’nosini yo’qotmaydi. Bu o’zgaruvchilarni e’lon qilish static kalit so’zi yordamida amalga oshiriladi:

static $var1, $var2;

Har qanday ma’no statistik o’zgaruvchi sifatida talqin qilinishi mumkin, faqat havola emas.

Misol. Statistik o’zgaruvchilardan foydalanish

```
<?
function Test_s(){
static $a = 1;
// ifoda Yoki havolani o’zlashtirib bo’lmaydi
  $a = $a*2;
  echo $a;
}
Test_s(); // 2 chiqadi
echo $a;  // hyech narsa chiqmaydi, zero $a faqat funksiya ichida kirish yo’lagi bor
Test_s(); // $a=2 funksiya ichida, shuning uchun funksiya ishi natijasi 4 soni bo’ladi
?>
```

Qaytariluvchan ma’nolar

Yuqorida misol qilib keltirilgan barcha funksiyalar biror-bir amal bajarishgan. Bunday hollardan tashqari, har qanday funksiya o’z ishi natijasi sifatida qandaydir qiymat qaytaradi. Bu return tasdig’i yordamida qilinadi. Qaytariluvchan qiymat har qanday turda, shu jumladan, ro’yxat va obyektlar bo’lishi mumkin. Interpretator funksiya tanasida return komandasiga uchraganda, u darhol uni bajarishni to’xtatadi va funksiya chaqirilgan qatorga o’tib ketadi.

Masalan, inson yoshini qaytaruvchi funksiya tuzamiz. Agar inson vafot etmagan bo’lsa, yosh joriy yilga muvofiq hisoblanadi.

```
<?php
/* agar ikkinchi parametr true kabi hisoblab chiqilsa, u vafot etgan sanaday ko’rib chiqiladi, */
function Age($birth, $is_dead){
  if ($is_dead) return $is_dead-$birth;
  else return date("Y")-$birth;
}
echo Age(1971, false); // vыvedet 33
echo Age(1971, 2001); // vыvedet 30
?>
```

Bu misolda return funksiyasidan foydalanmasa ham bo’ladi, shunchaki uni chiqarish funksiyasini echo ga almashtiriladi. Aksincha, agar biz funksiya biror-bir qiymat qaytaradigan qilsak (bu misolda inson yoshi), biz dasturda o’zgaruvchini bu funksiya qiymatini istalgan o’zgaruvchiga o’zlashtirishimiz mumkin.

```
$my_age = Age(1981, 2004);
```

Funksiya ishi natijasida faqat bitta qiymat qaytarilishi mumkin. Bir necha qiymatni qiymatlar ro’yxati qaytarilgan taqdirda olish mumkin (bir o’lchamli massiv). Biz inson yoshini kunigacha aniqlikda olmoqchimiz, deb hisoblaylik.

```
<?php
function Full_age($b_day, $b_month, $b_year)
{
    $y = date("Y");
```

```
        $m = intval(date("m"));
        $d = intval(date("d"));
        $b_month = intval($b_month);
        $b_day = intval($b_day);
        $b_year = intval($b_year);

        $day = ($b_day > $d ? 30 - $b_day + $d : $d - $b_day);
        $tmpMonth = ($b_day > $d ?  -1 : 0);
        $month = ($b_month > $m + $tmpMonth
         ? $b_month + $tmpMonth - $m : $m+$tmpMonth - $b_month);
        $tmpYear  = ($b_month >  $m + $tmpMonth ?  -1 : 0);
        if ($b_year > $y + $tmpYear)
        {
            $year = 0; $month = 0; $day = 0;
        }
        else
        {
            $year = $y + $tmpYear - $b_year;
        }
        return array ($day,$month,$year);
    }
    $age = Full_age("29","06","1986");
    echo "Siz $age[2] yosh, $age[1] oylar va $age[0] kunlar";
    ?>
```

Funksiya bir necha qiymatlarni ularni dasturda qayta ishlash uchun qaytarganda, bir amal bilan ma’noni birdaniga bir necha o’zgaruvchilarni o’zlashtirishga imkon beruvchi list() til konstruksiyasidan foydalanish qulay bo’ladi. Masalan, yuqoridagi misolda funksiyani, uning qiymatiga o’zgartirish kiritmay qayta ishlash quyidagicha bo’lishi mumkin:

```
<?
```

```
// Full_age() funksiya kiritish
list($day,$month,$year) = Full_age("07",
    "08","1974");
echo "Sizning yoshingiz $year, $month oy va
    $day kun";
?>
```

list() konstruksiyasini umuman o’zgaruvchini o’zlashtirish uchun istalgan massiv elementi qiymatidan foydalanish mumkin.

Misol. list() dan foydalanish

```
<?
$arr = array("first","second");
list($a,$b) = $arr;
   // o’zgaruvchi $a o’zlashtiriladi, birinchi massiv qiymati, $b – ikkinchi
echo $a," ",$b;
// «first second» qatori keltiriladi
?>
```

Havolani qaytarish

Funksiya o’z ishi natijasida shuningdek havolani biror-bir o’zgaruvchiga qaytarishi mumkin. Bu funksiyani qanday o’zgaruvchi havolaga o’zlashtirish kerakligini aniqlash uchun foydalaniladi. Funksiyadan havola olish uchun, e’lon oldidan ampersand (&) belgisini yozish kerak bo’ladi va har safar funksiya chaqiruvi paytida uning nomi oldidan ham ampersand (&) yozish kerak bo’ladi. Ko’pincha funksiya havolani biror-bir global o’zgaruvchiga (Yoki uning qismini – havolani global massiv elementiga), havolani statistik o’zgaruvchiga (Yoki uning qismini) Yoki havolani argumentlardan biriga qaytaradi, agar u havola bo’yicha berilgan bo’lsa.

Misol. Havolani qaytarish

```
<?
$a = 3; $b = 2;
function & ref($par){
```

```
global $a, $b;
  if ($par % 2 == 0) return $b;
  else return $a;
}
$var =& ref(4);
echo $var, " i ", $b, "<br>"; // 2 va 2 keltiriladi
$b = 10;
echo $var, " i ", $b, "<br>";  // 10 va10 keltiriladi
?>
```

Havola sintaksisidan foydalanishda bizning misoldagi $var o'zgaruvchi o'zgaruvchining $b qiymati $ref qaytarilgan funksiyasiga ko'chirilmaydi, bu o'zgaruvchiga havola tuziladi. Demak, endi $var va $b teng kuchli o'zgaruvchilar va ular bir paytda o'zgartiriladi.

4.5. Funksiyaning o'zgaruvchilari

PHP funksiyalar o'zgaruvchilariga ko'maklashadi. Bu degani, agar o'zgaruvchi nomi oddiy qavslar bilan tugasa, PHP shu kabi nomli funksiyani qidiradi va uni bajarishga harakat qiladi.

Misol. Funksiyalar o'zgaruvchilaridan foydalanish

```
<?
/* Ikkita oddiy funksiya tuzamiz: Add_sign – qatorga imzo qo'shadi va
SHow_text –matn qatorini chiqarib beradi*/
function Add_sign($string,
   $sign="Hurmat bilan, Murod"){
   echo $string ." ".$sign;
}
function SHow_text(){
  echo "Xabarni pochtadan jo'natish<br>";
}
$func = "SHow_text"; // ma'noga ega o'zgaruvchi tuzamiz, u funksiya
```

nomiga teng SHow_text

$func(); // u SHow_text funksiyani chaqiradi

$func = "Add_sign"; // ma’noga ega o’zgaruvchi tuzamiz, u funksiya nomiga teng Add_sign

$func("Hammaga salom
");

// bu funksiyani chaqiradi Add_sign "Hammaga salom" parametrli

?>

Bu misolda SHow text funksiya shunchaki matn qatorini chiqaradi. Agar echo maxsus funksiyasi mavjud bo’lsa, nega buning uchun alohida funksiya tuzish kerak, deyish mumkin. Gap shundaki, echo(), print(), unset(), include() kabi funksiyalardan funksiyalar o’zgaruvchilari sifatida foydalanib bo’lmaydi. Ya’ni biz yozsak:

<?

$func = "echo ";

$func("TEXT");

?>

Interpretator xatoni ko’rsatadi:

Fatal error: Call to undefined function:

echo() in c:\users\nina\tasks\func\var_f.php on line 2

SHuning uchun yuqorida keltirib o’tilgan istalgan funksiyalardan o’zgaruvchilar funksiyasi sifatida foydalanish uchun yuqoridagi misoldagi yo’lni tutdik.

4.6. Ichki joylashgan (ichma-ich) funksiyalar

Foydalanuvchi tomonidan aniqlanadigan funksiyalar haqida gapirganda ichki joylashgan funksiyalar haqida gap ketmasligi mumkin emas. Yuqorida biz echo(), print(), date(), include() kabi ichki joylashgan funksiyalar bilan tanishdik. Bundan tashqari date() funksiyadan boshqa barcha funksiyalar PHP dasturlash tili konstruksiyasiga ega. Ular PHP dasturlash tili yadrosiga joylashgan bo’lib, hyech qanday modullar va qo’shimcha o’zgartirishlar talab etmaydi. Ammo shunday

funksiyalar mavjudki, ularga turli fayl bibliotekalari va mos ravishda modullarni yuklamasdan iloji yo'q. Masalan, MySql ma'lumotlar bazasi bilan ishlaydigan funksiyalardan foydalanish uchun shunday kengaytmali fayllarni qo'llab quvvatlaydigan komponentalari kerak. Oxirgi vaqtlarda bu funksiyalardan foydalanish uchun qo'shimcha komponentalar kerak emas, chunki ularning barchasi hozirda PHP dasturlash tili yadrosiga kiritilgan.

4.7. Sinflar va obyektlar

Obyektga yo'naltirilgan dasturlashning asosiy tushunchalari – sinflar hamada obyektlardir. Bu tushunchalarni quyidagicha tushunish mumkin: obyekt – bu dasturda qo'llaniladigan tushuncha Yoki biror fizik predmet haqida ma'lumot beradigan strukturalangan o'zgaruvchidir, sinflar esa bu obyektlarning tavsifi va ular ustida bajariladigan harakatlardir.

PHP dasturlash tilida sinflar quyidagi sintaksis yordamida aniqlanadi:

```
class Sinf_nomi{
  var $xususiyat_nomi;
    /* xususiyatlar ro'yxati */
  function metod_nomi( ){
   /* usullarning tanasi */
  }
  /*usullar ro'yxati*/
}
```

Sinf obyektlari xususiyatlari nomi var kalit so'zi yordamida e'lon qilinadi, berilgan sinf obyektlariga qo'llanilgan usullar funksiya sifatida ishlatiladi. Sinf tanasi ichida this kalit so'zi yordamida taqdim qilinayotgan joriy sinfga murojaatni amalga oshirish mumkin.

Masalan, biz maqola kategoriyasini tasvirlovchi sinf tuzishimiz kerak. Har bir maqolaning nomi, muallifi va qisqa mazmuni kabi xususiyatlari bor. Biz maqola bilan qanday amal bajarmoqchimiz? Biz sanab o'tilgan xususiyatlarga ma'no berishimiz, maqolani brauzerda ko'rsatishimiz kerak bo'ladi. SHunda bu

sinfning ifodalanishi quyidagicha holatda bo’ladi:

```
<?
class Articles { // Maqola sinfini tuzamiz
  var $title;
  var $author;
  var $description;
// maqola atributi ma’nosini o’zlashtiruvchi usul
  function make_article($t, $a, $d){
    $this->title = $t;
    $this->author = $a;
    $this->description = $d;
  }
//sinf  nusxasini  ifodalash  uchun usul
  function show_article(){
    $art = $this->title . "<br>" .
          $this->description .
      "<br>Muallif: " . $this->author;
    echo $art;
} } ?>
```

SHunday qilib “maqola” turidagi fizik obyektlarni tasvirlash uchun biz uch o’zgartuvchidan tashkil topgan, maqola xarakteristikasini o’zida jamlagan Articles nomli sinf va muayyan maqola tuzish va uni tasvirlash uchun ikkita funksiya tuzdik.

Ma’lumki, PHP bilan ishlash davriy holatda HTML rejimida yuklanishi mumkin. Bu holda dastur bir necha kodning bo’laklari(bloklar)dan tashkil topadi. Sinfni ifodalash php-kodning har xil bloklari bo’yicha va qolaversa har xil fayllar bo’yicha tarqatilmasligi kerak. Ya’ni quyidagicha yozsak:

```
<?php
class Articles { // Sinfni tasvirlashning boshi
  var $title;
```

```
?>
<?php   // sinfni tasvirlashning davomi
   function show_article(){  // usulning tarkibi
   }
} // sinfni tasvirlashning yakuni
?>
```

bunda dastur tartibli ishlaydi.

Sinf nomi masalasida ayrim narsalarni e’tiborda tutish kerak. Sinfning nomi PHP tilidagi obyektlar nomlanishi qoidalariga javob berishi lozim, lekin bir qator nomlar borki, texnik mutaxassislar tomonidan o’z maqsadi uchun zaxira qilinadi. Birinchi navbatda bu nomlar “_” quyi chiziqdan boshlanuvchilardir. Sinflar va funksiyalar tuzish uchun bu kabi nomlarni ishlatmaslik kerak. Bundan tashqari stdClass nomi zaxira qilingan, zero u PHP surilgichi ichida ishlatiladi.

4.8. O’zgaruvchilarni inisiallashtirish

Ba’zan ayrim sinf atributlariga ma’noni sinf ishtirokchisini tuzish bilanoq o’zlashtirish kerak bo’ladi. Biz maqola sinfini tuzganimizda, sinf atributlari (xususiyatlari) ma’nolarini o’zlashtirish uchun maxsus funksiya make_article() dan foydalandik. Umuman olganda, biz to’g’ri yo’l tutmadik, chunki “velosiped ixtirosi” bilan shug’ullandik. Sinf atributlarining boshlang’ich ma’nolarini berish uchun maxsus ikkita standart usul mavjud. PHP4da ma’noni var operatori Yoki konstruktor funksiyasi yordamida inisiallashtirish mumkin. var yordamida faqat konstant ma’nolarni inisiallashtirish mumkin. Konstant bo’lmagan ma’nolarni berish uchun obyekt sinfdan ajrab chiqqanda o’z-o’zidan ishga tushuvchi konstruktor funsiyasidan foydalaniladi. Konstruktor-funksiya u ifodalangan butun sinfga mos keluvchi nomga ega bo’lishi kerak.

Misol. “maqola” deb nomlangan obyektni tuzishda uning xususiyatlarini quyidagicha belgilash mumkin: mualliflar – “Kamolov” satriga teng, nomlanish va qisqa mazmun - $_POST global massivi elementlariga mos, maqola nashri –

mazkur sanada.

```
<?
class Articles { // maqola sinfini tuzish
  var $title;
  var $author = "Kamolov";
  var $description;
  var $published;       // sinf atributi ma’nosini o’zlashtiruvchi usul
 function Articles(){
  $this->title = $_POST["title"];
  $this->description = $_POST["description"];
  $this ->published = date("Y-m-d");
}   }   ?>
```

PHP3 va PHP4 da konstruktorlar har xil ishlashini hisobga olish kerak. Funksiya PHP3 da, agar u sinfniki kabi nomga ega bo’lsa, konsrtuktorga aylanadi, PHP4 da esa – agar u ifodalangan sinfniki kabi nomga ega bo’lsa shunday bo’ladi. Bir sinf boshqasini kengaytirganda va xususiyatlarning va baza sinflar usullarining ergashishida usullar orasidagi farq ko’rinib turibdi. Lekin biz bu haqda biroz keyinroq gapiramiz. PHP5da sinf konstruktori _construct deb nomlanadi. Bundan tashqari, PHPda destruktorlar – obyektni yo’q qilishda o’z-o’zidan ishga tushuvchi funksiyalar paydo bo’ldi. PHP5 da funksiya-destruktor destruct deb nomlanishi kerak bo’ladi.

4.9. Obyektlar

Object (obyektlar) tipi

Obyektlar – obyektga yo’naltirilgan dasturlashdan kirib kelgan *ma’lumot tipi*dir. Obyektga yo’naltirilgan dasturlash tamoyiliga ko’ra, sinf – aniq xossalarga ega va ular bilan ishlaydigan metodli obyektlar to’plarni. Obyekt esa mos ravishda sinf nusxasidir. Masalan, dasturchilar – bu dasturni tuzuvchi, kompyuter adabiyotlarini o’rganadigan odamlar sinfi va bundan tashqari hamma odamlar qatori ism va familiyasi mavjud. Endi agarda biror aniq dasturchi – Azamat

Boboyevni olsak, u holda uni shu xossaga ega bo'lgan dasturchi sinfini obyekti sifatida qarash mumkin va u ham dastur tuzadi, hamda ismi mavjud va boshqalar.

PHP dasturlash tilida *obyekt metodi*ga murojaat -> *amali*dan foydalaniladi. Obyektni inisializasiya qilishda *obyektni o'zgaruvchan nusxasi*ni yaratadigan new ifodasidan foydalaniladi.

```
<?php     // odam sinfini yaratamiz.
class Person
{       // PHP dasturlash tilini o'rganadigan odam metodi
    function know_php()  {
            echo "Endi men PHP dasturlash tilini bilaman!";
        }
}
$bob = new Person; // odam sinfini obyektini yaratamiz.
$bob -> know_php(); // uni PHP tiliga o'rgatamiz.
?>
```

Sinf – bu obyekt tipidagi ma'lumotlarning bir turidagi ifodalanishidir. Sinflar real o'zgaruvchilar uchun shablon vazifasini o'taydi. Kerakli tipdagi o'zgartuvchi new operatori yordamida sinfdan tuziladi. Obyektni tuzib, biz barcha usullarni qo'llashiniiz va barcha sinf ifodasida ko'rsatib o'tilgan xususiyatlarni olishimiz mumkin bo'ladi. Buning uchun quyidagicha sintaksisdan foydalaniladi: $obyekt_nomi->xususiyat Yoki _nomi$obyekt_nomi->usulning_nomlanishi(argumentlar ro'yxati). Xususiyatlar Yoki usular nomlari oldidan $ belgisi qo'yilmaydi.

Misol: Obyekt usullari va xususiyatlariga erkin kirish (dostup)

```
<?php
$art = new Articles;     // obyekt tuzamiz $art
echo ($art ->title);     // obyekga nomlanish beramiz $art
$another_art = new Articles;     // obyekt tuzamiz $another_art
$another_art->show_article();     // obyektning brauzerdagi ifodasi uchun usulni chaqiramiz
```

?>

Sinfning har bir obyekti aynan bir xil xususiyatlar va usullarga ega bo'ladi. Demak, $art obyektda va $another_art obyektda title, description, author xususiyatlari va Articles(), show_article() usullari mavjud. Lekin bular ikki xil obyektlar. Obyektni fayllar sistemasidagi direktoriya deb hisoblaymiz, uning xarakteristikasi esa – bu direktoriyadagi fayllar singari bo'lsin. Aniqki, har bir direktoriyada bir xil fayllar yotishi mumkin, lekin shunday bo'lsa-da, ular har xil direktoriyalarda saqlanayotgani uchun har xil hisoblanishi mumkin. Xuddi shuningdek, xususiyatlar va usullar ham, agar ular turli obyektlarga qo'llaniladigan bo'lsa, har xil hisoblanadi. Yuqori bosqichdagi direktoriyadan kerakli faylni olish uchun bu faylga yo'lni batafsil yozib chiqamiz. Sinflar bilan ishlash mobaynida biz chaqirishni istagan funksiyaning nomini to'liq yozishimiz kerak bo'ladi. PHP dagi yuqori bosqich direktoriyalariga global o'zgaruvchilarning bo'sh o'rni bo'ladi, yo'l esa -> taqsinilovchisi yordamida ko'rsatiladi. SHu tarzda $art->title va $another_art->title nomlari ikki xil turli o'zgaruvchilarni angiatadi. PHP da o'zgaruvchi nom oldidan faqat bitta dollar belgisiga ega bo'ladi, shuning uchun $art->$title ko'rinishida yozish mumkin emas. Bu konstruksiya $art obyektining title xususiyatiga murojaat sifatida ko'rib chiqilmaydi, $title o'zgartuvchi ko'rinishida berilgan nomli xususiyat sifatida ko'riladi (masalan, $art->"").

Misol: Xususiyatlar ma'nosini o'rnatish

```
<?php
$art->title = " Internet ga kirish"; // obyekt xususiyati ma'nosini shunday o'rnatish mumkin
$art->$title = "Internet ga kirish"; // obyekt xususiyati ma'nosini bunday o'rnatib bo'lmaydi
$property = "title";
$art->$property = "Internet ga kirish"; // obyekt xususiyati ma'nosini shunday o'rnatish mumkin
?>
```

Sinfni tuzib, bu sinfning obyekti qanday nomga ega bo'lishini bila

olmaymiz, qolaversa obyektlar juda ko’p bo’lishi va ularning barchasi har xil nomga ega bo’lishi mumkin. Sinfni yuzaga chiqarish ichida obyektga qanday munosabatda bo’lishni bilmaymiz. Sinf yuzaga chiqishi ichida funksiyalar va o’zgaruvchilarga erkin kirish uchun, $thid o’rindosh o’zgaruvchisidan foydalanish kerak. Masalan, $this->title shunday sinf obyektining title ini qaytaradi. Ba’zan bu o’zgaruvchini “mening xususiy mulkim” (xususiyatga munosabat tariqasida) deb o’qishni taklif qilinadi.

Takrorlash uchun savollar:

1. PHP da funksiyalarni tushuntirib bering.
2. PHP da funksiyalar argumentlari qanday e’lon qilinadi?
3. PHP da o’zgaruvchilar e’lonini tushuntiring.
4. Ichma-ich funksiyalarni nima?
5. PHP da sinflar va obyektlar nima?

5-ma'ruza. PHPda ma'lumotlar bazalari bilan ishlash. MySQL ma'lumotlar bazasi

Reja:

5.1. Ma'lumotlar bazasi haqida tushuncha

5.2. Ma'lumotlar bazasi interfeysi

5.3. Ma'lumotlar bazasi bilan bog'lanish

5.4. Ma'lumotlar bazasi ustida amallar

Kalit so'zlar:ma'lumotlar bazasi, MySql dasturi, PHP da MySql bilan bog'lanish

5.1. Ma'lumotlar bazasi haqida tushuncha

Ushbu bo'lim PHP va MySql MBBS o'rtasidagi hamkorlik usullari bilan tanishishga mo'ljallangan. Asosiy e'tibor ma'lumotlar bazasi bilan bog'lanishni o'rnatish, so'rovlar jo'natish funksiyalari va javoblarni (mysql_connect, mysql_query, mysql_result, mysql_num_rows, mysql_close) qayta ishlashga qaratiladi. Misol sifatida virtual tarix muzeyi ma'muriyati uchun web-interfeys tuzish masalasini ko'raylik. PHP distributivida MySQL ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan funksiyalar mavjud. Bunda bu funksiyalarning MySql dagi ba'zi bir ma'lumotlar bazasini tasvirlash va to'ldirish maqsadida web-interfeyslarni tuzish imkonini beruvchi funksiyalar bilan tanishamiz. Ma'lumotlar bazasiga ma'lumotlarni qo'shish uchun web-interfeys bilan ishlashda bu ma'lumotlarni shunchaki html-formaga kiritish va ularni serverga jo'natish kerak bo'ladi.

Namoyish etishda bu interfeysni virtual muzey eksponatlari haqidagi ma'lumotlar saqlanadigan Artifacts jadvallari uchun tuzamiz. Artifacts kolleksiyasidagi har bir eksponat quyidagi xarakteristika yordamida tasvirlanishini eslatib o'tamiz:

nom (title);

muallif (author);

ifoda (description);

o’rindosh nom (alternative);

tasvir (photo).

Nomlanish va o’rindosh nomlanish uzunasiga 255 belgidan kam satr (ya’ni VARCHAR(255)), tasvirlash – matnli maydon (TEXT turiga mansub) hisoblanadi, “muallif” va “tasvir” maydonlarida esa Persons kolleksiyasidan muallifning identifikatorlari va Images kolleksiyasidan eksponat tasvirlariga muvofiq mavjud bo’ladi.

5.2. Ma’lumotlar bazasi interfeysi

Ma’lumotlar bazasidagi mavjud jadval strukturasini (ya’ni uning maydonlari jamlanmasini) html-formada tasvirlash uchun quyidagi tarkibiy topshiriqlarni rejalashtirish mumkin:

- MB bilan ulanishni o’rnatish;
- MB ishini tanlash;
- Jadval maydonlari ro’yxatini olish;
- html-formada maydonlarni tasvirlash.

Bundan keyin formaga kiritilgan ma’lumotlarni ma’lumotlar bazasiga kiritish mumkin.

5.3. Ma’lumotlar bazasi bilan bog’lanish (MySQL dasturi misolida)

Aloqa o’rnatish

Ma’lumotlar bazasi bilan aloqa o’rnatish uchun **mysql_connect** funksiyasidan foydalaniladi.

mysql_connect sintaksisi

mysql_connect resursi (“server qatori”, “username”, “password”)

Bu funksiya MySQL serveri bilan aloqa o’rnatadi va bu aloqaga ko’rsatkich qaytaradi Yoki muvaffaqiyatsiz chiqqanda FALSE ko’rsatadi. Odatda quyidagi parametrlar qiymati e’lon qilinadi:

server = 'localhost:3306'

username = server jarayoni egasidan foydalanuvchi ismi

password = bo’sh parol

Server bilan ulanish, agar u bungacha mysql_close() yordamida yopilmagan bo’lsa, skriptni amalga oshirish tugallanishida baza bilan aloqa yopiladi.

Misol:

<?

$conn = mysql_connect("localhost", "nina","123") or die("Ulanishni amalga oshirib bo’lmaydi: ". mysql_error());

echo "Ulanish amalga oshdi";

mysql_close($conn);

?>

mysql_connect amali

shell>mysql -u nina -p123 buyrug’i bilan teng kuchli.

5.4. Ma’lumotlar bazasi ustida amallar

Ma’lumotlar bazalarini tanlash

MySQL da ma’lumotlar bazasini tanlash use buyrug’i yordamida amalga oshiriladi:

mysql>use book;

PHP da buning uchun mysql_select_db funksiyasi mavjud.

mysql_select_db: sintaksisi

mantiqiy mysql_select_db (database_name qatori);

Bu funksiya TRUE qiymatni ma’lumotlar bazasini muvaffaqiyatli tanlanganda qaytaradi va FALSE ni esa – aksincha bo’lganda.

Misol: Book ma’lumotlar bazasini tanlash

<?

$conn = mysql_connect("localhost","user","123") or die("Ulanishni amalga oshirib bo’lmaydi: ". mysql_error());

echo "Ulanish amalga oshdi";

mysql_select_db("book");

?>

Jadval maydonlari ro'yxatini olish

PHP da ma'lumotlar bazasi bilan bog'lanilgandan so'ng, undagi jadvallar ro'yxatini olish mumkin. Bu funksiya - mysql_list_fields.

mysql_list_fields sintaksisi

mysql_list_fields (database_name qatori, table_name qatori)

mysql_field_name funksiyasi so'rov amalga oshirilishi natijasida olingan maydon nomini qaytaradi. mysql_field_len funksiyasi maydon uzunligini qaytaradi. mysql_field_type funksiyasi maydon tipini qaytaradi, mysql_field_flags funksiyasi esa probel bilan yozilgan maydon bayroqlari ro'yxatini qaytaradi. Maydon tiplari int, real, string, blob va h. bo'lishi mumkin. Bayroqlar not_null, primary_key, unique_key, blob, auto_increment va h. bo'lishi mumkin.

Bu barcha buyruqlar sintaksisi bir xil:

mysql_field_name (result qatori, butun field_offset) resursi;

mysql_field_type (result qatori, butun field_offset) resursi;

mysql_field_flags (result qatori, butun field_offset) resursi;

mysql_field_len (result qatori, butun field_offset)

Bu yerda result – bu so'rov natijasi identifikatori (masalan, mysql_list_fields Yoki mysql_query funksiyalar bilan jo'natilgan so'rov), field_offset esa – natijadagi maydonning tartib raqami.

mysql_num_rows(result resursi) buyrug'i result ning ko'pgina natijalari qatorlari miqdorini qaytaradi.

Misol: Artifacts (eksponatlar kolleksiyasi) jadvali maydonlari ro'yxatini olish.

```
<?
$conn = mysql_connect("localhost","user","123") or die("Aloqa o'rnatib bo'lmaydi: ". mysql_error());
echo "Aloqa o'rnatildi";
mysql_select_db("book");
$list_f = mysql_list_fields ("book","Artifacts",$conn);
```

```
$n = mysql_num_fields($list_f);
for($i=0;$i<$n; $i++){
  $type = mysql_field_type($list_f, $i);
  $name_f = mysql_field_name($list_f,$i);
  $len = mysql_field_len($list_f, $i);
  $flags_str =  mysql_field_flags ($list_f, $i);
echo "<br>Maydon nomi: ". $name_f;
echo "<br>Maydon turi: ". $type;
echo "<br>Maydon uzunligi: ". $len;
echo "<br>Maydon bayroqlari qatori: " . $flags_str . "<hr>";
}
?>
```

Natija sifatida taxminan quyidagilarni olish mumkin (albatta, jadvalda ikkita maydon bo’lganda):

- Maydon nomi: id
- Maydon turi: int
- Maydon uzunligi: 11
- Maydon bayroqlari qatori:
- not_null primary_key auto_increment
- Maydon nomi: title
- Maydon turi: string
- Maydon uzunligi: 255
- Maydon bayroqlari qatori:

html-formada maydonlar ro’yxatining aks etishi

Maydon haqidagi ma’lumotni html-forma elementida aks ettiramiz. BLOB turidagi elementlarni textarea ga o’tkazamiz (TEXT turida biz tuzgan description maydoni BLOB tipiga ega), raqamlar va qatorlarni <input type=text> matnli qatorlarida aks ettiramiz, avtoinkrement belgisiga ega elementni esa umuman aks ettirmaymiz, chunki uning ma’nosi o’z-o’zidan o’rnatiladi.

Buning uchun explode funksiyasidan foydalaniladi:

explode: sintaksisi

explode massivi (separator qatori, string qatori , int limit)

Bu funksiya string qatorini separator taqsinilagichi yordamida qismlarga bo’ladi va olingan qatorlar massivini qaytaradi.

Bizning holatda taqsinilagich sifatida probel “ ” ni olish kera, bo’lish uchun boshlang’ich qator sifatida esa – maydon bayroqlari qatorini.

Misol. Artifacts jadvaliga ma’lumot kiritish uchun forma (index.php)

```
<?php
  $dblocation = "localhost";
  $dbname = "book";
  $dbuser = "root";
  $dbpasswd = "";
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
@mysql_select_db($dbname,$dbcnx);
if(isset($_POST['save_hide'])) {
$title=($_POST['title']);
$query = "INSERT INTO Artifacts (title) VALUES ('$title')";
 if(@mysql_query($query))
  {
   echo "<HTML><HEAD><META HTTP-EQUIV='Refresh' CONTENT='0; URL=index.php'></HEAD></HTML>";
  } else {   print mysql_error();   error("Ma’lumotni bazaga yezishda xatolik");}
} else {
?>
<TABLE width="100%" align=center border=0><TR><TD>
<form method="POST" name="save" action="" enctype="multipart/form-data" >
<input type=hidden name=save_hide value=save_hide>
```

```
<TABLE width="100%" align=center border=0><TR>
<TD>Eksponat nomini kiriting </TD>
<TD><INPUT class="input" size="35" name="title" value=""></TD>
</TR></TABLE>
<input type="submit" value="Saqlash" class="batton" />
</form></TD></TR></TABLE>
<? } ?>
```

Ma’lumotlar bazasiga ma’lumotlar yozish

Ma’lumki, ma’lumotlarni jadvalga yozish uchun SQL tilidagi INSERT buyrug’i ishlatiladi:

```
mysql> INSERT INTO Artifacts SET title='Eksponat nomi';
```

PHP skriptda bunday buyruqdan foydalanish uchun mysql_query() funksiyasi mavjud.

mysql_query sintaksisi

mysql_query resursi (query qatori)

mysql_query() SQL-so’rovni MySQL ma’lumotlar bazasining ma’lumotlar bazasiga jo’natadi. Agar ochiq aloqa bo’lmasa, funksiya parametrsiz mysql_connect() funksiyasiga o’xshash holda MBBT bilan bog’lanishga urinadi.

So’rov natijasi buferlanadi.

Takrorlash uchun savollar:

1. Ma’lumotlar bazasi deganda nimani tushunasiz?
2. Qanaka ma’lumotlar bazasi dasturlarini bilasiz va ularni imkoniyatlari?
3. PHP da MySql bilan bog’lanish funksiyasini tushuntirib bering?
4. PHP da MySql so’rovlarini amalga oshirishga misol keltiring?

2.2. AMALIY MASHG’ULOT MATERIALLARI

1-Amaliy mashg’ulot

Mavzu: PHP sintaksisi

PHPni HTML tarkibiga qo'shish

PHP fayllari odatela .php kengaytmasiga ega bo'ladi. Web server bu kengaytmali faylga murojaat qilganida avtomat ravishda fayl PHP protsessorga beradi. Web serverni htm Yoki html fayllarini ham PHP protsessorga yuboradigan qilib sozlash mumkin.

PHP Parsemi chaqirish quyidagicha bo'ladi:

```
<?php

    echo 'Assalomu alaykum bo`lajak dasturchilar!<br>';

    echo "Bir nechta Satrni
            ham yozish mumkin. <br>";
?>
```

echo funksiyasi ma'lumotlarni chop qilish uchun ishatiladi.

ECHO FUNKSIYASINING YANA BIR KO’RINISHI

echo funksiyasi orqali bir nechta satrli HTML kodlarini php orqali chop qilish mumkin. Bunda yopuvchi END identifikatori alohida satrda yozilishi shart va hech qanday probel ham bo'lishi mumkin emas. Aks holda xatolik sodir bo'ladi.

```
<?php
echo <<<END
<!DOCTYPE HTML>
<html>
  <head>
        <meta charset="utf-8">
        <title>She'r</title>
        </style>
  </head>
       <body>
<p>        Vafosizda hayo yo'q, hayosizda vafo yo'q, <br>
                  Harkidaki bu ikki yo'q, iymom yo'q.     <br>
                  Andin yaxshilik kutib bo'lmas,          <br>
                   Chunkim, imkom yo'q.                <br>
                  (Alisher Navoiy)
</p>
       </body>
```

```
    </html>
END
?>
```

Ba'zi hollarda ochuvchi <?php teg o’rinda <? yozilganini ko'ramiz. Lekin bunday qilish tavsiya etilmaydi. Chunki, Web serverni qo'shimcha sozlash lozim bo'ladi.

HTML fayli tarkbida PHP dan foydalanish:

```
<!DOCTYPE html>
<html>
<head>
    <title>Sarlavha</title>
</head>
<body>
    <h1>Bu HTMLda</h1>
    <?php
        echo "Assalomu alaykum bo`lajak dasturchi!<br>";
        // date - bugungi sanani olish funksiyasi
        echo "Bugun ".date('d/m/y').'<br>';
        echo '(c) 2009 - '.date('Y');
     ?>
</body>
</html>
<?php
    echo phpinfo();
?>
```

Izohlarni yozish

Izohlar

Programmaning ma'lum qismini tavsiflash uchun ishlatiladi va bu qatorda hech qanday amai bajarilmaydi, ya’ni programmaning biror qismini yaxshiroq tushuntirish uchun xizmat qiladi. Izoh "/*" va "*/" sinivollari orasida beriladi. /* Bu yerga izoh yoziladi. */

Bundan tashqari bir satrli izohlardan ham foydalanish mumkin. Buning uchun izoh boshiga "//" belgisi qo'yiladi.

Ko'p satrli izoh yozishga misol

```
<?php
    /*
        Shu shaklda bir nechta
        satrga izoh yozish
        mumkin */
?>
```

Nuqtali vergul (;)

PHPda nuqtali vergul xuddi C++, C# va Java dasturlash tillari kabi operator tugallanganligini bildirish uchun ishlatiladi. Ko'p xollarda shu belgi tushirib qoldirilgani uchun "Parse error" sintaksis xatoligi chiqariladi.

$ belgisi

PHP hamma o'zgaruvchilar oldidan $ belgisi yozilishi shart. **O'zgaruvchilar oldidan $ belgisi yoziladi**

```
<?php
  $butun_son = 1;
  $satr      = "Hello";
  $massiv    = array("One", "Two", "Three");
?>
```

O'zgaruvchi :

Xotiraning nomlangan qismi bolib, o'zida malum bir qiymatni saqlaydi. O'zgaruvchining nomi va qiymati bo'ladi. O'zgaruvchining nomi orqali qiymat saqlanayotgan xotira qismiga murojaat qilinadi. Programma ishlashi jarayonida o'zgaruvchining qiymatini o'zgartirish mumkin.

O'zgaruvchilarga nom berish

O'zgaruvchilarga nom tanlaganda quyidagilarga ahamiyat berish kerak:

1. O'zgaruvchilarga nom lotin harflaridan Yoki ost chiziqdan (_) boshlanishi shart;

2. Ikkinchi sinivoldan boshlab raqamlardan foydalanish mumkin;

3. PHP da katta kichik harflar farq qiladi. Ya’ni quyidagilarning har biri alohida o'zgaruvchi hisoblanadi: KATTA, katta, KaTTa, kAttA, Katta, KattA, ...

4. Probeldan o'zgaruvchi nomida foydalanib bo'lmaydi. Probel o’rniga ost chiziqdan foydalanishingiz mumkin. Masalan, $talaba_ismi;

5. O'zgaruvchi nomida faqat katta va kichik lotin harflari, raqamlar va ost chiziqni ishlatish mumkin.

6. O'zgaruvchi nomi oldidan $ belgisi bo'lishi shart

Operator

Tilning yakunlangan jumlasi hisoblanadi va ma'lumot taxlilining tugallangan bosqichini ifodalaydi. Operatorlar nuqtali vergul ";" bilan ajratiladi. Ya’ni ";" operatorning tugallanganligini bildiradi.

Amallar	Izoh	Namuna
+	Qo'shish	$k + 1
-	Ayirish	$k - 1
*	Ko'paytirish	$k * 10
/	Bo'lish	$k / 10
%	Qoldiqni olish	$k % 10
++	Inkrement. O'zgaruvchi qiymatini birga oshirish	$k++
--	Dekriment. O'zgaruvchi qiymatini birga kamaytirish	$k--

Amallarga quyidagicha misollarni namuna qilib keltirish mumkin.

Operator	Namuna	Ekvivalent
+=	$k += 10;	$k = $k + 10;
-=	$k -= 10;	$k = $k - 10;
*=	$k *= 10;	$k = $k * 10;
/=	$k /= 10;	$k = $k / 10;
%=	$k %= 10;	$k = $k % 10;
++	$k++;	$k = $k + 1;
--	$k--;	$k = $k - 1;
.=	$k.=$c;	$k = $k.$c;

Apostrof(`) orasidagi o'zgaruvchilar o’zgarishsiz chiqariladi

```
<?php
        $a = 5;
        $b = 3;

        $c = $a + $b;

        echo '$a = '.$a.'<br>';
        echo '$b = '.$b.'<br>';
        echo '$a + $b = '.$c;
?>
```

Mustaqil ishlash uchun masalalar:

1. Kvadratning tomoni a berilgan. Uning perimetri aniqlansin. $P = 4 * a$.
2. Kvadratning tomoni a berilgan. Uning yuzasi aniqlansin. $S = a^2$
3. To'g'ri to'rtburchakning tomonlari a va b berilgan. Uning yuzasi S = a*b; va P = 2 * (a + b) perimetri aniqlansin.
4. Aylananing diametri d berilgan. Uning uzunligi aniqlansin $L = 2 * pi * r$ pi= 3.14
5. Ikki xonali son berilgan. Oldin uning o'nliklar xonasidagi raqamini, so'ng birlar xonasidagi raqamini chiqaruvchi programma tuzilsin.
6. Ikki xonali son berilgan. Uning raqamlari yig’indisini aniqlovchi programma tuzilsin.
7. Ikki xonali son berilgan. Uning raqamlari o’rnini almashtirishdan hosil bo'lgan sonni aniqlovchi programma tuzilsin.

8. Uch xonali son berilgan. Uning yuzlar xonasidagi raqamini aniqlovchi programma tuzilsin.
9. Uch xonali son berilgan. Oldin um birliklar xonasidagi raqamini so'ng o'nliklar xonasidagi raqamini chiqaruvchi programma tuzilsin.
10. Uch xonali son berilgan. Uning raqamlar yig’indisini aniqlovchi programma tuzilsin.
11. a va b butun sonlar berilgan. Shu sonlar qiymatlarini almashtiruvchi programma tuzilsin. (misol a=3, b=5 natija a=5, b=3)

2-Amaliy mashg'ulot

Mavzu: Matematik funksiyalar

Ishning maqsadi: PHP ni o'rganish va ma'lumotlar tiplaridan foydalanish, o'zgaruvchilar, amallar operasiyalar.

Talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak ma'lumotlar tiplari. o'zgaruvchilar, amallar va operasiyalar.

Qila olishi kerak PHPda ishlash;

NAZARIY BO'LIM

Barcha matemetik funksiyalar butun (int) va haqiqiy (float) sonlar uchun ishlatiladi.

#Funksiya	Izoh
1. number abs (mixed number)	|x| mobul
2. float floor (float value)	argumentdan kichik bo'lgan butun son qaytaradi
3. float ceil (float value)	argumentdan katta bo'lgan butun son qaytaradi
4. float round (float val [, int precision])	yaxlitlash uchun ishlatiladi
5. mixed max (number arg1, number arg2 [, ...]) mixed max (array numbers)	katta qiymatni qaytaradi
6. mixed min (number arg1, number arg2 [, ...]) mixed max (array numbers)	kichik qiymatni qaytaradi
7. int rand ([int min, int max])	min va max butun sonlari orasidan tasodifiy son qaytaradi
8. float sqrt (float arg)	ildiz chiqarish

fborva ceil funksiya la riga misol floor va ceil funksiyalariga misol

```
<?php
    // jasur.uz 02.05.2015
    // floor kichik butun son qaytaradi
    // ceil katta butun son qaytaradi

    $k = 7.9;     $k1 = -7.9;
    $q = 5.4;     $q1 = -5.4;

    echo 'floor(7.9)='.floor($k).'<br>';    // 7
    echo 'floor(5.4)='.floor($q).'<br>';    // 5


    echo 'ceil(7.9)='.ceil($k).'<br>';              // 8
    echo 'ceil(5.4)='.ceil($q).'<br>';              // 6

    echo 'floor(-7.9)='.floor($k1).'<br>'; // -8
    echo 'floor(-5.4)='.floor($q1).'<br>'; // -

    echo 'ceil(-7.9)='.ceil($k1).'<br>';    // -7
    echo 'ceil(-5.4)='.ceil($q1).'<br>';    // -5
```

```
?>
```

round funksiyasi

```
<?php
        // round yaxlitlash uchun ishltiladi
    $k = 7.9;     $k1 = -7.9;
    $q = 5.4;     $q1 = -5.4;

    echo 'round(7.9)='.round($k).'<br>';          // 8
    echo 'round(5.4)='.round($q).'<br>';       // 5

    echo 'round(-7.9)='.round($k1).'<br>';       // -8
    echo 'round(-5.4)='.round($q1).'<br>';       // -5
?>
```

Max funksiyasi

max funksiyasi katta sonni aniqlash uchun ishlatiladi, argument sifatida istalgancha sonlar Yoki massiv bo'lishi mumkin

```
<? php
    $k = 7;
    $t = 5;

    echo max($k, $t);  // 7 chiqadi

    echo '<br>';

    $a = array(2, 4, 5, 6, 7, 8);

    echo max($a);                 // 8 chiqadi
?>
```

rand tasodifiy son tanlash funksiyasi

```
<? php

    echo rand(10,  15).'<br>';         // 10 va 15 orasidan son chiqaradi
    echo rand(100, 200).'<br>';  // 100 va 200 orasidan son chiqaradi
    echo rand();          // Windowsda 0 va 32767 orasidan son chiqaradi
?>
```

Ko’p hollarda *PHP* tilini interpretatori ishlayotganligini tekshirib ko’rish uchun tuziladigan dastur eng sodda dastur deb ataladi. Hozir biz *PHP* tilidagi ushbu dasturni chuqur o’rganamiz hamda uni boshqa dasturlash tillari Ci, Perl va JavaScript lardan farqli tomonini tekshiramiz. Ushbu misolni ko’ramiz:

Misol. PHP kodida tuzilgan sodda html-fayl:

```
<html>
  <head>  <title> Misol </title>  </head>
  <body>
  <?php    echo "<p> Salom, bu men – PHP skript! </p>";    ?>
</body></html>
```

Bu *PHP* dasturlash tilining maxsus kodli teglari yordamida tuzilgan sodda html-fayldir.

Yuqorida aytib o’tganimizdek, *PHP* dasturlash tili Ci va Perl dasturlash tiliga o’xshash. Biroq keltirilgan dastur Ci va Perl dasturlash tilidagi dasturdan ancha katta farq qiladi. Bu yerda HTML sahifaga chiqarish uchun bir qator maxsus buyruqlarni yozish shart emas. Bevosita *PHP*-kod asosida qurilgan biror vazifani bajaradigan HTML-*skript* yoziladi (bizni misolda ekranda chiqarilgan matn). *PHP* dasturlash tilining Ci va Perl dasturlash tillaridan kamchiligi shuki, murakkab skriptlarni *PHP* dasturlash tili ancha sekin bajaradi.

PHP-skriptlar – bu serverda bajariladigan va qayta ishlanadigan dasturlardir. Bu skriptlarni JavaScript tipidagi skriptlar bilan taqqoslash mumkin emas, chunki JavaScript tilidagi skriptlarda yozilgan buyruqlar faqat kliyent kompyuteridagina bajariladi. Kliyent kompyuterida va server kompyuterida bajariladigan skriptlarning farqi nimada? Agarda skript serverda qayta ishlansa, mijoz kompyuteriga faqatgina natija yuboriladi. Masalan, agarda serverda skript bajarilayotgan bo’lsa, yuqorida keltirilganga o’xshab mijoz HTML-sahifa ko’rinishdagi natijani oladi:

```
<html>
   <head>  <title> Misol </title>   </head>
   <body>   <p> Salom, bu men – PHP skript! </p>
   </body></html>
```

Bu holatda mijoz qanday kod bajarilayotganini bilmaydi. O’z serveringizni HTML-fayllarni *PHP* prosessori qayta ishlaydigan qilib sozlab olishingiz ham mumkin. Ya’ni kliyentlar oddiy HTML-faylni qabul qildimi Yoki skript natijasini ko’rdimi buni bila olmaydi. Agarda skript kliyent kompyuterida qayta ishlansa (masalan, JavaScript tilidagi dastur), u holda kliyent skript kodidan iborat HTML-sahifani ko’radi.

Biz yuqorida aytib o’tgandikki, *PHP-skriptlar HTML*-kod ichida yoziladi. Qanday qilib degan savol tug’iladi. Buning bir nechta usullari mavjud. Bulardan biri birinchi misolda keltirilganidek, *<?php* tegi bilan boshlanib *?>* tegi bilan tugagan sintaksis. Bunday ko’rinishdagi maxsus teglar HTML va *PHP* rejimidagina ishlatiladi. Bu sintaksis *PHP* tilini *XML* hujjatlari bilan birgalikda ishlaydigan dasturlarida juda ma’qul ko’riladi (masalan, XHTML tilida yozilgan dasturlarda). Biroq bazan quyidagi alternativ variantdan foydalansa ham bo’ladi(echo "Some text" buyrug’i «Some text» matnini ekranga chiqaradi.):

```
<? echo "Bu PHP tilida  oddiy qayta ishlashning  instruksiyasi"; ?>
<script language="php">
 echo "Bir nechta redaktorlar (FrontPage) quyidagicha qabul qilishadi";
</script>
```

<% echo " ASP texnologiyasidagi tegdan ham foydalansa bo’ladi"; %>

Bu keltirilgan usullardan birinchisi har doim ham bajarilavermaydi. Undan foydalanish uchun qisqa teglarni ishlatish kerak, Yoki *PHP*3 uchun *short_tags*() funksiyani ishlatish kerak, Yoki *PHP* tilining konfigurasion fayliga short_open_tag buyruqni o’rnatish kerak, Yoki *PHP* dasturlash tilida *enable-short-tags* parametr bilan kompilyasiya qilish kerak. Agarda php.ini-dist buyruqqa yuqoridagilar avtomatik qo’shilgan bo’lsa, u holda qisqa teglardan foydalanish tavsiya etilmaydi. Ikkinchi usul xuddi o’rniga qo’yishga o’xshaydi, masalan, JavaScript kodlari va uning uchun mos html teglar. SHuning uchun undan har doim foydalanish mumkin, lekin bu noqulayligi uchun kamdan-kam ishlatiladi. Uchinchi usuldan faqat ASP texnologiyasidagi teglar *asp_tags* konfigurasiyasida ishlatilgandagina foydalaniladi.

PHP dasturlash tili faylni qayta ishlayotganda u oddiy matnni *PHP* kod interpretasiya qilishi kerak bo’lgan maxsus teglarni uchratmaguncha qaytarib beradi. Interpretator haqida gapirganda u topilgan barcha kodni yopiladigan teggacha bajaradi, so’ng yana oddiy matn qaytariladi. Bu mexanizm *PHP*-kodni HTML sahifaga aylantiradi, ya’ni barcha *PHP* teglardan tashqari barcha matnlarni o’zgarishsiz saqlaydi va ichkaridagilarni esa interpretasiyalaydi. Yana shuni aytish kerakki, php-fayl *CGI*-skriptga o’xshamaydi. php-fayl bajarilishi shart emas, Yoki yana qandaydir belgilanadi.

php-faylni serverda qayta ishlash uchun jo’natishda server tomonidan brauzer satrida bu faylni yo’lini ko’rsatish shart. *PHP* skriptlar www orqali kirishga ruxsat etilgan joyda joylashishi shart. Agarda php-fayl lokal kompyuterda mavjud bo’lsa, u holda uni buyruqlar satri interpretatori yordamida qayta ishlash mumkin.

ASOSIY SINTAKSISLAR

Instruksiyani bir nechta qismga bo’lib ko’rib chiqamiz, ya’ni kommentariylar yaratish, o’zgaruvchilar, o’zgarmaslar va ma’lumot tiplari, operatorlarga.

Biz endi *PHP* dasturlash tilining asosiy sintaksis elementlarini o’rganishga o’tamiz. Misol sifatida elektron maktub tayyorlash masalasini ko’rib o’taylik. Uning ma’nosi quyidagidan iborat.

Faraz qilarnizki, sizda qandaydir e’lon va e’lonni jo’natishingiz kerak bo’lgan bir nechta odamlar mavjud bo’lsin. Buning uchun siz e’lonni ichida o’zgaradigan (qabul qiluvchi bilan bog’liq bo’lmagan) bir nechta parametrlari mundarijasi bilan tayyorlaysiz.

Birinchi navbatda PHP dasturlash tili sintaksisiga nisbatan nimalarni bilish kerak. Bu HTML-kod ichiga o’rnatilgan va PHP dasturlash tilidagi koddir, uni interpretator farqlay biladi. Avvalgi bo’limlarda bular haqida aytib o’tgandik. Hammasini qaytarib o’tmaymiz, faqat biz ko’p hollarda misollarda <?php ?> variant o’rniga qisqartirilgan <? ?> teglardan foydalanishni aytib o’tamiz.

Instruksiyalarni ajratilishi

PHP dasturlash tilidagi dastur(ixtiyoriy dasturlash tilidagi) – bu buyruqlar (instruksiyalar) to'plarnidir. Dasturni qayta ishlash uchun bir buyruqni boshqa buyruqdan farqini bilish kerak. Buning uchun maxsus sinivollar – ajratgichlardan foydalaniladi. PHP dasturlash tilida instruksiyalarni xuddi Ci Yoki Perl dasturlash tillari kabi ajratiladi, ya'ni har bir ifoda nuqtali vergul (";") bilan tugaydi.

«?>» yopiladigan teg ham instruksiyani tugashini angiatadi, shuning uchun undan oldin nuqtali vergul qo'yilmaydi. Masalan, quyidagi ikki fragmentlar ekvivalentdir:

```
<?php
echo "Hello, world!"; // buyruqlar oxirida nuqtali vergul qo’yish shart
?>
<?php
echo "Hello, world!" ?>
<!-- "?>" borligi uchun nuqtali vergul tashlab ketildi -->
```

Kommentariylar

Ko'p hollarda dastur tuzganda kodni tushunarli bo'lishi uchun unga qandaydir izoh-***kommentariylar*** qo'yish kerak bo'lib qoladi. Bu holat katta hajmdagi dasturlar yaratganda hamda agarda bitta dastur ustida bir nechta dasturchi ishlayotganda juda muhim. Kommentariylar dasturning kodi tushunarli bo'lishi uchun yoziladi. Bundan tashqari masalani qismlarga ajratib hal qilinganda ishning kamchiligi bor joyida keyinchalik esdan chiqmasligi uchun kommentariya yozib qo'yiladi. Barcha dasturlash tillarida dastur ichiga kommentariya qo'shish imkoniyati mavjud. *PHP* dasturlash tili bir qancha ko'rinishdagi kommentariylarni qo'llab quvvatlaydi: Ci, C++ dasturlash tillari stilidagi hamda Unix qobig'idagi kommentariylar. // va # belgilar bir satrli kommentariylarni angiatsa, /* va */ belgilar esa mos ravishda ko'p satrli kommentariylarning boshlanish va tugashini angiatadi.

```
<?php
echo "Meni ismim Alisher";
  // Bu bir satrli kommentariy S++ dasturlash tili stilidagi
echo "Meni familiyam Boliyev";
/* Bu ko’p satrli kommentariy. Bu yerga bir qancha satr yozish mumkin. Dastur bajarilish jarayonida bu yerdagi barcha yozuvlar (kommentariylangan), o’qilmaydi. */
echo "Men PHP dasturlash tilini INTUIT.ru dan o’rganyapman";
  # Bu kommentariy Unix qobig’idagi kommentariy.
?>
```

Amallar

O'zgaruvchilar, o'zgarmaslar va ifodalar ustida turli hisoblashlarni bajaradigan bu ***amallar***dir. Biz hali bu ifodalar haqida to'xtab o'tganimiz yo'q. Ifodalar qiymatini ushbu amallar yordamida aniqlanadi. O'zgaruvchilar va o'zgarmaslar – bu ifodalarning asosiy va juda sodda shaklidir. SHunday ifodalarni

ko’paytirishi mumkin bo’lgan amallar to’plarni mavjud. Ularni quyida to’liqroq muhokama qilarniz:

2.1-jadval. Arifmetik amallar.		
Belgilanishi	**Nomlanishi**	**Misol**
+	Qo’shish	$a + $b
-	Ayirish	$a - $b
*	Ko’paytirish	$a * $b
/	Bo’lish	$a / $b
%	Bo’lishdagi qoldiq	$a % $b

2.2-jadval. Satrli amallar.		
Belgilanishi	**Nomlanishi**	**Misol**
.	*Konkatenasiya* (*satrlarni qo’shish*)	$c = $a . $b (bu $c satr $a va $b satrlardan iborat)

2.3-jadval. O’zlashtirish amallari.			
Belgilanishi	**Nomlanishi**	**Izox**	**Misol**
=	O’zlashtirish	*Operatordan* o’ng tomonda turgan o’zgaruvchilar ustida bajarilgan amallardan hosil bo’lgan natija qiymati o’zlashtiriladi.	$a = ($b = 4) +5; ($a 9 ga teng, $b 4 ga teng)
+=		Qisqartirish. O’zgaruvchiga son qo’shiladi va keyin natija o’zlashtiriladi.	$a += 5; ($a = $a + 5 ifodaga ekvivalent;)
.=		O’zlashtirish va konkatenasiya amallari kombinasiyasini qisqartirilgan shakli(dastavval satrlar qo’shiladi, so’ngra hosil bo’lgan satr o’zgaruvchiga o’zlashadi).	$b = "Hammaga "; $b .= "salom"; ($b = $b . "salom" ifodaga ekvivalent;) Natijasi: $b="Hammaga salom"

<table>
<tr><th colspan="4">2.4-jadval. Matiqiy amallar.</th></tr>
<tr><td>Belgilanishi</td><td>Nomlanishi</td><td>Izox</td><td>Misol</td></tr>
<tr><td>And</td><td>VA</td><td>$a va $b rost (True)</td><td rowspan="2">$a and $b
$a && $b</td></tr>
<tr><td>&&</td><td>VA</td><td></td></tr>
<tr><td>Or</td><td>Yoki</td><td>$a Yoki $b o’zgaruvchilardan hech bo’lmaganda bittasi rost bo’lsa (ikkalasi ham rost bo’lishish mumkin).</td><td>$a or $b</td></tr>
<tr><td>||</td><td>Yoki</td><td></td><td>$a || $b</td></tr>
<tr><td>Xor</td><td>Inversiya Yoki</td><td>O’zgaruvchilardan bittasi rost bo’lsa. Agarda ikkalasi ham rost bo’lsa inversiyalanadi.</td><td>$a xor $b</td></tr>
<tr><td>!</td><td>Inversiya (NOT)</td><td>Agarda $a=True, u holda !$a=False va aks holda teskarisi bo’ladi.</td><td>! $a</td></tr>
</table>

2.5-jadval. Taqqoslash amallari.

Belgilanishi	**Nomlanishi**	**Izox**	**Misol**
==	Tengiik	O’zgaruvchilar qiymatlari teng	$a == $b
===	Ekvivalentlik	O’zgaruvchilar qiymatlari va tiplari teng	$a === $b
!=	Tengsizlik	O’zgaruvchilar qiymatlari teng emas	$a != $b
<>	Tengsizlik		$a <> $b
!==	Noekvivalentlik	O’zgaruvchilar ekvivalent emas	$a !== $b
<	Kichik		$a < $b
>	Katta		$a > $b
<=	Kichik Yoki teng		$a <= $b
>=	Katta Yoki teng		$a >= $b

2.6-jadval. Inkrement va dekrement amallari.

Belgilanishi	**Nomlanishi**	**Izox**	**Misol**
++$a	Pre-*inkrement*	$a qiymati birga oshiriladi va $a qiymati qaytariladi	<? $a=4; echo "4 bo’lishi shart:" .$a++; echo "6 bo’lishi shart:" .++$a; ?>
$a++	Post-*inkrement*	$a qiymati qaytariladi va so’ngra $a qiymati birga oshiriladi	
--$a	Pre-*dekrement*	$a qiymati birga kamaytiriladi va $a qiymati qaytariladi	
$a--	Post-*dekrement*	$a qiymati qaytariladi va so’ngra $a qiymati birga kamaytiriladi	

3-Amaliy mashg'ulot
Mavzu: O'zgaruvchi va o'zgarmaslar

Ishning maqsadi: PHP ni o'rganish va ma'lumotlar tiplaridan foydalanish, o'zgaruvchilar, amallar operasiyalar.

Talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak ma'lumotlar tiplari. o'zgaruvchilar, amallar va operasiyalar.

Qila olishi kerak PHPda ishlash;

O'zgarmaslar o'zgaruvchilar kabi o'zida ma'lumot saqlaydi. O'zgarmaslarning asosiy farqi e'lon qilinganidan keyin uning qiymatini o'zgartirib bo'lmaydi.

O'zgarmasga misol qilib saytingiz serverning qaysi katalogida joylashganini ko'rsatish mumkin. Quyida ROOT_LOCATION o'zgarmasini e'lon qilish keltirilgan:

define(„O'zgarmas nomi", „O'zgarmas qiymqti");

Keyin undan oddiy o'zgaruvchi kabi foydalanish mumkin. Faqat o'zgarmas oldidan **$** belgisi qo'yilniaydi.

$uzgaruvchi=O'zgarmas nomi;

O'zgarmaslar bilan ishlash uchun 4 ta asosiy narsani bilish kerak.

1.O'zgarmaslar ***define*** kalit so'zi bilan e'lon qilinadi.

2.O'zgarmaslarga murojaat qilishda **$** belgisi ishlatilmaydi.

3.O'zgarmaslarning qiymatini dastur ishlashi jarayonida o'zgartirib bo'lmaydi.

4.O'zgarmaslarni katta harflar bilan yozishga kelishilgan. Kichik harflar bilan yozsangiz xato bo'lmaydi. Lekin professional dasturchilar kelishuviga xilof ish qilgan bo'lasiz.

PHP da bir qator o'zgarmaslar kiritilgan bo'lib ularni sehrli o'zgarmaslar (Magic constants) deyiladi. Sehrli o'zgarmaslar oldidan va ketidan ikkita ost chiziq yoziladi. Bu boshqa o'zgarmaslar bilan bir xil bo'lib qolmasligi uchun qilingan.

Sehrli o'zgarmaslar jadvali (PHP's magic constants)

Magic constants	Izoh
`__LINE__`	Fayldagi joriy satr nomeri
`__FILE__`	Fayl nomiga yo'lni korsatadi. Agar include orqali ishlatilgan bo'lsa, qo'shilgan fayl nomi qaytariladi.
`__DIR__`	Katalog nomini qaytaradi. `dirname(__FILE__)` funksiyasi bilan bir xil ishlaydi. PHP 5.3.0 versiyasidan boshlab qo'shilgan.
`__FUNCTION__`	Funksiya nomini qaytaradi. PHP 5 dan boshlab funksiya qanday e'lon qilingan bo'lsa, shundayligicha qaytaradi. PHP 4 da faqat kichkina harflarda qaytaradi. PHP 4.3.0 versiyasidan boshlab qo'shilgan.
`__CLASS__`	Klass nomini qaytaradi. PHP 5 dan boshlab klass qanday e'lon qilingan bo'lsa, shundayligicha qaytaradi. PHP 4 da faqat kichkina harflarda qaytaradi. PHP 4.3.0 versiyasidan boshlab qo'shilgan.
`__METHOD__`	Metod nomini qaytaradi. Metod qanday e'lon qilingan bo'lsa, shundayligicha qaytaradi. PHP 5.0.0 versiyasidan boshlab qo'shilgan.
`__NAMESPACE__`	Joriy nomlar fazosini qaytaradi. Bu o'zgarmas kompilyatsiya vaqtida aniqlanadi.PHP 5.3.0 versiyasidan boshlab qo'shilgan.

O'zgarmaslarni e’lon qilish va foydalanish

```
<?php
        // o'zgarmaslarni katta harflarda yozing
        // o’zgarmasni e'lon qilish
        define('PI', '3.14159');
        $r = 10;
        // o'zgarmasga murojaat qilganda $ belgisi yozilmaydi
        $l = 2 * PI * $r;
        echo $l;
?>
```

O’ZGARUVCHILAR, O’ZGARMASLAR VA OPERATORLAR.

Har bir dasturlash tilida muhim elementlardan biri bu *o’zgaruvchilar*, *o’zgarmaslar* va ular qo’llaniladigan *operatorlar*dir. PHP dasturlash tili bu elementlarni qanday belgilashi va qayta ishlashini ko’rib chiqamiz.

O’zgaruvchilar

PHP dasturlash tilida *o’zgaruvchilar* oldiga dollar belgisi (“$”) qo’yib e’lon qilinadi, masalan, $my_var.

O’zgaruvchilar nomlari registrlarni farqlaydi, ya’ni $my_var hamda bosh harfli $My_var o’zgaruvchilari turli xil o’zgaruvchilardir.

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

*PHP*3 dasturlash tilida o’zgaruvchilar har doim biror-bir qiymatga o’zlashtiriladi. Ya’ni o’zgaruvchini biror-bir ifodaga o’zlashtirsak, ifodaning qiymati o’zgaruvchiga o’zlashadi. Buni quyidagi misolda ko’rish mumkin, ya’ni bir o’zgaruvchini qiymati boshqasiga o’zlashtirilganda ulardan birini qiymatini boshqasiga ta’sir ko’rsatmaydi.

Misol. Qiymat bo’yicha o’zlashtirish

```
<?php
$first = ' Text '; // $first o’zgaruvchiga ' Text ' qiymatni o’zlashtirdik
$second = $first; // $second o’zgaruvchiga $first o’zgaruvchi qiymati o’zlashtirildi
$first = ' New text '; // $first o’zgaruvchi qiymati ' New text ' qiymatga o’zgartirildi
echo "first nomli o’zgaruvchi qiymati ".
    "$first ga teng. <br>";
     // $first o’zgaruvchi qiymatini ekranga chiqaramiz
echo "second nomli o’zgaruvchi qiymati ".
    "$second ga teng. ";
       // $second o’zgaruvchi qiymatini ekranga chiqaramiz
?>
```

Bu skriptni natijasi quyidagicha bo’ladi:

first nomli o’zgaruvchi qiymati Text ga teng.
second nomli o’zgaruvchi qiymati New text ga teng.

*PHP*4 dasturlash tilida bulardan tashqari o’zgaruvchiga qiymat o’zlashtirishning yana bir usuli mavjud: ssылka bo’yicha *o’zlashtirish*. Ssылka bo’yicha o’zgaruvchiga qiymat o’zlashtirish uchun uni nomi bo’lishi shart, ya’ni u qandaydir o’zgaruvchini taqdim etishi kerak. Bir o’zgaruvchi qiymatini boshqa o’zgaruvchiga Ssылka bo’yicha *o’zlashtirish* uchun birinchi o’zgaruvchi oldiga ampersand & belgisi qo’yish shart.

Bunga yuqoridagi misolni ko’rib chiqamiz, faqat first o’zgaruvchi second o’zgaruvchiga ssылka bo’yicha o’zlashtiriladi:

Misol. Ssылkalar bo’yicha o’zlashtirish

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

Bu skriptni natijasi esa quyidagicha bo’ladi:

first nomli o’zgaruvchi qiymati New text ga teng.

second nomli o’zgaruvchi qiymati New text ga teng.

Ya’ni $first o’zgaruvchi qiymati o’rniga $second o’zgaruvchi qiymati o’zlashtirildi.

O’zgarmaslar

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

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

O’zgarmaslar nomi registrga segirligi katta. Har bir o’zgarmaslarda uni o’zgartirish mumkin, ya’ni *registrga_sezgirligi_kichik* argumentni qiymati sifatida True ыiymati ko’rsatiladi. O’zgarmaslar nomi har doim katta registr bilan yozishga kelishib olingan.

O’zgarmasni qiymatini bilish uchun uni nomini ko’rsatish kerak. O’zgaruvchidan farqi o’zgarmas nomi oldiga $ belgi qo’yilmaydi. Bundan tashqari

o’zgarmasni qiymatini bilish uchun konstanta nomi bilan parametr sifatida constant() funksiyasidan foydalanish mumkin.

Misol. PHP dasturlash tilida o’zgarmaslar

```
<?php
// o’zgarmasni aniqlaymiz PASSWORD
define("PASSWORD","qwerty");
// registrlanmagan PI o’zgarmasni qiymatini aniqlaymiz 3.14
define("PI","3.14", True);
// PASSWORD o’zgarmas qiymatini olarniz,  ya’ni qwerty
echo (PASSWORD);
// bu ham qwerty ni chiqaradi
echo constant("PASSWORD");
echo (password);
/* password ni chiqaradi va biz registrlangan o’zgarmas PASSWORD ni kutgandik.*/
echo pi;
// 3.14 ni chiqaradi, chunki o’zgarmas PI  registrlanmagan va aniqlangan.
?>
```

Dasturchi tomonidan o’zgaruvchilardan tashqari yuqorida aytib o’tganimizdek *PHP* dasturlash tilida mavjud o’zgarmaslar ham interpretator tomonidan aniqlanadi. Masalan, __FILE__ o’zgarmas dastur bajarilish jarayonida fayl nomini (va fayl yo’lini), __FUNCTION__ funksiya nomidan tashkil topadi, __CLASS__ - sinf nomi, PHP_VERSION – *PHP* dasturlash tili interpretatori versiyasini o’zida saqlaydi. Bunday o’zgarmaslarning barcha ro’yxatini *PHP* dasturlash tili uchun mo’ljallangan qo’llanmalardan topish mumkin.

MA’LUMOTLAR TIPLARI

PHP dasturlash tili sakkizta sodda *ma’lumot tiplar*ini qo’llab quvvatlaydi:

To’rttasi skalyar *tiplar*:

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

Ikkitasi aralish *tiplar*:

- *array (massiv)*;
- *object (obyekt)*.

Ikkitasi maxsus *tiplar*:

- *resource (resurs)*;
- *NULL*.

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

Boolean tipi(Bul Yoki mantiqiy tip).

Bu sodda tip qiymatni rost ekanligini ifodalaydi, ya’ni o’zgaruvchi faqat ikkita qiymat qabul qiladi – rost TRUE Yoki yolg’on FALSE.

Mantiqiy tiplarni aniqlash uchun TRUE Yoki FALSE kalit so’zlaridan foydalanamiz. Bu ikkala tiplar registrlanmagan.

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

Mantiqiy tiplar turli *boshqariladigan konstruksiyalar*da (sikllar, shartlar va shunga o’xshash, bular haqida keyinroq aytib o’tamiz) qo’llaniladi. Bir qancha amallar (masalan, tengiik amali) ham mantiqiy tip qabul qilishi mumkin, ya’ni faqat ikki qiymat rost Yoki yolg’on qiymatni qabul qiladi. Ular *boshqariladigan konstruksiyalar*da shartlarni tekshirish uchun qo’llaniladi. Masalan, shartli konstruktorda amallar Yoki o’zgaruvchilar qiymati haqiqiyligini tekshiradi va natijadan qat’iy nazar shu Yoki boshqa amallarni bajarilishini tekshiradi. Bu yerda shart rost Yoki yolg’on bo’lishi mumkin, chunki *mantiqiy tip amallari* va *o’zgaruvchilar* ko’rsatilgan.

Misol. Mantiqiy tiplarning qo’llanilishi

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

Integer (butun) tipi.

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

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

```
<?php
# o’nlik raqam
$a = 1234;
# manfiy son
```

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

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

Agar siz *butun son* chegarasidan tashqari biror qiymat bersangiz interpretator bu sonni *qo'zg'aluvchan vergulli son*ga o'zgartiradi. Xuddi shunday *butun son* chegarasidan tashqari chiqib ketadigan biror amal bajarsangiz ham bu sonni *qo'zg'aluvchan vergulli son*ga o'zgartiriladi.

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

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

*Qo'zg'aluvchan vergulli son*lar (ular ikki karra aniqlik Yoki haqiqiy sonlardir) quyidagi sintaksislar yordamida aniqlanishi mumkin:

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

*Qo'zg'aluvchan vergulli son*ni o'lchami ham platformaga bog'liq, lekin qoidaga ko'ra maksinial qiymati ~1.8e308 aniqlik bilan 14 xonali raqam atrofida bo'ladi.

String (satr) tipi

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

PHP dasturlash tilida satrlar uchta turli xil usullarda aniqlanadi:

- *bittalik qo'shtirnoqlar* yordamida (‘’);
- *qo'shtirnoqlar* yordamida (“”);
- *heredoc-sintaksisi* yordamida.

Bittali tirnoqlar

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

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

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

```
<?php
echo 'Satrlar majmui';

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

Array (massiv) tipi

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

*Massivlar*ni *array*() konstruksiyasi yordamida aniqlanadi Yoki elementlariga qiymat berish bilan aniqlanadi.

array() konstruksiyasi yordamida aniqlash.

array ([key] => value, [key1] => value1, ...)

PHP dasturlash tilining ***array***() konstruksiyasi vergul bilan ajratilgan juft parametrlar *kalit* => ***qiymat*** bilan ajratilgan. => belgi mos ravishda qiymat va uning kaliti o'rtasida aloqa o'rnatadi. Kalit butun son bo'lishi mumkin, uning qiymati esa *PHP* dasturlash tilidagi ixtiyoriy tipni qabul qilishi mumkin. Kalit raqamini biz ko'pincha indeks deb ataymiz. *PHP* dasturlash tilida indekslash noldan boshlanadi. Massiv elementining qiymatini olish uchun massiv nomi va

kvadrat qavs ichida uning kaliti ko'rsatilishi kerak. Agar massiv kaliti standart butun son bo'lsa, u holda uning qiymatini butun son deb qarasa bo'ladi, aks holda u satr deb qaraladi. SHuning uchun $a["1"] yozuv $a[1] yozuvga teng kuchli, $a["-1"] yozuv esa $a[-1] yozuvga teng kuchli.

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

Misol. PHP dasturlash tilida massivlar.

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

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

Misol. PHP dasturlash tilida massivlar.

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

Kvadrat qavs sintaksisi yordamida aniqlash.

Massivga qiymat berish orqali massiv yaratish mumkin. Biz yuqorida aytib o'tganimizdek, massiv elementi qiymatiga ega bo'lish uchun kvadrat qavs ichiga uning kaliti ko'rsatilishi kerak, masalan, $book["php"]. Agarda yangi kalit va yangi qiymat ko'rsatsangiz quyidagicha bo'ladi: $book["new_key"]="new_value" hamda massivga yangi element qo'shiladi. Agarda kalitni ko'rsatmay faqat qiymatni o'zlashtirsak, ya'ni $book[]="new_value", u holda massivga yangi element qo'shiladi va uni kaliti mavjud maksinial qiymatga bir qo'shiladi. Agarda biz qiymat bergan *massiv* yaratilmagan bo'lsa, u holda biz qiymat bergandan keyin u *yaratiladi*.

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

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

```
<?php
$books = array ("php" =>"PHP users guide",12 => true);
$books[] = "Book about Perl"; /* 13-kalit(indeks) bilan yangi element qo’shildi, bu quyidagiga ekvivalent $books[13] ="Book about Php"; */
$books["lisp"] = 123456; /* Bu massivga yangi "lisp" kalitli 123456 qiymali yangi element qo’shish*/
unset($books[12]); // Bu 12-kalitli elementni massivdan o’chirish
unset ($books); // massivni butunlay o’chirish
?>
```

Bo’sh kvadrat qavsdan foydalanganda kalitning maksinial qiymati massivda mavjud oxirgi qayta indekslangan kalitlar orasidan qidiriladi. Massivni ***array_values***() funksiyasi yordamida ***qayta indekslash*** mumkin.

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

Misol. Massivni qayta indekslaymiz.

Bu skriptning natijasi quyidagicha bo’ladi:

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

4-Amaliy mashg'ulot

Mavzu: SQL so'rovlar tili

Ishning maqsadi: PHP da ma'lumotlar bazasi bilan ishlashni o'rganish

Talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak: ma'lumotlar bazasi bilan bog'lanishni va so'rovlar bilan ishlashni

Qila olishi kerak: PHPda MySql bilan birga ishlash;

SQL tili (**Structured Query Language** - strukturalashgan so'rovlar tili) sintaksisi juda oson. SQL tilida katta - kichik harflar farqlanlmaydi. Ya'ni Select operatorini SELECT, Select, select shakllarida yozish mumkin. Agar bir nechta operatordan foydalanilsa operator oxirida qo'yiladi. Faqat bitta operator ishlatilsa operator oxirida ; qo'yish shart emas. Izoh yozish ba'zi sistemalarda/* <IZOH> */ shaklida, ba'zi sistemalarda esa {<IZOH>}shaklida bo'ladi.

SQL so'rovlar tilidan deyarli barcha (MS SQL Server, MySQL, PostgreSQL, Oracle, Informix, Paradox. Interbase, FireBird) ma'lumotlar bazasida foydalanish mumkin. SQL so'rovlari ma'lumotlar bazasiga qarab qisman o'zgarishi mumkin.

Select tanlash operatori

Select operatorining umumiy ko'rinishi

SELECT<maydon nomlari ro'yhati> **FORM** <jadvallar> **WHERE** <izlash sharti>

GROUP BY<guruhlanishi lozim bo'lgan maydonlar>

HAVING <izlash sharti>

ORDER BY <maydon nomlari ro'yhati>;

Quyidagi maydonlarni o'zida saqlovchi namuna nomli jadval hosil qiling va uni Lesson sahifasiga saqlang.

Field Name	Type	Size
ID	+	
Familiya	A	20
Ism	A	20
Otasi	A	20
Guruh	A	5
Tug_kun	D	
Tug_yil	S	
Jinsi	A	5

1. Bu jadvaldan ma'lumot ko'rish quyidagicha bo'ladi.

SELECT* FROM namuna

2. **J**advaldagi ba'zi maydonlarni ko'rish uchun shu maydon nomlari ro'yhat selectdan keyin keltiriladi:

SELECT Familiya, Ism, Otasi **FROM** namuna

3. Familiya bo'yicha tartiblash

Select familiya, ism, otasi **From** namuna **OrderBy** Familiya, Ism

4. Tug'ilgan yili bo'yicha kamayib borish tartbida tartiblash

Select familiya, ism, otasi, tug_yili **FROM** namuna **OrderBy** tug_yil **DESC**

5. Select operatoridan keyin faqat maydon nomi emas, ixtyoriy arifmetik amai (+, -, *,/) ishlatish mumkin. Misol uchun, yuqoridagi jadvaldagi tug'ilgan yilni joriy yildan ayirib, talabaning yoshini aniqlaymiz.

Select familiya, ism, otasi, (2015-tug_yil**)** **AS** Yoshi **FROM** namuna

6. Agar maydon nomini ruscha shriftda chiqarish talab etilsa quyidagicha so’rov yoziladi: (Maydon nomidan keyin **AS** xizmatchi so'zi bilan maydon sarlavhasi keltirilladi. Faqat "" harfida muammo bo'lishi mumkin)

Select familiya **AS** Фамилия, ism **A S Исм FROM** namuna

Where operatoridan keyin tanlash shartida mantiqiy amallar **AND, OR, NOT,** arifmetik amallar va munosabat amallari:

=	Teng
>	tetta
>=	tetta yok teng
<	Kichik
<=	Kichik: Yoki teng
!= yold <>	Teng emas
LIKE	belgilar ketma - ketligi
between ... and	Qiymatlar oralig'i
m	Toplamga tegishlilikni aniqlash

7. 1990 yildan keyin tug’ilganlar ro’yhatini chiqaramiz.

Select Familiya, Ism, Otasi, tug_yiI, (2015-tug_yil) **AS** yoshi **FROM** namuna

Where jinsi=”Ayol” **and** tug_yil> 1990

8. **Like** amali sintaksisi quyidagicha: <maydon> **LIKE** *<belgilar ketma - ketligi>"

Bu amal satrli maydonlar uchun qollaniladi va izlanayotgan ma’lumot topilsa true qiymat qaytaradi. Belgilar ketma - ketligini toliq kiritish Yoki belgisi bilan tugatish mumkin. Bu belgi ixtiyoriy belgilar ketma - ketligini bildiradi.

Misol:

Select familiya, ism, otasi

FROM namuna

Where Familiya **LIKE** "A%"

Amali "A" harfidan boshlanuvchi birinchi familiyalani bildiradi. (Adambaev, Abdurahimov. Alimov, Azamov, Asqarov, Azizov,...)

Select familiya, ism, otasi **FROM** namuna

Where Familiya **LIKE** "Abdu%"

Amali *Abdu' harflaridan boshlanuvchi barcha familiyalarni bildiradi. (Abdurahimov, Abdug‘aniev, Abduqayumov, Abdullaev)

Select familiya, ism, otasi

FROM namuna

Where Ism **LIKE** "%im%"

Amali ketma – ketlikda “im” qismi bor bo’lgan ism borligini bildiradi. (Karimboy, Abdukarim, Rahimjon, Hakimboy, Salimjon,...)

9.**between ... and** amali sintaksisi quyidagicha:

<maydon> **between** <qiymat> **and** <qiymat>

1985 va 1990 yillar oraliglda tugllganlar ro"yhatini chiqarish

Select familiya, ism, otasi **FROM** namuna

WHERE tug_yiI **BETWEEN** 1985 AND 1990

10. IN amali sintaksisi quyidagicha: <maydon> IN (< to"plam >)

Maydondagi malumot qiymati, to’plam elementlariga tegishli bolganlarini chiqarib beradi. Masalan, ismi Azamat, Qudrat va Ne’mat bolgan talabalar haqidagi malumotlarni chiqarish talab etilsin:

SELECT * FROM namuna

WHERE Ism IN ("Azamat"," Qudrat"," Ne ■mat")

Tug’ilgan yili 1980,1982,1989 bo’lganlar ro’yxati esa quyidagicha:

SELECT familiya, ism, otasi, tug_yil FROM namuna

WHERE tug_yil IN (1980,1982,1989)

11. DISTINCT xizmatchi so’zi ko’rsatilgan maydon bo’yicha bir xil qiymatli malumotlardan faqat bittasini olishni bildiradi

SELECT DISTINCT familiya FROM namuna

12. **Count** ko’rsatilgan maydon boyicha, Yoki butun jadvaldagi yozuvlar sonini aniqlash uchun ishlatiladi

Select count (*) FROM namuna

95-99 guruhidagi talabalar sonini aniqlash:

Select count (*) as soni FROM namuna WHERE guruh =“95-99"

13. **min**(<maydon>), **max**(<maydon>), **avg**(<maydon>), **sum**(<maydon>) funksiyalari mos ravishda maydonning eng kichik qiymatini (min), eng katta qiymatini (max), o'rtacha qiymatini (avg) va yig’indisini (sum) aniqlaydi.

Select min(tug_yil), **max**(tug_yil), **avg**(2010-tug_yil) FROM namuna

14. Quyida yoshi eng katta bo’lgan talabani aniqlovchi so'rov keltirilgan

Select familiya, ism, tug_yil FROM namuna

WHE RE tug_yi l=**(select min**(tug_yi I) FROM na muna)

15. Guruhg agi talabalar sonini aniq lovchi so "rov

Select guruh. count(guruh) **as** soni **From** namuna **Group by** guruh

16. **Having** bo'limi. group by bo'limi bo'lgan hollarda ishlatiladi. **Having** ning ishlatilishi deyarli **Where** bilan bir xil. Faqat having bo'limidagi so'rov butun jadvallarga emas. **where** bo'limi orqali olingan natijaga yoziladi.

Quyida birdan ortiq takrorlanuvchi familiyalar va ularning takrorlanishlar sonini chiqaruvchi so’rov keltirilgan.

select familiya. count(familiya) **as** soni
from namuna
group by familiya
having count(familiya) ***>=2***

Jadvalga yangi yozuv qo’shish **INSERT** operatori orqali amalgani oshiriladi

INSERT INTO <jadval nomi> (<maydonlar ro’yhati>) **VALUES** (<qiymatlar ro’yhati>)

Namuna:

INSERT INTO namuna (familiya, ism) **VALUES** ["Abdurahimov", "Ne'mat")

Mavjud yozuvni tahrirlash (qiymatini yangilash) **Update** operatori orqali amalga oshiriladi:

UPDATE <jadval nomi> **SET** <maydon>=<qiymat> **WHERE** <shart>

Namuna: Abdurahimov Ne'matning tug’ilgan yilini 1989 ga o’zgartirish

Update namuna **SET** tug_yil=1989
WHERE familiya ="Abdurahimov" **AND** ism=”Nemat"

Yozuvni o’chirish quyidagicha amalga oshiriladi:

DELETE FROM <jadvalnomi> **WHERE** <shart>

DELETE FROM namuna **WHERE** familiya=”Abdurahimov"

Bu so'rov orqali, familiyasi "Abdurahimov" bo'lgan barcha ma'lumotlar o'chiriladi.

MySQL ma'lumotlar bazasi

PHP dasturchilari orasida eng mashhur ma'lumotlar bazasi MySQLdir. **MySQL nima?**

MySQLwebda ishlatiladigan ma'lumotlar bazasi.
MySQL ma'lumotlar bazasi serverda ishlaydi.
MySQL ni kichik va katta loyihalar uchun ham qo'llash mumkin.
MySQL tez, ishonchli va foydalanishga qulay.
MySQL da standart SQL so'rovlar tili ishlatiladi.
MySQL foydalanish va yuklab olish uchun tekin.
MySQL Oracle Corporation tomonidan ishlab chiqilgan.

5-Amaliy mashg'ulot

Mavzu: **MySQL da ma'lumot turlari**

Ishning maqsadi: PHP da ma'lumotlar bazasi bilan ishlashni o'rganish

Talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak: ma'lumotlar bazasi bilan bog'lanishni va so'rovlar bilan ishlashni

Qila olishi kerak: PHPda MySql bilan birga ishlash;

MySQL da 3 ta asosiy turlar mavjud. Ularni atn (text) son [number), va sana/vaqt (Date/Time) turlaridir. In MySQL there are three main types: text, nurrtoer, and Datemme types.

Matnli turlar (Text):

Data type	Tavsifi
CHAR(size)	Aniq o'lchamdagi satrni saqlash uchun ishlatiladi. Bunda harf, son va maxsus belgilar bo'lishi mumkin. Satr o'lchami qavs ichida beriladi. 255 ta belgini saqlash mumkin.
VARCHAR(size)	Satrli ma'lumotlarni saqlash uchun ishlatiladi. Bunda harf, son va maxsus belgilar bo'lishi mumkin. Satr o'lchami qavs ichida beriladi. 255 dan katta son yozilsa toifa TEXT toifasiga o'zgarib ketadi.
TINYTEXT	Ko'pi bilan 255 ta belgili satrni saqlash uchun ishlatiladi.
TEXT	Ko'pi bilan 65,535 ta belgili satrni saqlash uchun ishlatiladi.
BLOB	(Binary Large Objects). Katta binar ob'ektlarni saqlash uchun ishlatiladi. Maksimal 65,535 bayt bo'lishi mumkin.
MEDIUMTEXT	Ko'pi bilan 6,777,215 ta belgili satrni saqlash uchun ishlatiladi.
MEDIUMBLOB	(Binary Large Objects). Binar ob'ektlarni saqlash uchun ishlatiladi. Maksimal 16,777,2 bayt bo'lishi mumkin.
LONGTEXT	Ko'pi bilan 4,294,967,295 ta belgili satrni saqlash uchun ishlatiladi.
LONGBLOB	(Binary Large Objects). Binar ob'ektlarni saqlash uchun ishlatiladi. Maksimal 4,294,967,295 bayt bo'lishi mumkin.
ENUM(x,y,z,etc.)	O'zingiz maydon qabul qilishi mumkin bo'lgan qiymatlar ro'yxatini kiritasiz. 65535 ta qiymat kiritishingiz mumkin. Agar maydonga kiritilgan qiymat siz ko'rsatgan ro'yxatda bo'lmasa, bo'sh qiymat qo'shiladi. Ro'yxatni quyidagi shaklda kiritishingiz mumkin: ENUM('X','Y','Z')
SET	ENUM ga o'xshash. Farqi 64 ta element bo'lishi mumkin.

Son turlari (Number types):

Data type	Tavsifi
TINYINT(size)	Odatda -128 dan 127 gacha. UNSIGNED* bo'lsa, 0 - 255. Raqamlarning maksimal sonini qavs ichida berish mumkin.
SMALLINT(size)	Odatda -32768 dan 32767 gacha. UNSIGNED* bo'lsa 0 - 65535 . Raqamlarning maksimal sonini qavs ichida berish mumkin.
MEDIUMINT(size)	Odatda -8388608 dan 8388607 gacha. UNSIGNED* bo'lsa 0 - 16777215. Raqamlarnin maksimal sonini qavs ichida berish mumkin.
INT(size)	Odatda -2147483648 dan 2147483647 gacha. UNSIGNED* bo'lsa 0 - 4294967295. Raqamlarning maksimal sonini qavs ichida berish mumkin.
BIGINT(size)	Odatda -9223372036854775808 dan 9223372036854775807 gacha. UNSIGNED* bo'l 0 - 18446744073709551615. Raqamlarning maksimal sonini qavs ichida berish mumkin
FLOAT(size,d)	Kichik haqiqiy son. Raqamlaming maksimal soni size parametrida berishilishi mumkin. Verguldan keyingi xonalar soni d parametrida beriladi.
DOUBLE(size,d)	Katta haqiqiy son. Raqamlarning maksimal soni size parametrida berishilishi mumkin. Verguldan keyingi xonalar soni d parametrida beriladi.
DECIMAL(size,d)	Juda katta haqiqiy sonlar satr ko'rinishida saqlanadi. Raqamlarning maksimal soni size parametrida beriladi. Verguldan keyingi xonalar soni d parametrida beriladi.

* Butun sonlar UNSIGNED nomli qo'shimcha xususiyatga ega. Odatda butun sonlar manfiy va musbat bo'ladi. UNSIGNED xususiyati qo'shilishi orqali noldan boshlanadi va qabul qilishi mumkin bo'lgan oraliq suriladi.

Sana turlari (Date types):

Data type	Tasnifi
DATE()	YYYY-MM-DD formatidagi sana. '1000-01-01' dan '9999-12-31' gacha oraliqda qiymat qabul qiladi
DATETIME()	YYYY-MM-DD HH:MI:SS formatidagi sana va vaqt birligi. '1000-01-01 00:00:00' dan '9999-12-31 23:59:59' gacha oraliqda qiymat qab qiladi
TIMESTAMP()	*TIMESTAMP qiymati sekundlarda saqlaniladi. ('1970-01-01 00:00:00' UTC). Formati YYYY-MM-DD HH:MI:SS **Eslatma:**'1970-01-01 00:00:01' UTC dan '2038-01-09 03:14:07' UTC gacl oraliqda qiymat qabul qiladi.
TIME()	Vaqt. Formati: HH:MI:SS **Eslatma:** '-838:59:59' dan '838:59:59' gacha oraliqda qiymat qabul qiladi
YEAR()	Yil. Ikki yoki to'rt xonali raqamdan iborat bo'ladi. **Eslatma:** 4 xonali: 1901 , 2155. Ikki xonali: 70 dan 69, 1970 dan 2069 gacl oraliqda

* DATETIME va TIMESTAMP bir xil formatli bo'lsada, har xil ishlaydi. INSERT yoki UPDATE so'rovlarida TIMESTAMP avtomat ravishda joriy sana va vaqtni qo'shadi.TIMESTAMP da yana YYYYMMDDHHMISS, YYMMDDHHMISS, YYYYMMDD, yoki YYMMDD formatlarini qo'llash mumkin.

6-Amaliy mashg'ulot

Mavzu MySQL ga ma'lumot qo'shishni **rejalashtirish**

Ishning maqsadi: PHP da ma'lumotlar bazasi bilan ishlashni o'rganish

Talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak: ma'lumotlar bazasi bilan bog'lanishni va so'rovlar bilan ishlashni

Qila olishi kerak: PHPda MySql bilan birga ishlash;

NAZARIY BO'LIM

MySQL ga ma'lumot qo'shishni rejalashtrish SQL injectionlarga qarshi juda foydali uslubdir.

MySQL ga ma'lumot qo'shishni rejalasrflnsh jumlasining ma'nosi quyidagicha. Dastlab SQL so'rov yoziladi. Bunda qiymatlar o'rniga "?" belgi si qoyiladi. Masalan, INSERT INTO MyGuests VALUES (?, ?, ?). Keyin shu SQL so'rovi shabloni asosida istalgancha ma'lumot qo'shish mumkin. Bunda taqat parametrlar yuboriladi.

MySQL ga ma'lumot qo'shishni rejalashtirish

```
<?php
        $dbhost  = 'localhost';    // o'zgartirish shart emas
        $dbname  = 'mydb';              // Ma'lumotlar bazasi nomi
        $dbuser  = 'root';              // user nomi
        $dbpass  = '';                  // parol

        // server bilan aloqa o'rnatish
        $connection = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

        if ($connection->connect_error) {
           die("MySQLga ulanishda xatolik sodir bo`ldi: ".
                $connection->connect_error);
        }
        else
                echo "MySQL bilan aloqa o`rnatildi<br>";

        // ma'lumot qo'shishni rejalashtirish
        $tmp = $connection->prepare("INSERT INTO MyGuests (firstname,
lastname, email) VALUES (?, ?, ?)");
        $tmp->bind_param("sss", $firstname, $lastname, $email);

        // parametrlarni o'rnatish va ishga tushirish
        $firstname = "John";
        $lastname = "Doe";
        $email = "john@example.com";
```

```
$tmp->execute();

$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$tmp->execute();

$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$tmp->execute();

echo "Yangi ma'lumotlar qo'shildi";

$tmp->close();
// aloqani uzish
$connection->close();

?>
```

Bu quyidagicha tushuntiriladi.

"INSERT INTO MyGuests (firstname, lastrame, email) VALUES (?, ?)"

SQL so'rovicla qiymatlar o'miga "?" belgisini qo'yamiz. Bu belgi o'rnida butun Yoki haqiqiy son, satr Yoki blob qiymat bo'lishi mumkin. bind_param() funksiyasida qanday qiymat qo'shmoqchi bolsa shunga mos bo'lgan tur tanlaniladi.

1. i-integer
2. d-double
3. s - string
4. b-BLOB

Yuqoridagi misol uchun quyidagicha bo'ladi. $tmp->bind_param{"sss", ffirstname, flastnamej Jemail);

5. C hunki biz 3 ta satrii qiymat qo'shyapm iz. S huning uchun "sss" yozi Id i. Agar emai I o'mi ga tug'i Igan yili yozilishi lozim bo'lganida "ssi" yozilar edi. Chunki tug'ilgan yil butun sondan iborat bo'ladi.
6. S hunday qi li b biz S QL injecti on tahd idi ni ka mayti ram iz.
7. Eslatma: Ma'lumotlar bazasiga tashqaridan (masalan foydalanuvchilartian) nimadir qo'shish lozim bolsa, uni albatta tekshirish va tozalash lozim bo'ladi.

Ushbu bo’lim PHP va MySql MBBS (SUBD) o’rtasidagi hamkorlik usullari bilan tanishishga mo’ljallangan. Asosiy e’tibor ma’lumotlar bazasi bilan

bog'lanishni o'rnatish, so'rovlar jo'natish funksiyalari va javoblarni (mysql_connect, mysql_query, mysql_result, mysql_num_rows, mysql_close) qayta ishlashga qaratiladi. Misol sifatida virtual tarix muzeyi ma'lumotlar bazasining ma'muriyati uchun web-interfeys tuzish masalasini ko'raylik. PHP distributiviga MySQL ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan funksiyalarni oluvchi kengaytma kiradi. Bu bo'limda MySQL bilan ishlash uchun ba'zi bir ma'lumotlar bazasini tasvirlash va to'ldirish maqsadida web-interfeyslarni tuzish topshirig'ini yechish uchun kerak bo'ladigan asosiy funksiyalar bilan tanishamiz. Savol tug'iladi: bunday interfeyslarni tuzish nega kerak? SQL so'rovlar tili bilan notanish odamlar ma'lumotni ma'lumotlar bazasiga kiritish va uning tarkibini ko'rib turish imkoniyati bo'lishi uchun shunday qilinadi. Ma'lumotlar bazasiga ma'lumotlarni qo'shish uchun web-interfeys bilan ishlashda bu ma'lumotlarni shunchaki html-formaga kiritish va ularni serverga jo'natish kerak bo'ladi, bizning skript esa qolgan barcha amallarni bajaradi. Jadval tarkibini ko'rib turish uchun havola ustiga bir marta bosish va kerakli sahifaga kirish kifoya.

Ko'rinib turishi uchun bu interfeyslarni virtual muzey eksponatlari haqidagi ma'lumotlar joylanadigan Artifacts jadvallari uchun tuzamiz. Avvalgi bo'limda bu kolleksiyaga strukturani hamda uning shaxs (Persons) va tasvirlar (Images) tavsiflari kolleksiyalari bilan aloqasini kiritgan edik. Artifacts kolleksiyasidagi har bir eksponat quyidagi xarakteristika yordamida tasvirlanishini eslatib o'tamiz:

nom (title);
muallif (author);
ifoda (description);
o'rindosh nom (alternative);
tasvir (photo).

Nomlanish va o'rindosh nomlanish uzunasiga 255 belgidan kam satr (ya'ni VARCHAR(255)), tasvirlash – matnli maydon (TEXT turiga mansub) hisoblanadi, "muallif" va "tasvir" maydonlarida esa Persons kolleksiyasidan muallifning identifikatorlari va Images kolleksiyasidan eksponat tasvirlariga muvofiq mavjud bo'ladi.

Ma'lumotni qo'shish uchun interfeys tuzish

Demak, bizda ma'lumotlar bazasida biron-bir jadval bor. Bu jadvalga ma'lumotni qo'shishga mo'ljallangan interfeys tuzish uchun uning strukturasini (ya'ni uning maydonlari jamlanmasini) html-formada tasvirlash kerak bo'ladi.

Bu topshiriqni quyidagi tarkibiy topshiriqlarga bo'lib chiqamiz:

- MB bilan ulanishni o'rnatish;
- MB ishini tanlash;
- Jadval maydonlari ro'yxatini olish;
- html-formada maydonlarni tasvirlash.

Bundan keyin formaga kiritilgan ma'lumotlarni ma'lumotlar bazasiga kiritish kerak bo'ladi.

Bu topshiriqlarning barchasini tartib bilan ko'rib chiqamiz.

Aloqa o'rnatish

Demak, birinchi galdagi vazifa – bu ma’lumotlar bazasi bilan aloqa o’rnatish. **mysql_connect** funksiyasidan foydalanamiz. mysql_connect sintaksisi

mysql_connect resursi ([server qatori[, username qatori[, password qatori[, mantiqiy **new_link**[, butun client_flags]]]]])

Bu funksiya **MySQL** serveri bilan aloqa o’rnatadi va bu aloqaga ko’rsatkich qaytaradi Yoki muvaffaqiyatsiz chiqqanda FALSE ko’rsatadi. Yetishmayotgan parametrlar uchun quyidagi ma’nolar muvaqqat qo’yiladi:

server = 'localhost:3306'

username = server jarayoni egasidan foydalanuvchi ismi

password = bo’sh parol

Agar funksiya aynan bir xil parametrlar bilan ikki marta chaqiriladigan bo’lsa, yangi ulanish o’rnatilmaydi, aksincha havola eski aloqaga qaytariladi. Bundan qochish uchun esa, har qanday holatda yangi bir aloqa ochishga majbur qiluvchi new_link parametridan foydalaniladi.

client_flags parametri – bu quyidagi konstantalar kombinasiyasi:

MYSQL_CLIENT_COMPRESS (siqish protokolidan foydalaniladi),
MYSQL_CLIENT_IGNORE_SPACE

(funksiya nomidan so’ng probel qo’yishga ruxsat beradi),

MYSQL_CLIENT_INTERACTIVE (interactive_timeout bir soniya kutish - wait_timeout – ulanish tugatilishigacha). new_link parametri PHP 4.2.0 da mavjud, client_flags esa – PHP 4.3.0 da.

Server bilan ulanish, agar u bungacha mysql_close() yordamida yopilmagan bo’lsa, skriptni amalga oshirish tugallanishida berkiladi.

SHunday qilib, “123” parolli nina foydalanuvchisi uchun lokal serverda ma’lumotlar bazasi bilan ulanishni o’rnatamiz.

```
<?
$conn = mysql_connect(
   "localhost", "nina","123")
or die("Ulanishni amalga oshirib bo’lmaydi: ". mysql_error());
echo "Ulanish amalga oshdi";
mysql_close($conn);
?>
```

mysql_connect amali

shell>mysql -u nina -p123 buyrug’i bilan teng kuchli.

Ma’lumotlar bazalarini tanlash

Ulanishni amalga oshirgandan so’ng birga ishlashiniiz kerak bo’lgan ma’lumotlar bazasini tanlashiniiz kerak bo’ladi. Bizning ma’lumotlarimiz book ma’lumotlar bazasida saqlanadi.

MySQL da ma’lumotlar bazasini tanlash use buyrug’i yordamida amalga oshiriladi:

```
mysql>use book;
```

PHP da buning uchun mysql_select_db funksiyasi mavjud

mysql_select_db: sintaksisi

```
mantiqiy mysql_select_db (
   database_name qatori
   [,link_identifier resursi])
```

Bu funksiya TRUE qiymatni ma’lumotlar bazasini muvaffaqiyatli tanlanganda qaytaradi va FALSE ni esa – aksincha bo’lganda.

Book ma’lumotlar bazasini ishga tushiramiz:

```
<?
$conn = mysql_connect(
   "localhost","nina","123")
or die("Ulanishni amalga oshirib bo’lmaydi: ". mysql_error());
echo "Ulanish amalga oshdi";
mysql_select_db("book");
?>
```

Jadval maydonlari ro’yxatini olish

Endi topshiriqni yechish bilan alohida shug’ullansa bo’ladi. Jadval maydonlari ro’yxatini qanday olish mumkin? Juda oddiy. PHP da bu holda ham o’ziga tegishli buyruq mavjud - mysql_list_fields.

mysql_list_fields sintaksisi

```
mysql_list_fields ( resursi
database_name qatori,
table_name  qatori
[, resurs link_identifier])
```

Bu funksiya maydonlar ro’yxatini database_name ma’lumotlar bazasidagi table_name jadvaliga qaytaradi. Kelib chiqadiki, ma’lumotlar bazasini tanlash bizga majburiy emas, lekin keyinroq asqotadi. Bu funksiya ishining natijasini – resurs tipi o’zgaruvchisini qanday baholash mumkin. Ya’ni bu biz olishni xohlagan narsa emas. Bu ularning nomlari, tiplari va bayroqlari kiradigan jadval maydonlari haqidagi ma’lumotlarni olish uchun foydalanish mumkin bo’lgan havola. mysql_field_name funksiyasi so’rov amalga oshirilishi natijasida olingan maydon nomini qaytaradi. mysql_field_len funksiyasi maydon uzunligini qaytaradi. mysql_field_type funksiyasi maydon tipini qaytaradi, mysql_field_flags funksiyasi esa probel bilan yozilgan maydon bayroqlari ro’yxatini qaytaradi. Maydon tiplari int, real, string, blob va b. bo’lishi mumkin. Bayroqlar not_null, primary_key, unique_key, blob, auto_increment va b. bo’lishi mumkin.

Bu barcha buyruqlar sintaksisi bir xil:

mysql_field_name (result qatori, butun field_offset) resursi;

mysql_field_type (result qatori, butun field_offset) resursi;

mysql_field_flags (result qatori, butun field_offset) resursi;

mysql_field_len (result qatori, butun field_offset)

Bu yerda result – bu so'rov natijasi identifikatori (masalan, mysql_list_fields Yoki mysql_query funksiyalar bilan jo'natilgan so'rov(bu haqda keyinroq gapirib o'tiladi)), field_offset esa – natijadagi maydonning tartib raqami. Umuman olganda, mysql_list_fields Yoki mysql_query turidagi funksiyalarni qaytaruvchi jadvalni, aniqrog'i uning unga bo'lgan ko'rsatkichini aks ettiradi. Bu jadvallardan aniq ma'nolar olish uchun maxsus bu jadvalni ostki qator sifatida o'quvchi funksiyalarni ishga tushirish kerak. Bu funksiyalarga mysql_field_name va shu kabilar ham kiradi. So'rovni amalga oshirish natijasi jadvalidagi barcha qatorlarni saralash uchun bu jadvaldagi qatorlar miqdorini bilish kerak.

mysql_num_rows(result resursi) buyrug'i result ning ko'pgina natijalari qatorlari miqdorini qaytaradi.

Endi esa Artifacts (eksponatlar kolleksiyasi) jadvali maydonlari ro'yxatini olishga urinib ko'ramiz.

```
<?
$conn = mysql_connect(
   "localhost","nina","123")
or die("Aloqa o'rnatib bo'lmaydi: ". mysql_error());
echo "Aloqa o'rnatildi";
mysql_select_db("book");
$list_f = mysql_list_fields (
  "book","Artifacts",$conn);
  $n = mysql_num_fields($list_f);
for($i=0;$i<$n; $i++){
  $type = mysql_field_type($list_f, $i);
  $name_f = mysql_field_name($list_f,$i);
  $len = mysql_field_len($list_f, $i);
  $flags_str =  mysql_field_flags (
  $list_f, $i);
echo "<br>Maydon nomi: ". $name_f;
echo "<br>Maydon turi: ". $type;
echo "<br>Maydon uzunligi: ". $len;
echo "<br>Maydon bayroqlari qatori: ".
   $flags_str . "<hr>";
}
?>
```

Natija sifatida taxminan quyidagilarni olish mumkin (albatta, jadvalda ikkita maydon bo'lganda):

- Maydon nomi: id
- Maydon turi: int
- Maydon uzunligi: 11
- Maydon bayroqlari qatori:

- not_null primary_key auto_increment
- Maydon nomi: title
- Maydon turi: string
- Maydon uzunligi: 255
- Maydon bayroqlari qatori:

html-formada maydonlar ro'yxatining aks etishi

Endi biroz yuqoridagi misolni tekshirib ko'ramiz. Maydon haqidagi ma'lumotni shunchaki chiqarmaymiz, uni munosib html-forma elementida aks ettiramiz. BLOB turidagi elementlarni textarea ga o'tkazamiz (TEXT turida biz tuzgan description maydoni BLOB turiga ega ekanligini e'tibordan qochirmaymiz), raqamlar va qatorlarni <input type=text> kirish qismi matnli qatorlarida aks ettiramiz, avtoinkrement belgisiga ega elementni esa umuman aks ettirmaymiz, chunki uning ma'nosi o'z-o'zidan o'rnatiladi.

Bularning barchasi anchayin oddiy hal qilinadi, bayroqlar ro'yxatidaig auto_increment bayrog'i bundan mustasno. Buning uchun explode funksiyasidan foydalaniladi:

- explode: sintaksisi
- explode massivi (separator qatori,
- string qatori [, int limit])

Bu funksiya string qatorini separator taqsinilagichi yordamida qismlarga bo'ladi va olingan qatorlar massivini qaytaradi.

Bizning holatda taqsinilagich sifatida probel " " ni olish kera, bo'lish uchun boshlang'ich qator sifatida esa – maydon bayroqlari qatorini.

SHunday qilib, ma'lumotlarni Artifacts jadvaliga kiritish uchun forma tuzamiz:

Misol. Artifacts jadvaliga ma'lumot kiritish uchun forma

```
<?
$conn=mysql_connect("localhost","nina","123");
        // aloqa o'rnatamiz
$database = "book";
$table_name = "Artifacts";
mysql_select_db($database); // ish uchun
        // ma'lumotlar bazasi tanlaymiz
$list_f = mysql_list_fields($database,$table_name);
 // bazadan maydonlar ro'yxatini olarniz
$n = mysql_num_fields($list_f); // avvalgi so'rovdagi natijadagi qatorlar soni (Artifacts jadvalida ja'mi qancha maydonlar bor)
echo "<form method=post action=insert.php>";
// ma'lumotlarni kiritish uchun forma tuzamiz
echo " <TABLE BORDER=0 CELLSPACING=0 width=50% ><tr>
   <TD  BGCOLOR='#005533' align=center><font color='#FFFFFF'>
   <b> Add new row in $table_name</b></font></td></tr><tr><td></td></tr></TABLE>";
```

```
echo "<table border=0 CELLSPACING=1 cellpadding=0 width=50% >";
// har bir maydon uchun uning nomi, turi, uzunligi va bayrog'ini olarniz
for($i=0;$i<$n; $i++)
{    $type = mysql_field_type($list_f, $i);
    $name_f = mysql_field_name ($list_f,$i);
    $len = mysql_field_len($list_f, $i);
    $flags_str = mysql_field_flags ($list_f, $i);
    // bayroqlar qatoridan massiv qilarniz,
    // unda massivning har bir elementi – maydon bayrog'i
    $flags = explode(" ", $flags_str);
    foreach ($flags as $f){
        if ($f == 'auto_increment') $key = $name_f;
            // avtoinkrement nomini eslatib o'tamiz
    }
/* avtoinkrement hisoblangan har bir maydon uchun, uning turidan kelib chiqib mos keluvchi forma elementini chiqaramiz*/
if ($key <> $name_f){
echo "<tr><td align=right bgcolor='#C2E3B6'><font size=2>
        <b> ". $name_f ."</b></font></td>";
switch ($type){
        case "string":
            $w = $len/5;
            echo "<td><input type=text name=\"$name_f\"
                size = $w ></td>";
        break;
        case "int":
            $w =  $len/4;
            echo "<td><input type=text name=\"$name_f\"
                size = $w ></td>";
        break;
        case "blob":
            echo "<td><textarea rows=6 cols=60 name=\"$name_f\"></textarea></td>";
        break;
        }
    }
    echo "</tr>";
}
echo "</table>";
echo "<input type=submit name='add' value='Add'>";
echo "</form>";
?>
```

Ma'lumotlar bazasiga ma'lumotlar yozish

SHunday qilib forma tuzildi. Endi eng asosiysini bajarish qoldi – bu formadagi ma'lumotlarni bizning ma'lumotlar bazasiga jo'natish. Ma'lumki,

ma’lumotlarni jadvalga yozish uchun SQL tilidagi INSERT buyrug’i ishlatiladi. Masalan:

```
mysql> INSERT INTO Artifacts SET title='Kamolov';
```

PHP skriptda bunday buyruqdan (Yoki SQL dagi istalgan buyruqdan) qanday foydalaniladi, degan savol tug’iladi. Buning uchun mysql_query() funksiyasi mavjud.

mysql_query sintaksisi

mysql_query resursi (query qatori
   [, resurs link_identifier])

mysql_query() SQL-so’rovni MySQL ma’lumotlar bazasining link_identifier ko’rsatkichi yordamida aniqlanadigan aktiv ma’lumotlar bazasiga jo’natadi (bu MySQL serveri bilan biron-bir aloqaga havola). Agar link_identifier parametri o’tkazib yuborilgan bo’lsa, so’nggi ochiq aloqa ishlatiladi. Agar ochiq aloqa bo’lmasa, funksiya parametrsiz mysql_connect() funksiyasiga o’xshash holda MBBT (SUBD) bilan bog’lanishga urinadi.

So’rov natijasi buferlanadi.

Object (obyektlar) tipi

Obyektlar – obyektga yo’naltirilgan dasturlashdan kirib kelgan *ma’lumot tipi*dir. Obyektga yo’naltirilgan dasturlash tamoyiliga ko’ra, sinf – aniq xossalarga ega va ular bilan ishlaydigan metodli obyektlar to’plarni. Obyekt esa mos ravishda sinf nusxasidir. Masalan, dasturchilar – bu dasturni tuzuvchi, kompyuter adabiyotlarini o’rganadigan odamlar sinfi va bundan tashqari hamma odamlar qatori ism va familiyasi mavjud. Endi agarda biror aniq dasturchi – Azamat Boboyevni olsak, u holda uni shu xossaga ega bo’lgan dasturchi sinfini obyekti sifatida qarash mumkin va u ham dastur tuzadi, hamda ismi mavjud va boshqalar.

PHP dasturlash tilida *obyekt metodi*ga murojaat -> *amali*dan foydalaniladi. Obyektni inisializasiya qilishda *obyektni o’zgaruvchan nusxasi*ni yaratadigan new ifodasidan foydalaniladi.

Misol. PHP dasturlash tilida obyektlar.

```
<?php
class Person {      // PHP dasturlash tilini o’rganadigan odam metodi
   function know_php()
      {
         echo "Endi men PHP dasturlash tilini bilaman!";
      }
}
$bob = new Person; // odam sinfini obyektini yaratamiz.
$bob -> know_php(); // uni PHP tiliga o’rgatamiz.
?>
```

Resource (resurslar) tipi

Resurs – bu tashqi resursga (masalan, ma’lumotlar bazasi bilan bog’lanish) ссылка orqali bog’langan maxsus o’zgaruvchidir. Resurslar maxsus funksiyalar

(masalan, mysql_connect(), pdf_new() va shunga o'xshashlar) yordamida yaratiladi va foydalaniladi.

Null tipi

Maxsus *NULL* qiymati *o'zgaruvchi*ni qiymatga ega emasligi haqida ogohlantiradi.

O'zgaruvchi NULL qiymat qabul qiladi, agarda:

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

*NULL tip*li faqat bitta qiymati mavjud – registrga sezgirligi kichik NULL kalit so'zidir.

Masalaning yechilishi

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

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

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

7-Amaliy mashg'ulot

Mavzu: **MySQL ga bir nechta ma'lumot qo'shish**

Ishning maqsadi: PHP da ma'lumotlar bazasi bilan ishlashni o'rganish

Talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak: PHP da bir MySQL ma'lmotlar bazasiga necha ma'lumotlarni qo'shishni o'rganish

Qila olishi kerak: PHPda MySql bilan birga ishlash;

Quyidagi tartbda MySQL ga bir nechta malumori qo'shishimiz mumkin. Video ma'ruzada batafsil to'xtalamiz.

MySQL ga bir nechta ma'lumot qo'shish

```
<?php
	$dbhost  = 'localhost';    // o'zgartirish shart emas
	$dbname  = 'mydb';               // Ma'lumotlar bazasi nomi
	$dbuser  = 'root';               // user nomi
	$dbpass  = '';                   // parol

	// server bilan aloqa o'rnatish
	$connection = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

	if ($connection->connect_error) {
	   die("MySQLga ulanishda xatolik sodir bo`ldi: ".
	        $connection->connect_error);
	}
	else
	        echo "MySQL bilan aloqa o`rnatildi<br>";

	// MyGuests jadvaliga ma'lumot qo'shish
	$sorov = "INSERT INTO MyGuests (firstname, lastname, email)
	                VALUES ('John', 'Doe', 'john@example.com');";
	$sorov .= "INSERT INTO MyGuests (firstname, lastname, email)
	                VALUES ('Mary', 'Moe', 'mary@example.com');";
	$sorov .= "INSERT INTO MyGuests (firstname, lastname, email)
	                VALUES ('Julie', 'Dooley', 'julie@example.com')";

	if ($connection->multi_query($sorov) === TRUE) {
	        $last_id = $connection->insert_id;
	   echo "MyGuests jadvaliga ma`lumotlar qo`shildi.";
	} else {
	   echo "Jadvalga ma`lumotlarni qo`shishda xatolik: " .
	   $conn->error;
```

```
    }
    // aloqani uzish
    $connection->close();

?>
```

2.3. LABORATORIYa MASHG'ULOTLARI MATERIALLARI
1-Laboratoriya ishi

Mavzu: Shart operatori

NAZARIY BO'LIM

Ishning maqsadi: PHP da shart operatorlarini o'rganish va qo'llash
Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:
Bilishi kerak – PHP da shart operatorlarini, yani if, else, elseif;

Shart operatori shakllari

1. if (mantiqiy ifoda) - shart operatorining bu ko'rinishi mantiqiy ifoda rost bo'lgan holda qandaydir kod bajarilishi uchun ishlatiladi
2. if (mantiqiy ifoda) ... else - shart operatorining bu ko'rinishida mantiqiy ifoda rost bo'lsa bir kod yolg'on bo'lganida boshqa bir kod bajarilishi uchun ishltiladi
3. if (mantiqiy ifoda) ... elseif mantiqiy ifoda) else - shart operatorining bu ko'rinishida oldingi shart yolg'on bo'lganda keyingi shart tekshiriladi, va hokazo.

if operatoriga misol

```
<?php
	$a = 5;
	$b = 3;
	$c = 4;

	if ($a < $b + $c &&
		$b < $a + $c &&
		$c < $b + $a)
	{
		echo "Uchburchak qurish mumkin";
	} else {
		echo "Uchburchak qurish mumkin EMAS";
	}
```

If shartli operatorga misollar

```
<?php
	$baho = 5;

	if ($baho == 5){
		echo 'Besh baho';
	}

 ?>
```

Bror sonning juft toqligini aniqlash

```php
<?php

	$a = 11; // ixtiyoriy butun son yozish mumkin

	if ($a % 2 == 0)
	{
		echo $a." juft son";
	}
	else
	{
		echo $a." toq son";
	}
?>
<?php
	$a = 5;
	$b = 0;

	if ($a % 2 == 0 &&
		$a == 0 &&
		$b % 2 == 0 &&
		$b == 0) {
		echo "Har ikkalasi ham juft";
	} elseif ($a % 2 != 0 &&
			$b % 2 != 0)
	{
		echo "Har ikkalasi ham toq";
	} elseif ($a % 2 == 1 ||
			$b % 2 == 1)
	{
		echo "Kamida bittasi toq";
	}

 ?>
```

Else operatori

```php
<?php
	$a = 5;
	$b = 3;
	$c = 4;

	if ($a < $b + $c &&
		$b < $a + $c &&
		$c < $b + $a)
	{
		echo "Uchburchak qurish mumkin";
```

```
    } else {
        echo "Uchburchak qurish mumkin EMAS";
    }

?>
```

Sonning musbat Yoki manfiyligini anqlash

```
<?php
    // if - elseif operatoridan foydalanish
    // $a   $a = 5;

    if ($a > 0) {
        echo $a." musbat son";
    } elseif ($a < 0) {
        echo $a." manfiy son";
    } else {
        echo $a." nolga teng";
    }

?>
```

Mantiqiy amallar

! (inkor qilish) - mantiqiy operatori mantiqiy ifodalar Yoki o'zgaruvchilar oldidan qo'yiladi. Mantiqiy ifoda Yoki o'zgaruvchining qiymatini teskarisiga o'zgartiradi.

<table><tr><td>X</td><td>Y</td><td>X</td><td>X||Y</td></tr></table>

&& (Mantiqiy ko'paytirish) - mantiqiy operatori ikkita mantiqiy o’zgaruvchini birlashtiradi. Agar ikkala o'zgaruvchi ham rost qiymatga ega bo'lsa natija rost, aks holda yolg'on natija beradi.

|| (mantiqiy qo'shish) - mantiqiy operatori ikkita mantiqiy o’zgaruvchini birlashtiradi. Agar o'zgaruvchilardan kamida bittasi rost qiymatga ega bo'lsa natija rost, aks holda yolg'on natija beradi.

! - mantiqiy inkor operatori jadvali

X	!X
fal	true
tru	false

&&, || mantiqiy operatorlari jadvali

false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	True

Munosabat amallari

`==` - teng

`!=` - teng emas

`<` - kichik

`>` - katta

`<=` - kichik yoki teng

`>=` - katta yoki teng

2-Laboratoriya ishi

Mavzu: $_GET o'zgaruvchilari

NAZARIY BO'LIM

Ishning maqsadi: PHP da $_GET o'zgaruvchilarini o'rganish va qo'llash

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – PHP da $_GET metodini qo'llashni;

isset funksiyasi

bool isset { mixed, $var mixed…])

isset funksiyasi biror o'zgaruvchini bor - yo'qligini aniqlash uchun ishlatiladi. Agar o'zgaruvchi mavjud bo'lsa, **true** aks holda **false** qiymat qaytaradi. Agar o'zgaruvchi **unset()** funksiyasi orqali o'chirilgan bo'lsa ham **false** qiymat qaytaradi. Agar parametr slfatida bir nechta qiymat berilgan bo'lsa, ularning barchasi mavjud bo'lsa **true** qiymat qaytariladi.

isset funksiyasiga misol

```
<?php
	// isset funksiyasi o'zgaruvchining bor Yoki
	// yo'qligini aniqlash uchun ishlatiladi
	$k = 5;

	if (isset($k)) {
		echo '$k o`zgaruvchisi mavjud <br>';
	}
	else {
		echo '$k o`zgaruvchisi mavjud EMAS  <br>';
	}

	if (isset($t)) {
		echo '$t o`zgaruvchisi mavjud';
	}
	else {
		echo '$t o`zgaruvchisi mavjud EMAS';
	}
?>
```

Natijasi:

$k o'zgaruvchisi mavjud

$t o'zgaruvchisi mavjud EMAS

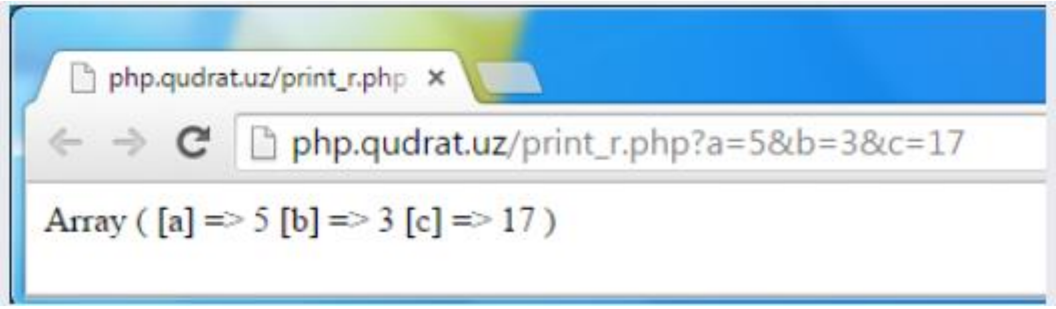
$_GET o'zgaruvchilari

$_GET superglobal o'zgaruvchi hisoblanadi. $_GET orqali foydalanuvchidan ma'lumotlarni qabul qilib olishimiz mumkin. $_GET va $_POST orqali o'zgaruvchilar massiv shaklida qabul qilinadi.

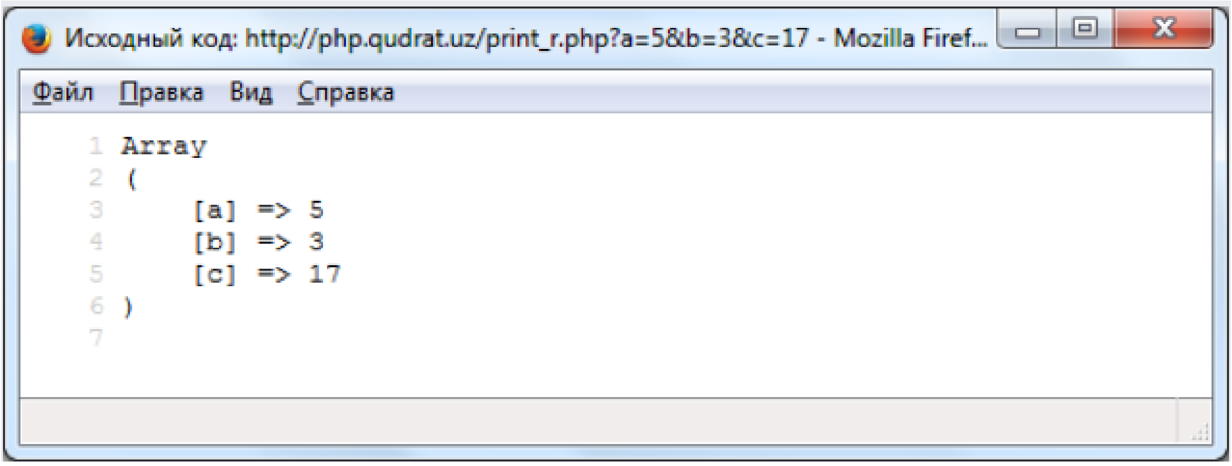
$_GET orqali yuborilgan o'zgaruvchilarni chop qilish

```
<?php
    // print_r funksiyasi massivni o'qishga qulay shaklda chop qiladi
    print_r($_GET);
?>
```

Yuqoridagi kodni ***print_r.php*** fayliga saqlang va browser orqali chaqiring. Faqat
Array() yozuvi chiqadi. Biror o'zgaruvchiga qiymat berish uchun adreslar satrida ***print_r.php?&a=5,&b=3,&c=17*** shaklida yozing va enterni bosing. O'zgaruvchilarni bir biridan ajratish uchun **&** belgisi ishlatiladi



Browserlar **print_r** orqali chiqarilgan ma'lumotlarni chiroyli chiqara olmaydi. Shuning uchun sahifaning kodini ochib korishingiz mumkin. Quyidagicha shaklda chiqariladi.



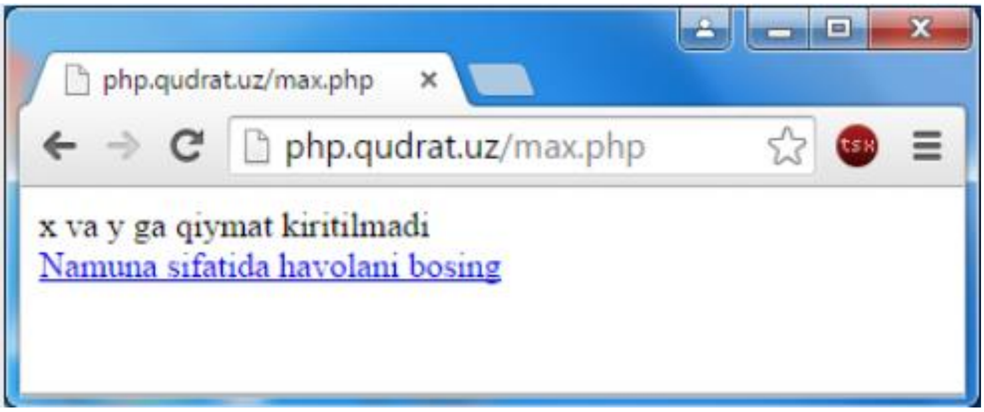
Endi biror o'zgaruvchini $_GET orqali yuborilgan Yoki yuborilmaganligini tekshiradigan script yozamiz.

```
<?php
	// isset funksiyasi o'zgaruvchining bor Yoki
	// yo’qligini aniqlash uchun ishlatiladi

	if (isset($_GET['a']))
	{
		$a = $_GET['a'];
		echo $a.' kiritildi <br>';
	}
	else
	{
		echo ' a o`zgaruvchisiga qiymat berilmadi<br>';
	}
			// chop qilishning yana bir KO’RINISHI
	print_r($_GET);

?>
```

isset, $_GET va if operatoridan birgalikda foydalanish

```
<?php
	// isset va $_GET ga oid misol
	// shart operatoridan foydalanishni o'rganish

	if (isset($_GET['x'], $_GET['y']) ) {
		$x = $_GET['x'];
		$y = $_GET['y'];

		if ($x > $y) {
			echo $x;
		} else {
			echo $y;
		}
	} else {
		echo 'x va y ga qiymat kiritilmadi <br>
			<a href="?x=5&y=6"> Namuna sifatida havolani
bosing</a>';
	}
?>
```

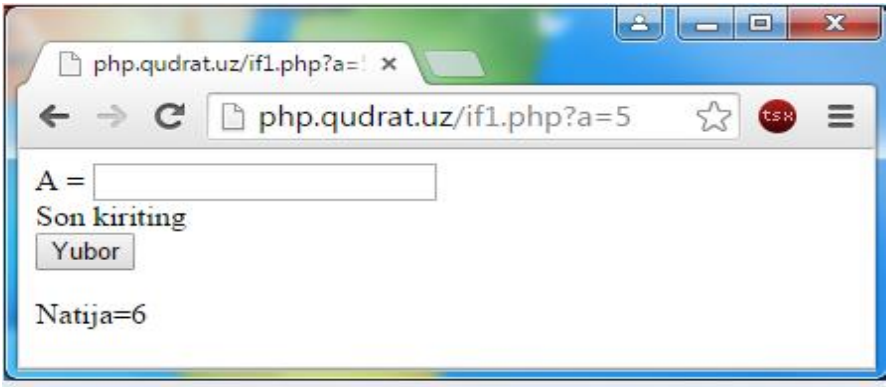
Yuqoridagi kodni ishga tushirsak quyidagi natija chiqadi.



HTML Forma orqali ma'lumot yuborish

Maqsad if operatorni yaxshilab o'rganish. if1 - masalasi. KiritiIgan son musbat bolsa, birga oshiruvchi aks holda o'zgarishsiz qoldiruvchi dastur tuzish. Quyidagi kodni if1.php nomi bilan saqlang.

```
<html>
<body>
<form action="if1.php" method="get">
        A = <input type="text" name="a"><br>
        Son kiriting<br>
        <input type="submit" value="Yubor"> <br>
</form>

        <?php
                if (isset($_GET['a'])) {
                        $a = $_GET['a'];

                        if ($a > 0) {
                                $a++;
                        }
                        // is_numeric - funksiyasi o'zgaruvchini son Yoki
                        // son emasligini aniqlash uchun ishlatiladi
                        if (is_numeric($a)) {
                                echo "Natija=".$a;
                        } else {
                                echo 'Siz son kiritmadingiz';
                        }
                }
        ?>
</body>
</html>
```



Shart operatoriga oid masalalar

if1 Butun son berilgan. Agar, berilgan son musbat bo'lsa, 1 ga oshirilsin, aks holda o'zgartirilmasin. Hosil bo'lgan sonni ekranga chiqaruvchi programma tuzilsin.

if2 Butun son berilgan. Agar, berilgan son musbat bo'lsa. 1 ga oshirlng, aks holda 2 ga kamaytiring. Hosil bo'lgan sonni ekranga chiqaruvchi programma tuzilsin.

if3. Butun son berilgan. Agar, berilgan son musbat bo'lsa, 1 ga oshiring, agar manfiy bo'lsa 2 ga kamaytiring. Agar 0 ga teng bo'lsa, 1 0 ni o'zlashtirsin. Hosil bo'lgan sonni ekranga chiqaruvchi programma tuzilsin.

if4. Uchta butun son berilgan. Shu sonlar orasidan nechta musbat son borligini aniqlovchi programma tuzilsin.

if5. Uchta butun son berilgan. Shu sonlar orasidan nechta musbat va manfiy son borligini aniqlovchi programma tuzilsin.

if6. Ikkita butun son berilgan. Shu sonlarning kattasini aniqlovchi programma tuzilsin.

if7. Ikkita butun son berilgan. Shu sonlarning kichigini tartib raqamini aniqlovchi programma tuzilsin.

if8. Ikkita butun son berilgan. Shu sonlarning avval kattasini keyin kichigini ekranga chiqaruvchi programma tuzilsin.

if9. A va B haqlqiy sonlari berilgan. Shu sonlarni shunday o'zgartirish kerakki. A son kichik B son katta bo'lsin. A va B ning qiymati ekranga chiqarilsin.

if10. A va B butun sonlari berilgan. Agar o'zgaruvchilar o'zaro teng bo'lmasa, A va B o'zgaruvchilah ularning yig’indisini o'zlashtirsin. Agar teng bo'lsa, 0ni o'zlashtirsin. A va B ning qiymati ekranga chiqarilsin.

if11. A va B butun sonlari berilgan. Agar o'zgaruvchilar o'zaro teng bo'lmasa, A va B bu sonlarning kattasini o'zlashtirsin. Agar teng bo'lsa, 0ni o'zlashtirsin. A va B ning qiymati ekranga chiqarilsin.

if12. Uchta son berilgan. Shu sonlarni kichigini aniqlovchi programma tuzilsin.

if13. Uchta son berilgan. Shu sonlarni o'ratachasi (ya’ni katta va kichik sonlar orasldagi son)ni aniqlovchi programma tuzilsin.

if14. Uchta son berilgan. Shu sonlarni avval kichigini keyin kattasini ekranga chiqaruvchi programma tuzilsin.

if15. Uchta son berilgan. Shu sonlarning yig'indlsl eng katta bo'ladigan ikkitasini ekranga chiqaruvchi programma tuzilsin.

if16. A, B, C haqiqiy sonlari berilgan. Agar berilgan sonlar o'sish tartibida berilgan bo'lsa. sonlarni ikkilantiring. aks holda sonlarni ishorasi o'zgartirilsin. A, B, C ning qiymatlari ekranga chiqarilsin.

if17. A, B, C haqiqiy sonlari berilgan. Agar berilgan sonlar o'sish Yoki kamayish tartibida berilgan bo'lsa. sonlarni ikkilantiring. aks holda sonlarni ishorasi o'zgartirilsin. A, B. C ning qiymatlari ekranga chiqarilsin.

if18. Uchta butun son berilgan. Shu sonlarni Ikkitasl o'zaro teng. qolgan bittasini tartib raqami aniqlansin.

if19. To'rtta butun son berilgan. Shu sonlarni uchtasi o'zaro teng. qolgan bittasini tartib raqami aniqlansin.

if20. Sonlaro'qida uchta A, B, C nuqtalarberilgan. A nuqtaga eng yaqin nuqta va ular orasidagi masofa topilsin.

if21. Koordinatalar tekisligida butun son berilgan. Agar nuqta koordinata boshida yotsa, 0 chiqarilsin. Agar nuqta OX Yoki OY o'qlarida joylashsa mos holda 1 va 2 chiqarilsin. Agar nuqta koordinata o'qida joylashmasa 3 chiqarilsin.

if22. OX va OY koordinata o'qlarida yotmaydigan nuqta berilgan. Nuqta joylashgan koordinata choragi aniqlansin.

if23 Koordinata o'qlariga parallel ravishda to'g'ri to'rtburchakning uchta uchi berilgan, to'rtinchi uchi koordinatasi aniqlansin.

if24. X haqiqiy soni berilgan. Quyidagi funksiya hisoblansin.

if25. Yil benlgan (rnusbat butun son). Benlgan yilda nechta kun borligini aniqlovchi programma tuzilsin. Kabisa yilida 366 kun bor, kabisa bo'lmagan yilda 365 kun bor. Kabisa yil deb 4 ga karrali yillarga aytiladi. Lekin 100 ga karrali yillar ichida faqat 400 ga karrali bo'lganlari kabisa yil hisoblanadi. Masalan 30 D, 1300 va 1900 kabisa yili emas. 1200 va 2000 kabisa yili.

if26. Butun son berilgan. Berilgan sonni inusbat toq son", ¦'manfiy juft son^{N}, ¦'son nolga teng^{N} va x.k. ekranga yozadigan programma tuzilsin.

If27. 1 -999 oraliqdagi sonlar benlgan. Berilgan sonni "ikki xonali juft son"", "uch xonali toq son" va x.k. ekranga yozadigan programma tuzilsin.

3-Laboratoriya ishi

Mavzu: PHP da satrlar bilan ishlash

NAZARIY BO’LIM

Ishning maqsadi: PHP da satrlar bilan ishlash o’rganish va qo’llash

Ishning natijasida talaba quyidagilarga ega bo’lishi kerak:

Bilishi kerak – PHP da String toifasidagi funksiyalarni o’rganish;

String - satrlar bilan ishlash toifasi. Satr deb ixtiyoriy belgilar ketma - ketligiga aytiladi. Satrni apostrof Yoki qo'shtirnoq orasida yozish mumkin.

PHP 5 da satrlar bilan ishlovchi funksiyalarning jami soni 100 ga yaqin. Satrlar bilan ishlovchi funksiyalar bilan mustaqil ravishda tanishib chiqing. Biz esa quyidagilar bilan kifoyalanamiz.

 # Satrlar bilan ishlovchi funksiyalar

strlen - Satr uzunligini anlqlash uchun ishlatiladi. Satrdagi belgilar sonini qaytaradi.

str_word_count - Satrdagi so'zlar sonini aniqlash uchun ishlatiladi. Satrdagi so'zlar sonini qaytaradi.

strrev - Satrni teskarisiga o'girish.

strpos - Satrdan so’zni Yoki biror Satrni qidirish. Agar qidirilayotgan satr bor bo'lsa joylashgan o’rnini qaytariladi aks xolda false qaytadi.

str_replace - Satrni amashtirish. str_replace(search. replace, subject);

stripslashes -teskari slashni o'chirish uchun ishlatiladi. *magic_quotes_gpc* yoqilgan bo'lishi lozim.

trim – satr boshidagi va oxiridagi probel va boshqa belgllarni o'chirish uchun ishlatiladi

string trim (string $str[, string $character_ma5k = " \t\n\r\0\x0B ']);

" " (ASCII 32(0x20)), oddiy probel.

“\t” (ASCII 9 (0x09)), tabulyatsiya belgisi.

“\n” (ASCII 10 (OxOA)), keyingi satrga otish belgisi.

“\0” (ASCII 0 (0x00)), NUL-bayt

'***“1x0B” (ASCII 11 (0x0B)), vertikal tabulyatsiya.***

htmlspecialchars - maxsus belgilarni HTML kodiga o'girish uchun ishlatiladi.

string htmlspecialchars { string string [, int quote_style [, string charset]]);

Yozish majburiy bo'lmagan quote_styie xususiyati 3 xil qiymat qabul qiladi.

ENTCOMPAT - ikkitalik qo'shtirnoqm HTP*1L kodiga o'giradi. Odatda shu qiymat o'rnatilgi bo'ladi.

ENT_QUOTE3 -harikkala (" ", ' ') qo'shtirnoqm ham HTML kodiga o'giradi.

ENT_NOQUOTES - har ikkalasi ham o'zgarishsiz qoldiriladi. Bu funksiya orqali faqat quyidagi belgilar HTML kodiga o'giriladi:

& Ampersandm **&** kodiga o'giradi.

“” ikkitalik qo'shtirnoqm &**quot;** kodiga o'giradi. AgarENT_NOQUOTES o'rnatilmagan bo'lsa.

‘’’ apostrofm **&039;** kodiga o'giradi. Agar ENT_QUOTES o'rnatilgan bo'lsa.

strlen, str_word_count, strrev funksiyalaridan foydalanish

```
<?php
	$k = "Assalomu alaykum bo'lajak dasturchilar";
	echo $k.'<br>';
	echo 'Satrdagi belgilar soni='.strlen($k).'<br>';
	echo 'Satrdagi so`zlar soni='.str_word_count($k).'<br>';
	echo 'Telefon so`zining teskarisi='.strrev('Telefon');

?>
```

Natijasi:

Assalomu alaykum bo'lajak dasturchilar

Satrdagi belgilar soni=38

Satrdagi so'zlar soni=5

Telefon so'zining teskarisi=nofeleT

strpos, str_replace funksiyalariga misol

```
<?php
	// satrdan qidirish va almashtirish
	$k = 'Assalomu alaykum bo`lajak dasturchilar';
	$d = 'dasturchi';

	echo $k.'<br>';
	echo 'Satrdagi belgilar soni='.strlen($k).'<br>';
	echo "dasturchi so`zi boshlangan indeks=".strpos($k, $d).'<br>';

	// bo`lajak so'zim professional so'ziga almashtirish
	$k = str_replace('bo`lajak', 'professional', $k);
	echo $k;

?>
```

Natijasi:

Assalomu alaykum bo'lajak dasturchilar

Satrdagi belgilar soni=38

dasturchi so'zi boshlangan indeks=26

Assalomu alaykum professional dasturchilar

Stripslashes funksiysining qo'llamlishi

```
<?php
	$str = "Assalomu alaykum bo\'lajak dasturchilar!";
	// teskari slashni '\' olib tashlash
	echo $str.'<br>';
```

echo stripslashes($str);

?>

Natijasi:

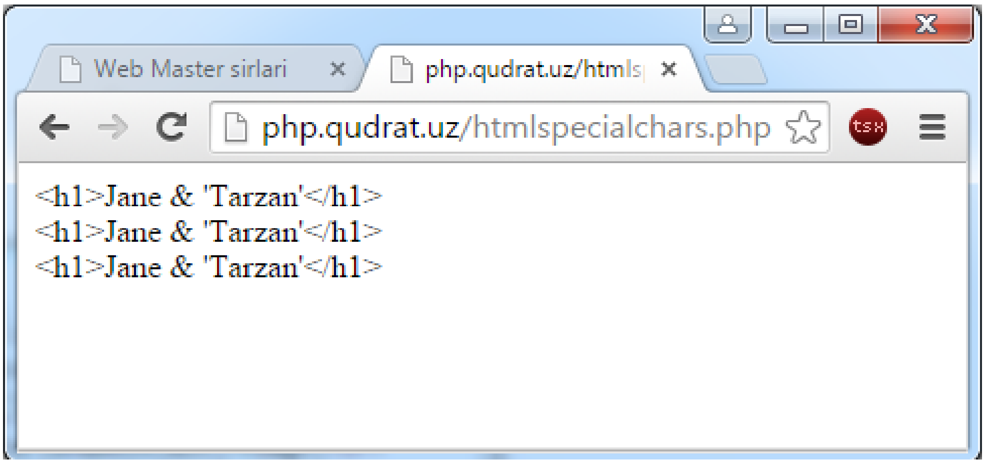
Assalomu alaykum bo\’lajak dasturchilar!
Assalomu alaykum bo'lajak dasturchilar!

Trim funksiyasi

```
<?php
        $text   = "\t\t Bu yerda bir nechta so`z bor:) ...  ";
        $binary = "\x09Satrga namuna\x0A";
        $hello  = "Hello world";

        var_dump($text);   echo '<br>';
        var_dump($binary);        echo '<br>';
        var_dump($hello); echo '<br>';

        echo "<br>";

        $trimmed = trim($text);
        var_dump($trimmed);     echo '<br>';

        $trimmed = trim($text, " \t.");
        var_dump($trimmed);     echo '<br>';

        $trimmed = trim($hello, "Hdle");
        var_dump($trimmed);     echo '<br>';

        // $binary satri boshi va oxiridagi
        // boshqaruvchi ASCII-belgilarim olib tashlAyniiz
        // ([0 dan 31] gacha)
        $clean = trim($binary, "\x00..\x1F");
        var_dump($clean);

?>
```

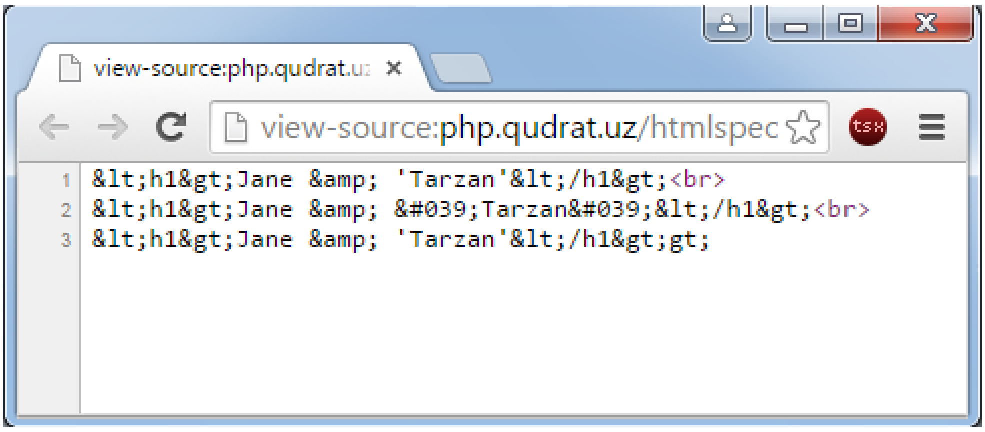
Natijasi:

string(39) " Bu yerda bir nechta so' z bor:) ...
string(15) "Satrga namuna "
string(11) 'Hello world"
string(34) 'Bu yerda bir nechta so' z bor:) ..."
string(30) 'Bu yerda bir nechta so'z bor:)"
string(5) "o wor"
string(13) «Batrga namuna"



htmlspecialchars funksiyasidan foydalanish

```
<?php
        $str = "<h1>Jane & 'Tarzan'</h1>";
        // faqat ikkitalik qo'shtirnoqm o'girish
        echo htmlspecialchars($str, ENT_COMPAT);
        echo "<br>\n";
        // har ikkala qo'shtirnoqm ham o'girish
        echo htmlspecialchars($str, ENT_QUOTES);
        echo "<br>\n";
        // har ikkalasini ham o'zgarishsiz qoldirish
        echo htmlspecialchars($str, ENT_NOQUOTES);

?>
```



4-Laboratoriya ishi

Mavzu: Tanlash operatori

NAZARIY BO'LIM

Ishning maqsadi: PHP da boshqarishni uzatish operatorlarida ishlashni o'rganish va qo'llash

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – PHP da switch, case, break, default funksiyalarni bilish;

Boshqarishni uzatish operatorlaridan yana biri tanlash operatoridir. Tanlash operatori asosan bir nechta qiymatdan, o'zgaruvchiga mos qiymatni tanlashda va qiymatlarga mos ravishda boshqarishni uzatishda ishlatiladi.

Tanlash operatorining umumiy KO'RINISHI:

switch {<o'zganivchi>} {

case <o'zgarmas ifoda l> : <operator 1>; **break; case <o'zgarmas** ifoda2> **: <operator 2**>; **break;**

case <o'zgarmas ifodaN> : <operator N>; break; [default : operator H + 1]; }

Tanlash operatorida boshqarilish o'zgaruvchiga mos ravishda qiymatlarga uzatiladi va mos operator ishga tushadi. **default** operatori birorta ham qiymat o'zgaruvchiga to'gri kelmasa ishlatiladi. **default** operatorini ishlatmasdan tashlab ketish ham mumkin.

Eslatma:

Dasturlashga doir kitoblarni o'qiganingizda, biror operatorning umumiy ko'rinishining to'rtburchak qavs [ ] belgisi oralig'ida yozilgan qismini ishlatmasdan tashlab ketish mumkin. Operatorning bu qismidan foydalanish ixtiyoriy bo'ladi.

Songa mos hafta kunni chiqarish

```
<?php
    $k = 5;
    switch ($k) {
        case 1:
            echo "Dushanba";
            break;
        case 2:
            echo "Seshanba";
            break;
        case 3:
            echo "Chorshanba";
            break;
        case 4:
            echo "Payshanba";
            break;
        case 5:
            echo "Juma";
```

```
                break;
            case 6:
                echo "Shanba";
                break;
            case 7:
                echo "Yakshanba";
                break;

            default:
                echo "Bunday hafta kum yo`q";
                break;
        }
?>
```

Tanlash operatorida bir nechta qiymatga bir xil operator ishlatishi quyidagicha bo'ladi.

```
<?php
        $k = 'MAY';
        $k = strtolower($k); // kichik harflarga o'girish

        switch ($k) {
            case 'yanvar':
            case 'fevral':
            case 'dekabr':
                echo "Qish oyi";
                break;

            case 'mart':
            case 'aprel':
            case 'may':
                echo "Bahor oyi";
                break;

            case 'iyun':
            case 'iyul':
            case 'avgust':
                echo "Yoz oyi";
                break;

            case 'sentyabr':
            case 'oktyabr':
            case 'noyabr':
                echo "Kuz oyi";
                break;        }
?>
```

5-Laboratoriya ishi

Mavzu: for sikl operatori

NAZARIY BO'LIM

<u>Ishning maqsadi</u>: PHP da sikl operatorlarida ishlashni o'rganish va qo'llash
<u>Ishning natijasida talaba quyidagilarga ega bo'lishi kerak</u>:
Bilishi kerak – PHP da for sikl funksiyalarni bilish;

Bir xil hisoblash jarayonlarini bir necha bor takrorlanish **Sikl** deyiladi. PHP da sikl operatorining bir necha xil turi mayjud.

- for sikl operatori
- do ... while sikl operatori
- while sikl operatori
- fareach sikl operatori

Yechilayotgan masalaga qarab, programmist o'zi uchun qulay bo'lgan sikl operatoridan foydalanishi mumkin

for takrorlanish operatorining sintaksisi quyidagicha:

for (<±fodal>; <±foda2>; <±foda3>} <operator Yoki blok>;

Bu operator amai qillshm <lfoda1> bajarishdan boshlaydl. Keyln takrorlash qadamlari boshlanadi. Har bir qadamda <lfoda2> bajariladi, agar natija 0 dan farqli Yoki rost (true) bo'lsa, sikl tanasi -<operator Yoki blok> bajariladi va oxirida <ifoda3> bajariladi, aks holda boshqaruv takrorlash operatoridan keyingi operatorga o'tiladi. Sikl tanasi - <operator Yoki blok> slfatlda bitta operator, shujumladan bo'sh operator, Yoki operatorlar bloki kellshi mumkin.

Sikl takrorlanishi davomida bajarilishi lozim bo'lgan operatorlar majmuasi sikl tanasi deyiladi. Sikl tanasi sifatida bir Yoki bir nechta operatordan foydalanish mumkin. Agar sikl tanasida bir nechta operatordan foydalanmoqchi bo'lsak bu operatorlarni blok {} orasiga olishimiz kerak

for sikl operatori mumumiy ko'rinishda quyidagicha ham ifodalash mumkin:

for (sikl_o'zgaruvchisining_boshlang'ich_qiymati; takrorlanish_sharti; sikl_o'zgaruvchisini_oshirish) sikl_tanasi; sikl_o'zgaruvchisini_oshirish o'rinda kamaytirish ham bo'lishi mumkin.

```
<?php
	for ($i=1; $i <= 10; $i++) {
		echo $i."-satr<br>";
	}
 ?>
```

Natijasi:

1-satr
2-satr
3-satr
4-satr
5-satr
6-satr
7-satr
8-satr
9-satr
10-satr

break funksiyasi

break - funksiyasini har qanday sikl operatoriga qo'llash mumkin. Bu funksiya sikl tugatilishim ta'minlaydi. Ya’ni boshqarilishni sikl operatoridan keyingi operatorga uzatadi.

```
<?php
    for ($i=1; $i <= 10; $i++) {
        if ($i == 5) {
            break;      // siklm tugatish
        }
        echo $i."-satr<br>";
    }
?>
```

Natijasi:

1-satr
2-satr
3-satr
4-satr

continue funksiyasi

continue - funksiyasini har qanday sikl operatoriga qo'llash mumkin. Bu funksiya sikl parametrining keyingi qiymatni qabul qilishini taminlaydi. Boshqacha so'z bilan aytganda sikl tanasi tugatiladi. Bunda slklning o'zi tugatilmaydi.

```
<?php
    for ($i=1; $i <= 10; $i++) {
        if ($i == 5) {
            continue;   // keyingi qadamga o'tish 5 chiqarilmaydi
        }
        echo $i."-satr<br>";
    }
```

Natijasi:

1-satr
2-satr
3-satr
4-satr
6-satr
7-satr
8-satr
9-satr
10-satr

Mustaqil ishlash uchun masalalar:

1.N natural soni berilgan. Shu son bo'luvchllarim chiqaruvchi programma tuzilsin. Masalan: 30ning bo'luvchllari: 1, 2,3,5,6, 10,15, 30

2.N natural soni berilgan. Shu son bo'luvchllari yig'indislm chiqaruvchi programma tuzilsin. PJasalan: N=15;javob: 24 (chunkl 1+3+5+15)

3.N natural soni berilgan. Shu son mukammal Yoki mukammal emasligini anlqlovchi programma tuzilsin. O'zidan boshqa bo'luvchilari yig'indisi o'ziga teng bo'lgan son mukammal son deyiladi. Masalan:6 = 1+2+3 mukammal son.

4.N natural soni berilgan. Shu songacha bo'lgan mukammal sonlarni chiqaruvchi programma tuzilsin.

5.N natural soni berilgan. N gacha bo'lgan 3 ga bo'llnadigan lekin 5 ga bo'linmaydigan sonlarni chiqaruvchi programma tuzilsin.

6.N natural soni berilgan. Shu son tub Yoki tub emasligini aniqlovchi programma tuzilsin. Faqat 1 ga va o'ziga bo'linadigan son tub son deyiladi.

7.N natural soni berilgan. Shu songacha bo'lgan tub sonlarni chiqaruvchi programma tuzilsin.

8.N natural soni berilgan. N gacha bo'lgan do'st sonlarni chiqaruvchi programma tuzilsin. Agar birinchi son bo'luvchilari ylg'indisi ikkinchi songa, ikkinchi son bo'luvchilari yig'indisi birinchi songa teng bo'lsa, bu sonlar do'st sonlar deyiladi.

6-Laboratoriya ishi

Mavzu: do-while, while sikl operatorlari

NAZARIY BO'LIM

Ishning maqsadi: PHP da sikl operatorlarida ishlashni o'rganish va qo'llash
Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:
Bilishi kerak – PHP da do-while, while sikl funksiyalarni bilish;

Ushbu mavzuda do-while va while sikl operatorlari bi Ian tanishamiz.
do - while opera tori
do {
operator; } while (shart);
Bu yerda dova while xizmatchi so'zlar.
(shart) sikl tanasi bajargandan so'ng, sikldan chiqish uchun tekshiriladigan shart. (mantiqiy ifoda).

do - while operatorning ishlash tartibi

do xizmatchi so'zidan keyingi operatorlar bajariladi. keyin while xizmatchi so'zidan keyingi shart tekshiriladi. Agar shart rost (true) natija bersa do xizmatchi so'zidan keyingi operatorlar qayta bajanladi. Shart qayta tekshiriladi. bu jarayon shart yolg'on (false) natija berguncha takrorlanadi. Qachon while xizmatchi so'zidan keyingi shart yolg'on (false) qiymatga ega bolsa, boshqarilish do - while operatoridan keyingi operatorga uzatiladi.

do - while sikl operatorida sikllanib qolish

DIQQAT: *do - while sikl operatoridan, qachon while xizmatchi so'zidan keyingi (shart) false (yolg'on) qiymat qabul qilsa chiqiladi. Ya'ni boshqarilish do - while operatoridan keyingi operatorga uzatiladi. Agar (shart) false qiymat qabul qilmasa. do - while sikl operatoridan chiqib ketilmaydi va bu jarayon sikllanib qolish deyiladi.*

```
<?php
	do{
		$k++;
		echo $k."-satr<br>";
	} while ($k < 10)

?>
```

Natijasi:

1-satr
2-satr
3-satr
4-satr
6-satr

7-satr
8-satr
9-satr
10-satr

do - **while** operatorida siklning tanasi kamida bir marta takrorlanadi. Shu bir marta hisoblash ham yechilayotgan masalani mohiyatini buzib yuborishi mumkin. Bunday hollarda **while** sikl operatoridan foydalangan maqsadga muvofiq.

while operatori

while sikl operatorning ishlash tartibi

Agar (shart) rost (**true**) qiymatga ega bo'lsa, **sikl_tanasi** bajariladi. Qachonki shart yolg'on **(false)** qiymatga teng bo'lsa sikl tugatiladi.

Agar (shart)true qiymatga ega bo'lmasa sikl tanasi biror marta ham bajarilmaydi. **while sikl operatoridan qanday chiqiladi?**

while sikl operatoridan qachon (**shart**) false (yolg'on) qiymat qabul qilsa chiqiladi. Ya’ni boshqarilish **while** operatoridan keyingi operatorga uzatiladi. Agar (**shart**) false qiymat qabul qilmasa. while sikl operatoridan chiqib ketilmaydi va bu jarayon sikllanib qolish deyiladi.

```
<?php
    while ($k < 10) {
        $k++;
        echo $k."-satr<br>";
    }

?>
```

Natijasi:

1-satr
2-satr
3-satr
4-satr
6-satr
7-satr
8-satr
9-satr
10-satr

while sikl operatori ma'lumotlar bazasi bilan ishlaganda ko'proq ishlatiladi.

Mustaqil yechish uchun masalalar:

1. A va B natural sonlari berilgan. Evklid algoritmi bo'yicha EKUB(a,b)ni aniqlovchi dastur tuzilsin. EKUB - Eng Katta Umumiy Bo'luvchisi. Agar a >= b bolsa, a ni b ga bo'lgandagi qoldiq. q=0 bo'lsa. EKUB(a,b)=b. Agar q nolga teng bo'lmasa ushbu tengiik o'rinli bo'ladi. EKUB(a,b)=EKUB(b,q). Masalan: EKUB(15,6)=EKUB(6,3)=EKUB(3,0)=3
2. N soninini soniga bo'lgandagi qoldiqm va butun qismini bo'lish amallarini (/, %) ishlatmasdan topuvchi dastur tuzilsin.
3. N natural soni berilgan. Shu son raqamlari yig'indisini chiqaruvchi dastur tuzilsin.

7-Laboratoriya ishi

Mavzu: Funksiyalar bilan ishlash

NAZARIY BO'LIM

Ishning maqsadi: PHP da qism dasturlar bilan ishlashni o'rganish va qo'llash

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – PHP da funksiyalarni e'lon qilish va ixtiyoriy joyda chaqirib ishlatishni bilish;

Dasturlash mobaynida bir xil ifodalarni hisoblash jarayonlarini qayta - qayta hisoblashga to'g'ri keladi.

Dasturlash tillarida, kompyuter xotirasini va dasturchining vaqtini tejash maqsadida. bunday takrorlanuvchi jarayonlarni dasturda ajratib yozib, unga asosiy daturdan, boshqa funksiyalardan murojaat qilish imkomyatlari keltirilgan.

Dasturning istalgan qismidan murojaat qilib, bir necha bor ishlatish mumkin bo'lgan operatorlar guruhiga **funksiya** deyiladi.

PHP da har xil vazifalarni bajaruvchi 1000 ga yaqin funksiyalar bor. Bulardan tashqari har bir dasturchi o'zining funksiyasini yozishi va istalganicha foydalanishi mumkin. Bunday funksiyalarni foydalanuvchi funksiyalari deyiladi. Ya'ni php foydalanuvchisi funksiyasi.

Parametrsiz funksiyani e'lon qilish quyidagicha bo'ladi.

```
function funksiya_nomi{ ) {
// bajarilishi lozim bo'lgan kodlar;
}
```

Funksiya bilan ishlashda quyidagilarga e'tibor bering.

Funksiya nomi lotin harflaridan Yoki ost chiziqdan (_) boshlanishi mumkin. Lekin sondan boshlanichi mumkin emas.

Funksiyaga u bajaradigan ma'nosini bildiruvchi nom bering. Funksiya nomi yozilishida katta kichik harflar ahamiyatga ega emas.

Funksiya nomi yozishida katta kichik harflar ahamiyatga ega emas

```
<?php
    function salom(){
        echo 'Assalomu alaykum bo`ljak dasturchi!<br>';
    }

    // Funksiya nomi yozilishida katta kichik harflar axamiyatga ega emas
    salom();
    SaLoM();
    SALOM();

?>
```

Natijasi:

Assalomu alaykum bo'lajak dasturchi
Assalomu alaykum bo'lajak dasturchi
Assalomu alaykum bo'lajak dasturchi

Parametrsiz funksiyalar asosiy programmadan (chaqiruvchi funksiyadan) hech qanday o'zgaruvchi qabul qilib olmaydi. Shuning uchun parametrsiz funksiya deyiladi.

Agar funksiyaga biror qiymatni yuborish lozim bolsa, parametrli funksiyadan foydalaniladi. Parametrli funksiyaga qiymat sifatida istalgancha o'zgaruvchi yuborish mumkin. Bunda o'zgaruvchllar vergul bilan ajratiladi.

Parametrli funksiyani e'lon qilish quyidagicha bo'ladi.

```
function funksiyaNomi{$qiymat1, $qiymat2,$qiymatN …)
 {  // bajarilishi lozim bo'lgan kodlar; }
```

```
<?php
        function salom($ism){
                echo "Assalomu alaykum $ism!<br>";
        }

        salom('Azamat');
        salom('Qudrat');
        salom('Ne`mat');

?>
```

Natijasi:

Assalomu alaykum Azamat!
Assalomu alaykum Qudrat!
Assalomu alaykum Ne'mat!

Funksiya parametrlariga boshlang'ich (default) qiymat berish mumkin. Agar parametrli funksiyaga murojaat qilishda qiyrnat parametri tushirib qoldirilsa. boshlang'ich (default) qiymat ishlatiladi.

Parametrli funksiyaga boshlang'ich qiymat berish

```
<?php
        function salom($ism = 'bo`lajak dasturchi'){
                echo "Assalomu alaykum $ism!<br>";
        }
        salom('Ismoil');
        salom('Qudrat');
        salom(); // bu xolda default qiymat ishlatiladi

?>
```

Natijasi:

Assalomu alaykum Isroil!
Assalomu alaykum Qudrat!
Assalomu alaykum bo'lajak dasturchi!

Funksiya biror qiymat qaytarilishi lozim bo'lsa, return xizmatchi so'zidan foydalaniladi.

```
<?php
    function ekub($x, $y){
       while ($x != $y) {
          if ($x > $y) {
               $x -= $y;
          } else {
               $y -= $x;
          }
       }
       return $x;
    }
    echo ekub(24, 18) . "<br>";
    echo ekub(10, 35) . "<br>";
?>
```

Global o'zgaruvchilar

Ham asosiy programmada ham funksiyada ishlatish mumkin bo'lgan o'zgaruvchilar global o'zgaruvchilar deyiladi. Global o'zgaruvchilar asosiy programmada e'lon qilinishi kerak.

Lokal o'zgaruvchilar

Faqat funksiyada ishlatish mumkin bo'lgan o'zgaruvchilarga lokal o'zgaruvchilar deyiladi. Ular funksiyada e'lon qilinadi. Funksiyada yana bir nechta ichki funksiyalardan foydalanish mumkin.

Blok ichida e'lon qilingan o'zgaruvchilar. Shu blok uchun lokal o'zgaruvchilar hisoblanadi. Bu o'zgaruvchilardan faqat blok ichida foydalanish mumkin.

1. Global va lokal o'zgaruvchilar

```
<?php
    $x = 5; // global
    function myTest(){

          $y = 10; // funksiya uchun lokal

          echo "<p> funksiya ichidagi o'zgaruvchilar</p>";

          echo "x=$x<br>";
```

```
        echo "y=$y<br>";
        echo "Global x=".$GLOBALS['x'].'<br>';

    } // funksiya tugadi

    myTest();

    echo "<p> funksiyadan tashqaridagi o'zgaruvchilar</p>";

    echo "x=$x<br>";
    echo "y=$y<br>";

 ?>
```

Natijasi:

Funksiya ichidagi o'zgaruvchilar qiymati:
x=
y=10
Funksiyadantashqaridagi o'zgaruvchilar qiymati:
x = 5
y =

2. Global va lokal o'zgaruvchilar

```
<?php
    function namuna(){
        // $k lokal o'zgaruvchi
        $k = 'Bu o`zgaruvchi faqat funksiyda ishlaydi<br>';

        echo $k;
    }

    namuna();

    echo $k.'-bu yerda hech narsa chiqmaydi. <br>
        Chunki, bu $k global o`zgaruvchi va unga qiymat berilmagan!';

?>
```

Natijasi:

Lokal o'zgaruvchi
Global o'zgaruvchi
Global o'zgaruvchi

Global o'zgaruvchilarga global kalit so'zi orqali ham murojaat qilish mumkin.

Funksiya da global o'zgaruvchilarni e'lon qilish

```php
<?php
	$x = 5; $y = 10;
	function sum() {
	  // $x va $y global o'zgaruvchi ekanligim e'lon qilish
	  global $x, $y;
	  $y += $x;
	}
	sum();
	echo $y; // 15 chiqadi
?>
```

Static o'zgaruvchilar

Funksiyaga murojaat qilinganida lokal o'zgaruvchilar har doim boshlang'ich qiymatdan boshlanadi. Static o'zgaruvchilarda esa, funksiyaga oldingi murojaatdagi qiymati saqlab qoladi. Masalan : biror funksiyaga necha marta murojaat qilinganini aniqlashiniiz mumkin.

Static o'zgaruvchilarga misol

```php
<?php
	function test(){
		// static o'zgaruvchini e'lon qilish
		static $soni = 0;
		$soni++;

		echo "<p>Funksiyaning chaqirilishlar soni = $soni</p>";
	}
	test();
	test();
	test();
?>
```

Natijasi:

Funksiyaning chaqirilishlar soni = 1 Funksiyaning chaqirilishlar soni = 2 Funksiyaning chaqirilishlar soni = 3

Na'munaviy masala: №1

```php
<?php
	$s = 'Global o`zagruvchi<br>';

	function namuna(){
		$s = 'Lokal o`zgaruvchi<br>';
		echo $s;

		// funksiyada global o'zgaruvchiga murojaat qilish
		echo $GLOBALS['s'];
```

```
        }
        // funksiyani chaqirish
        namuna();
        echo $s; // global o’zgaruvchini chiqarish

?>
```

Na’munaviy masala: №2

```
<?php

        function test(){

                static $s = 0;
                $k = 0;

                $s++;
                $k++;

                echo "Static o`zgaruvchi qiymati=$s<br>";
                echo "Oddiy o`zgaruvchi qiymati=$k<br>";
                echo "<br>";
        }

        test();
        test();
        test();

?>
```

Na’munaviy masala: №3

```
<?php
        $x = 5; // global o'zgaruvchi

        function myTest() {
          $y = 10; // local o'zgaruvchi
          echo "<p>Funksiya ichidagi o`zgaruvchilar qiymati:</p>";
          echo "x = $x";
          echo "<br>";
          echo "y = $y";
        }

        myTest();
        echo "<p>Funksiyadan tashqaridagi o`zgaruvchilar qiymati:</p>";
        echo "x = $x";
        echo "<br>";
        echo "y = $y";
?>
```

8-Laboratoriya ishi

Mavzu: PHP da massivlar bilan ishlash

NAZARIY BO'LIM

Ishning maqsadi: PHP da to'plam ma'lumotlar bilan ishlashni o'rganish va qo'llash

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – PHP da massivlarni e'lon qilish va turli xil amallar bajarishni bilish;

Massiv bu ko'p qiymatlarni bir o'zgaruvchida saqlashdir.

Massivga namuna

```
<?php
    $cars = array("Malibu", "Captiva", "Orlando", "Lacetti", "Cobalt",
"Nexia", "Damas", "Spark", "Matis");

    echo "GMUzbekistan avtomobillari:";
    echo "<br>". $cars[0]."<br>" . $cars[1] . "<br>" . $cars[2];
    echo "<br>". $cars[3]."<br>" . $cars[4] . "<br>" . $cars[5];
    echo "<br>". $cars[6]."<br>" . $cars[7] . "<br>" . $cars[8];

?>
```

Natijasi:

GMUzbekistan avtomobillari:
Malibu
Captiva
Orlando
Lacetti
Cobalt
Nexia
Damas
Spark
Matiz

M assiv mma?

Massiv bir vaqtning o'zida bir nechta qiymatni o'zida saqlovchi o'zgaruvchi.

Sizda ba'zi narsalarning ro'yxati bor. Masalan, mashinalar nomlarim oddly o'zgaruvchilarda quyidagicha yozishingiz mumkin.

$mashina1 = "Malibu";
$mashina2 = "Lacetti";
$mashina3 = "Cobalt";

Agar sizda 100 dan ortiq mashinaning ro'yxati bo'lsachi? Bu o'rlnda massivdan foydalanishingiz lozim bo'ladi.

Massivda bir nom bilan istalgancha qiymat saqlash mumkin bo'ladi. Massiv elementiga uning indeksi orqali murojaat qilinadi.

Massiv elementlari	$cars[0]	$cars[l]	$cars[2]	$cars[3]
qiymati	Malibu	Captiva	Orlando	Lacetti

cars massivining 0 - indeksdagi qiymati Malibu. Lacetti esa 3 -indeksda joylashgan.

PHPda 3 toifadagi massivlar bor.

1.**Indexed arrays.** Indekslangan massivlar. Bunda massiv indekslari sonlar bilan indekslanadi. Massiv elementiga shu Indekslar orqali murojaat qilinadi.

2.**Associative arrays**. Bog'langan massivlar. Boshqacha so'z bilan aytgandan massiv elementlari biror kalit so'zlarga bo'glab qo'yiladi. Massiv elementiga shu kalit so'zorqali rnurojaat qilinadi.

3.**Multidimensional arrays**. Ko'p o'lchamli massivlar. **Indexed arrays. Indekslangan massiv**

PHPda massiv hosil qilish uchun array() funksiyasidan foydalanlladi. Indekslangan massivlarni hosil qilishning 2 xil usuli bor.

Massivlarni quyidagicha hosil qilish mumkin. Bunda massiv elementlar avtomat 0 dan boshlab indekslanadi.

```
$cars = array("Malibu", "Captiva", "Orlando", "Lacetti", "Cobalt");
```

Yoki quyidagicha qiymat berish mumkin.

```
$cars[e]    =    "Halibu";
$cars[l]    =    "Captiva";
$cars[2]    =    "Orlando";
$cars[3]    =    "Lacetti";
$cars[4]    =    "Cobalt";
```

count funksiyasi orqali massiv elementlari sonini aniqlash mumkin.

```
<?php
    $cars = array("Malibu", "Captiva", "Orlando", "Lacetti", "Cobalt",
"Nexia", "Damas", "Spark", "Matis");

    echo '$cars massivi elementlari soni='.count($cars);

?>
```

Natijasi:

$cars massiv elementlar soni 9 ta

Indekslangan massivga siklni qo'llash

```
<?php
        $cars = array("Malibu", "Captiva", "Orlando", "Lacetti", "Cobalt",
        "Nexia", "Damas", "Spark", "Matiz");

        $soni = count($cars);
        for($i = 0; $i < $soni; $i++)
        {
                echo $i.") ".$cars[$i]."<br>";
        }

?>
```

Natijasi:

0) Malibu
1) Captiva
2) Orlando
3) Lacetti
4) Cobalt
5) Nexia
6) Damas
7) Spark
8) Matiz

Associative Arrays. Bog'langan massivlar

Bu turdagi massivlarda. massiv elementlari biror kalit so'zlarga bo'glab qo'yiladi. Massiv elementiga shu kalit so'z orqali rnurojaat qilinadi.

Buturdagi masslviami hosil qilishm ham 2xil usuli bor.

```
<?php
        $cars = array("Malibu" => "99 750 000", "Captiva" => "93 001
500", "Orlando" => "76 200 000");

        foreach ($cars as $kalit_soz => $qiymat) {
                echo $kalit_soz." narxi ".$qiymat." so`m<br>";
        }

?>
```

Natijasi:

Malibu narxi 99 750 000 so'm
Captiva narxi 93 001 500 so'm
Orlando narxi 76 200 000 so'm

9-Laboratoriya ishi

Mavzu: PHPda ko'p o'lchamli massivlar

NAZARIY BO'LIM

<u>Ishning maqsadi</u>: PHP da to'plam ma'lumotlar bilan ishlashni o'rganish va qo'llash

<u>Ishning natijasida talaba quyidagilarga ega bo'lishi kerak</u>:

Bilishi kerak – PHP da massivlarni e'lon qilish va ko'p o'lchamli massivlar ustida turli xil amallar bajarishni bilish;

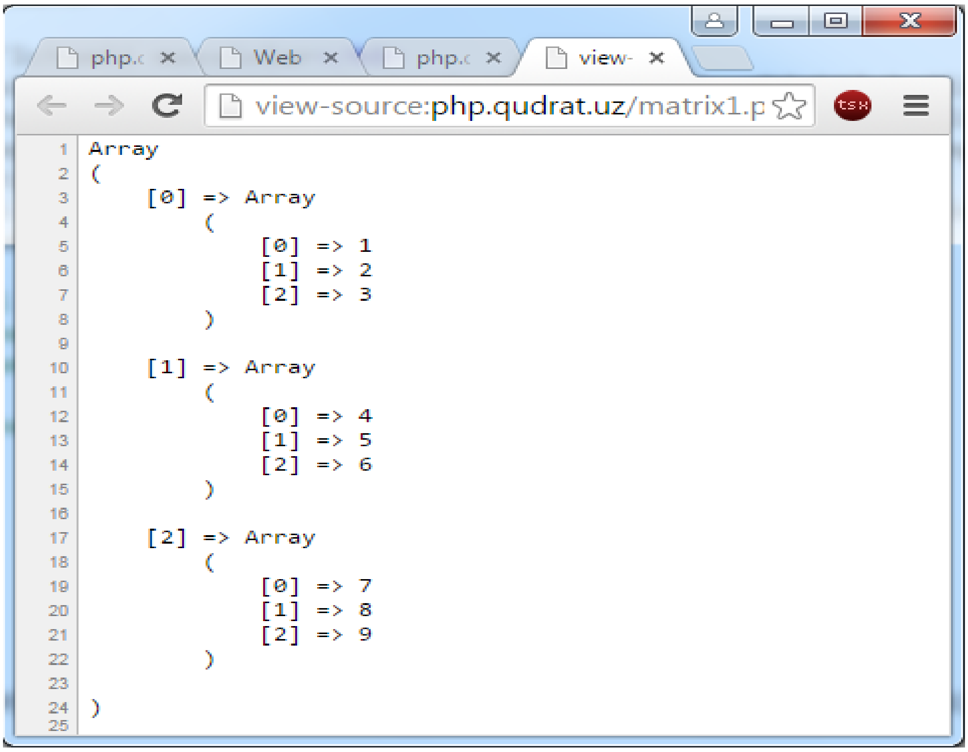
PHPda massiv elementlari ixtiyoriy tolfada (son, satr, ob'ekt) bo'lishi mumkin. Massiv element] sifatida yana massiv bo'llshi ham mumkin. Shu orqali ko'p o'lchamli massiv hosil qilinadi. Odatda 3 o'lchamlidan ko'p o'lchamli massiv ishlatilmaydl. Men bir martagim 3 o'lchamli masslvdan foydalanganlmm eslAynian. Asosan bir va ikki o'lchamli massivlar ishalatlladi.

Ko'p o'lchamli massivni array() konstruksiyasi yordamida hosil qilish mumkin. Bunda massivning har bir elementi yana massiv bo'ladi.

Ikki o'lchamli massivga misol

```
<?php
	$matrix = array(
		array(1, 2, 3),
		array(4, 5, 6),
		array(7, 8, 9)
		);
	print_r($matrix);

?>
```

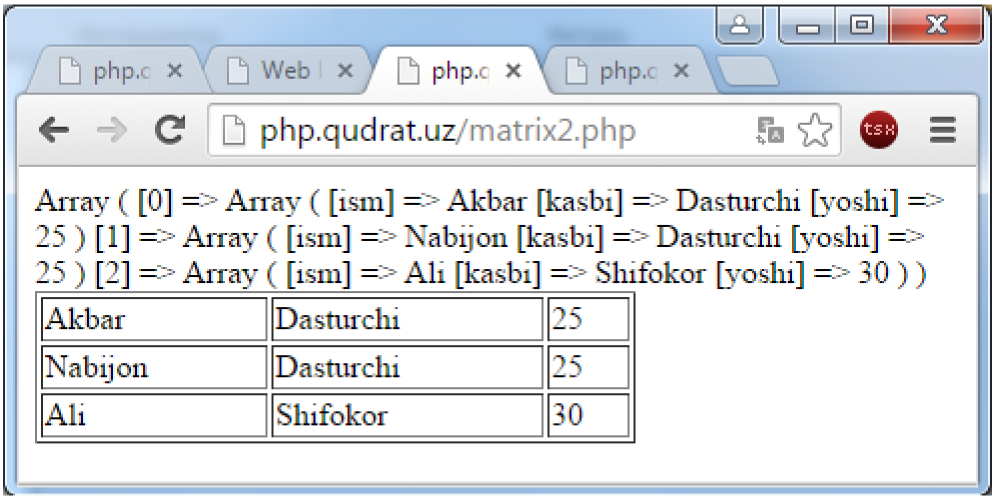
Natijasi:



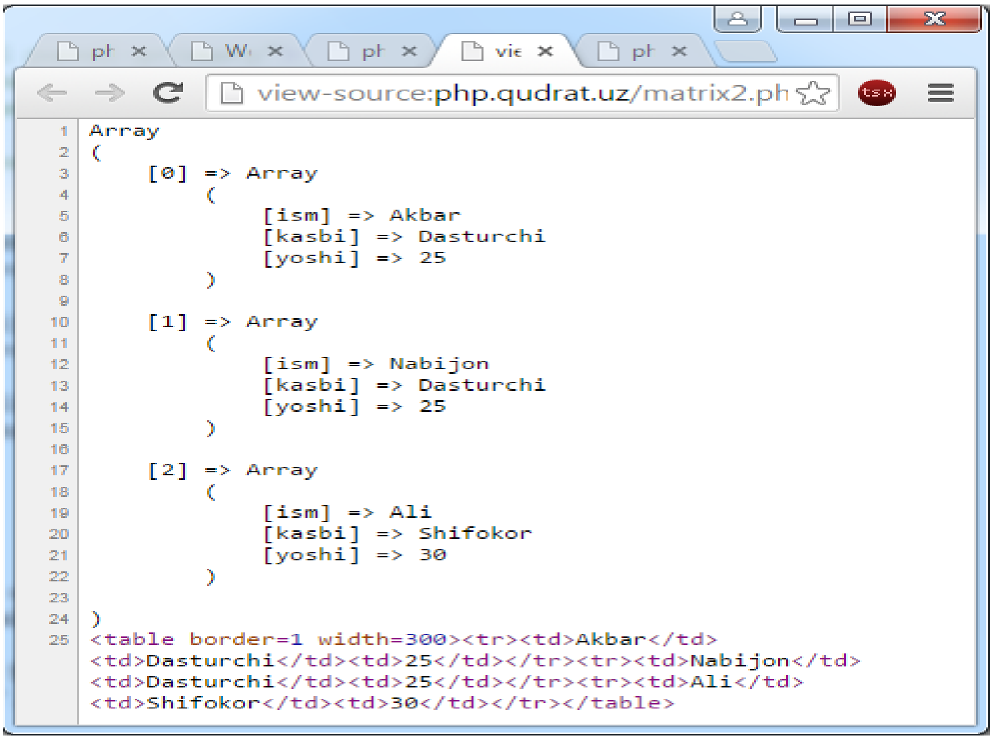
Bog'langan massiv

```
<?php
        $matrix = array(
        array('ism'   => 'Akbar',
                  'kasbi'       => 'Dasturchi',
                  'yoshi'       => 25),
        array('ism'   => 'Nabijon',
                  'kasbi'       => 'Dasturchi',
                  'yoshi'       => 25),
        array('ism'   => 'Ali',
                  'kasbi'       => 'Shifokor',
                  'yoshi'       => 30)
                );

        print_r($matrix);

        echo "<table border=1 width=300>";
        foreach ($matrix as $odamlar) {
                echo "<tr>";
                foreach ($odamlar as $odam => $qiymat) {
                        echo '<td>'.$qiymat.'</td>';
                }
                echo '</tr>';
        }
```

echo "</table>";
?>
Natijasi:



Script Natijasi:



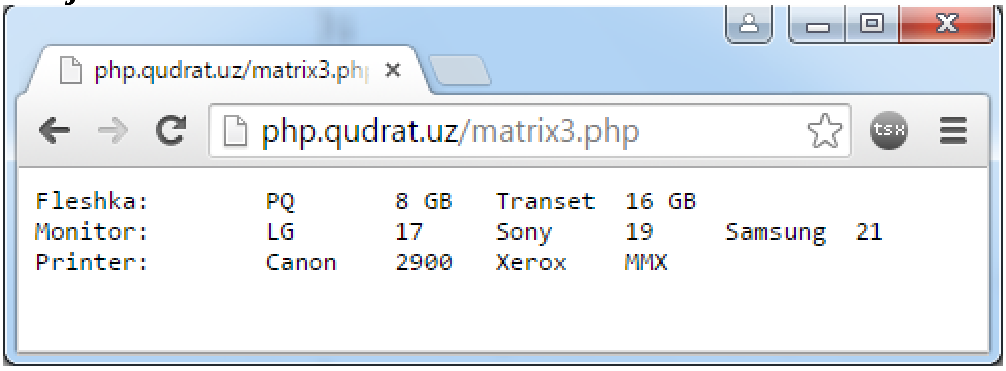
Sahifa kodi

Quyidagi misolda product nomli massiv hosil qilingan. product massivining 3 ta elementi bor Fleshka. Momtor, Printer. Ularning har biri esa kalit so'z (key) va qiymatga (value) ega bo'lgan yangi massiv. Eng qizig'i ichki massiv elementari soni bir xil bo 'lishi shartemas. Fleshka va Printer massivi 2 ta elementga ega. Momtor massivi esa 3 ta elementga ega.

echo da yangi satrga o'tish uchun \n va tabulyatsiya uchun \t ishlatilgan.

```
<?php
	$product = array(
	'Fleshka' => array(
			'PQ'		=> '8 GB',
			'Transet'	=> '16 GB'),
	'Momtor' => array(
			'LG'	=> 17,
			'Sony' => 19,
			'Samsung' => 21),
	'Printer' => array(
			'Canon' => '2900',
			'Xerox' => 'MMX')
		);

	echo '<pre>';
	foreach ($product as $session => $item) {
		echo "\n$session:\t";
		foreach ($item as $key => $value) {
		echo "$key\t $value \t";
		}
	}
	echo "</pre>";

?>
```

Natijasi:



11-Laboratoriya ishi

Mavzu: PHP da massivlar bilan ishlovchi funksiyalar

NAZARIY BO'LIM

Ishning maqsadi: PHP da to'plam ma'lumotlar bilan ishlovchi funksiyalarni o'rganish va qo'llash;

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – PHP da massivlar bilan ishlovchi funksiyalarni o'rganish va turli xil amallar bajarishni bilish;

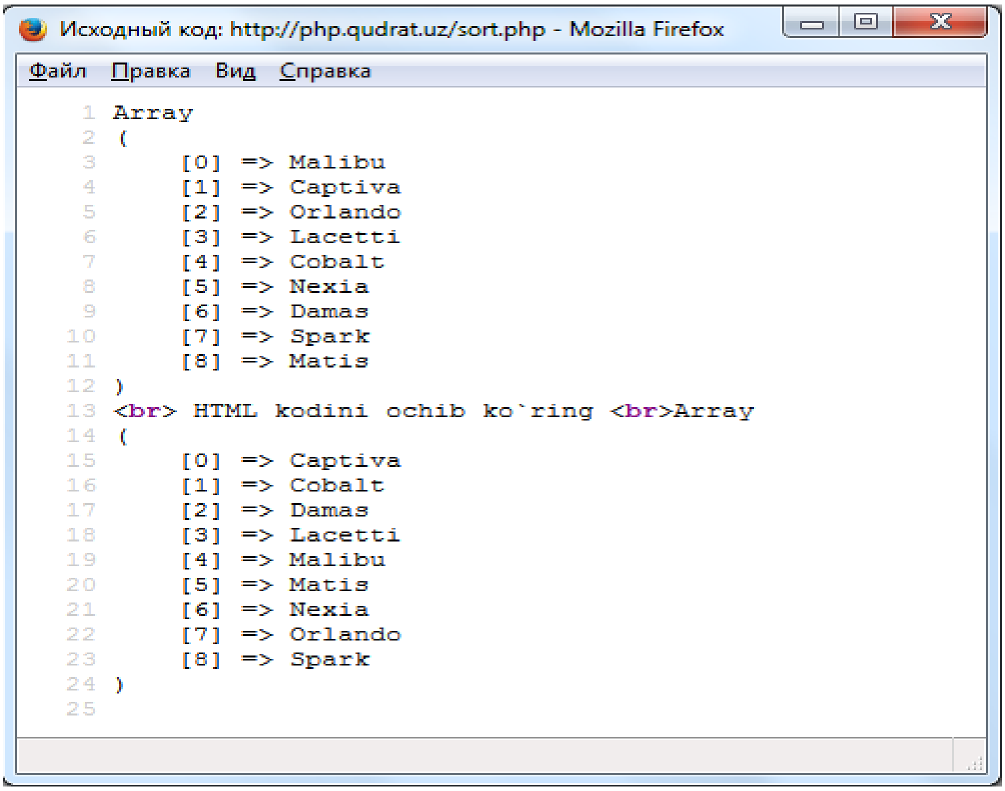
Bu darsda massivlar bilan ishlovchi ba'zi funksiyalarni o'rganib chiqamiz.

- **sort()**-funksiyasi massivni o'sish tartibida joylashtiradi
- rsort(}-funksiyasi massivni kamayish tartibida joylashtiradi
- **asort()** -funksiyasi bog'langan massivni qiymatlar bo'yicha o'sish tartibida joylashtiradi
- arsort()- funksiyasi bog'langan massivni qiymatiari bo'yicha kamayish tartibida joylashtiradi
- **ksort()** - funksiyasi bog'langan massivni kalit so'zlari bo'yicha o'sish tartibida joylashtiradi
- krsort()-funksiyasi bog'langan massivni kalitso'zlari bo'yicha kamayish tartibida joylashtiradi
- **shuffle**-funksiyasi massiv elementini aralashtirish uchun ishlatiladi
- **is_array** -funksiyasi o'zgaruvchini massiv Yoki massiv emasligini tekshirish uchun ishlatiladi
- **explode** - funksiyasi satrdagi so'zlarni ajratib massivga beradi.
- **extract**-funksiyasi massivdan qiymatlarnl o'zgaruvchilarga olish uchun ishlatiladi
- **compact**-funksiyasi o'zgaruvchilardan massiv hosil qilish uchun ishlatiladi
- **Massiv elementlarini o'qish uchun current, reset, end, next, prev funksiyalari ishlatiladi**

sort() - funksiyasi massivni o'sish tartibida joylashtiradi

```
<?php
        $cars = array("Malibu", "Captiva", "Orlando", "Lacetti", "Cobalt",
                "Nexia", "Damas", "Spark", "Matis");

        print_r($cars);

        echo "<br> HTML kodini ochib ko`ring <br>";

        sort($cars);
        print_r($cars);
?>
```

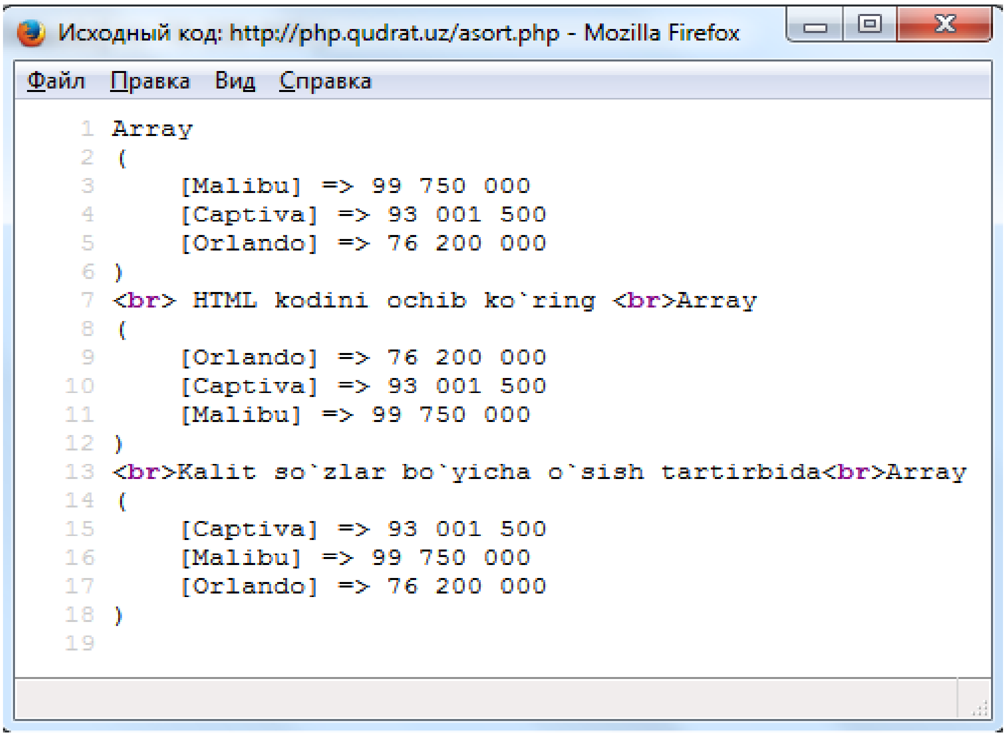
Natijasi:



asort() va ksort() funksiyasiga misol

```
<?php
    $cars = array("Malibu" => "99 750 000", "Captiva" => "93 001 500", "Orlando" => "76 200 000");

    print_r($cars);
    echo "<br> HTML kodini ochib ko`ring <br>";

    asort($cars);
    print_r($cars);

    echo "<br>Kalit so`zlar bo`yicha o`sish tartirbida<br>";
    ksort($cars);
    print_r($cars);

?>
```

Natijasi:




```
Array
(
    [Malibu] => 99 750 000
    [Captiva] => 93 001 500
    [Orlando] => 76 200 000
)
<br> HTML kodini ochib ko`ring <br>Array
(
    [Orlando] => 76 200 000
    [Captiva] => 93 001 500
    [Malibu] => 99 750 000
)
<br>Kalit so`zlar bo`yicha o`sish tartirbida<br>Array
(
    [Captiva] => 93 001 500
    [Malibu] => 99 750 000
    [Orlando] => 76 200 000
)
```

shuffle funksiyasi massivni aralashtirish uchun ishlatiladi

```
<?php
    $a = array(1, 2, 3, 4, 5, 6, 7, 8, 9);

    foreach ($a as $key => $value) {
        echo $value." ";
    }

    echo "<br>";

    shuffle($a); // massiv elementlarini aralashtirish
    foreach ($a as $key => $value) {
        echo $value." ";
    }
?>
```

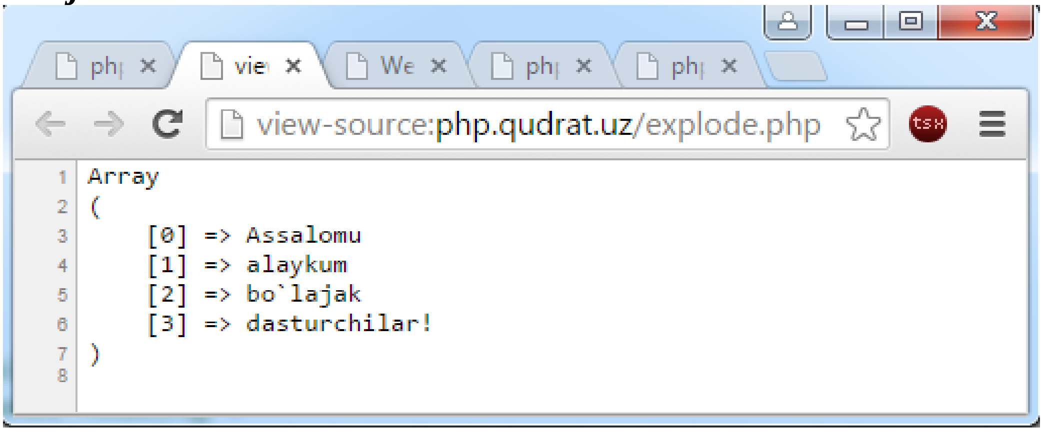
Natijasi:

1 2 3 4 5 6 7 8 9
5 7 8 9 6 4 2 3 1

is_array funksiyasi o’zgaruvchini massiv Yoki massiv emasligini tekshirish uchun ishlatiladi

```
<?php
	$a = array(1, 2, 3);
	$s = 'Assalomu alaykum';

	if (is_array($a)) {
		echo '$a massiv<br>';
	}
	else {
		echo '$a massiv EMAS <br>';
	}

	if (is_array($b)) {
		echo '$b massiv';
	}
	else {
		echo '$b massiv EMAS';
	}

 ?>
```

Natijasi:



extract - funksiyasi massivdan qiymatlarni o'zgaruvchilarga olish uchun ishlatiladi

```
<?php
	$a = "Asl o`zgaruvchi qiymati";
	$my_array = array("a" => "Cat",
	                  "b" => "Dog",
	                  "c" => "Horse");

	extract($my_array, EXTR_PREFIX_SAME, "oldqoshimcha");

	echo "\$a = $a; \$b = $b; \$c = $c;
```

```
\$oldqoshimcha_a = $oldqoshimcha_a<br>";

    echo "\$oldqoshimcha_b = $oldqoshimcha_b";
?>
```

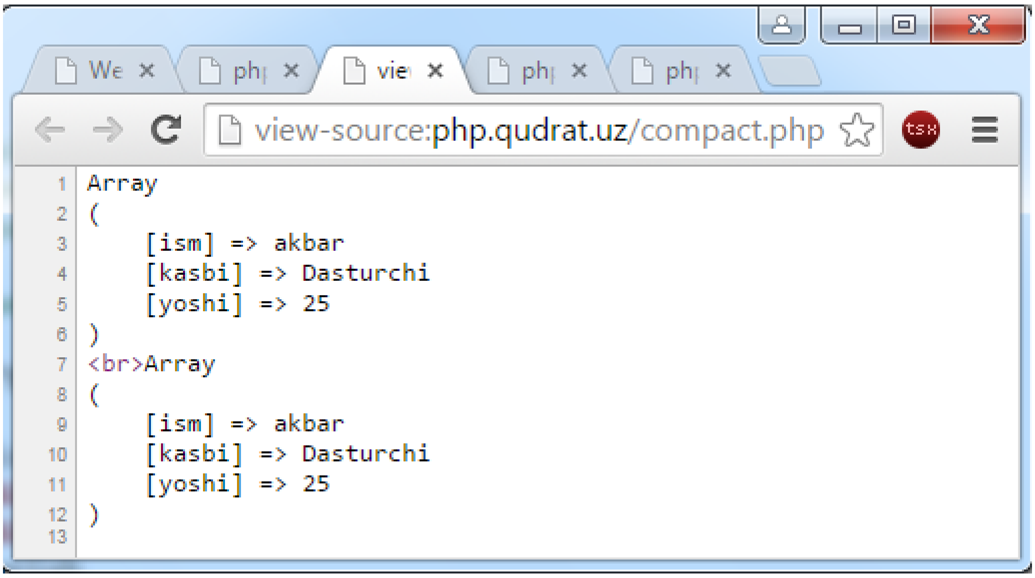
compact - funksiyasi o'zgaruvchilardan massiv hosi qilish uchun ishlatiladi

Bu funksiya extract funksiyasining teskarisini bajaradi. Ya’ni o'zgaruvchilardan massiv hosil qilish uchun ishlatiladi. E'tibor bering compact funksiyasida qo'shtirnoq ichidagi o'zgaruvchi oldidan S bo'lmaydi.

Quyidagi misolda $a va $b o'zgaruvchilariga bir xil massiv o'zlashtiriladi. $b massivni hosil qilishda oldin $ism kasbi yoshi satri so'zlarga ajratiladi va ular asosida massiv hosil qilinadi.

```
<?php
    $ism   = "akbar";
    $kasbi       = "Dasturchi";
    $yoshi       = "25";

    $a = compact('ism', 'kasbi', 'yoshi');
    print_r($a);
    echo "<br>";

    $b = compact(explode(' ', 'ism kasbi yoshi'));
    print_r($b);
?>
```

Natijasi:



Massiv elementlarini o'qish

current-joriy qiymatni o'qish uchun ishlatiladi. Bunda ko'rsatkich keyingi elementga otmaydi. **next** - ko'rsatkich joriy elementdan keyingisiga o'rnatiaidi va qiymat o'qiladi.

prev- ko'rsatkich joriy elementdan oldingisiga o'rnatialdi va qiymat o'qiladi.

reset - ko'rsatkich binnchi elementga o'rnatiladi va o'qiladi.

end - ko'rsatkich oxirgi elementga o'rnatiladi va o'qiladi.

```
<?php
	$people = array("Akbar", "Baxrom", "Domyor", "Qudrat");

	// joriy elementm chiqarish
	echo current($people) . '<br>';
	echo current($people) . '<br>';
	// keyingi elementm chiqarish
	echo next($people) . '<br>';

	// birinchi elementm chiqarish
	echo reset($people) . '<br>';
	// oxirgi elementm chiqarish
	echo end($people);
?>
```

Natijasi:

Akbar
Akbar
Baxrom
Akbar
Qudrat

11-Laboratoriya ishi

Mavzu: PHP da superglobal o'zgaruvchilar

NAZARIY BO’LIM

Ishning maqsadi: PHP da superglobal o'zgaruvchilar funksiyalarni o’rganish va qo’llash;

Ishning natijasida talaba quyidagilarga ega bo’lishi kerak:

Bilishi kerak – PHP da S_SERVER superglobal o'zgaruvchisi o’rganish va turli xil amallar bajarishni bilish;

PHP 4.1.0 versiyasidan boshlab ba'zi o'zgaruvchilar paydo bo'ldi. Ularni "**SuperGlobal o'zgaruvchilar**" deb nomlandi. Buning sababi bu o'zgaruvchilarga istalgan joydan murojaat qilish mumkin. Istalgan funksiya, sinf va fayldan murojaat qilish mumkin.

PHP superglobal o'zgaruvchilari:

1. **SGLOBALS**
2. **$_SERVER**
3. **$_REQUEST**
4. **$_POST**
5. **$_GET**
6. **$_FILES**
7. **$_ENV**
8. **$_COOKIE**
9. **$_SESSION**

Funksiyalar mavzusida SGLOBALS o'zgaruvchisi bilan tanishgan edik. Bu mavzuda bir nechta superglobal o'zgaruvchilar bilan tanishamiz. Qolganlari bilan keyingi mavzularda tanishib chiqamiz.

S_SERVER superglobal o'zgaruvchisi

SSERV E R server haqidagi ma’lumotlarni o'zida saqlaydi.

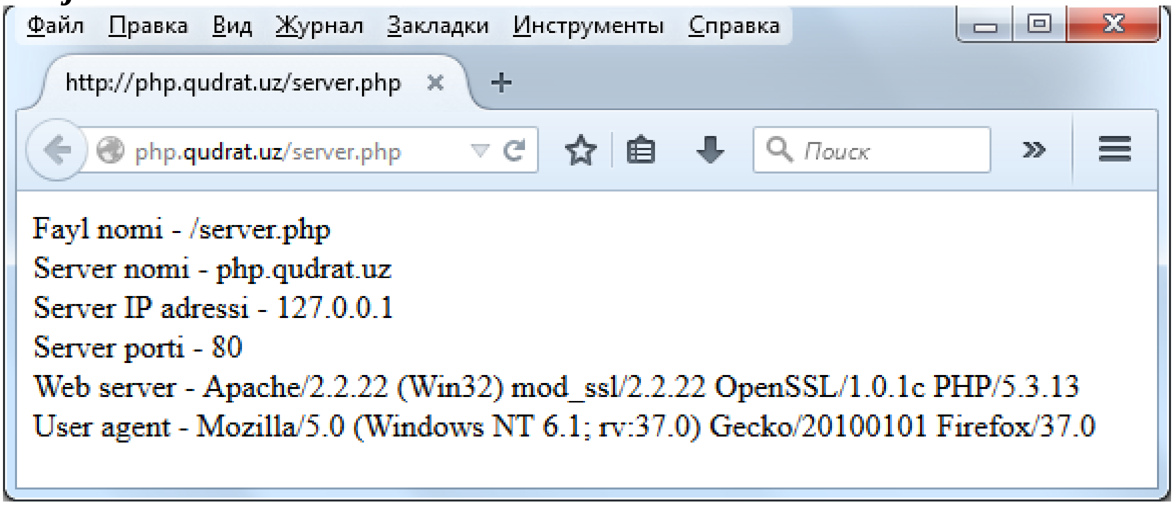
O'zgaruvchi	Izoh
$_SERVER['PHP_SELF']	Joriy script fayl nomini qaytaradi.
$_SERVER ['GATEWAY_INTERFACE']	Common Gateway Interface (CGI) versiyasini qaytaradi
$_SERVER['SERVER_ADDR']	Serverning IP adresini qaytaradi
$_SERVER['SERVER_NAME']	Server nomini qaytaradi masalan http://qudrat.uz
$_SERVER['SCRIPT_FILENAME']	Joriy scriptga to'liq yo'lni qaytaradi
$_SERVER['QUERY_STRING']	So'rov satrini qaytaradi
$_SERVER['HTTP_REFERER']	Joriy sahifaga qayerdan o'tilganini ko'rsatadi. Hamma holatlar uchun to'g'ri ishlamaydi.
$_SERVER['REQUEST_METHOD']	Murojaat qilish metodini qaytaradi
$_SERVER['REMOTE_ADDR']	Foydalanuvchining IP adresini qaytaradi

$_SERVERning barcha qiymatlarni ko'rlsh uchun *print_r* funksiya si orqali chop qilish lozim.

$_SERVER superglobal o'zgaruvchisiga namuna

```
<?php
	echo "Fayl nomi - ".$_SERVER['SCRIPT_NAME']."<br>";
	echo "Server nomi - ".$_SERVER['SERVER_NAME']."<br>";
	echo "Server IP adressi - ".$_SERVER['SERVER_ADDR']."<br>";
	echo "Server porti - ".$_SERVER['SERVER_PORT']."<br>";
	echo "Web server - ".$_SERVER['SERVER_SOFTWARE']."<br>";
	echo "User agent - ".$_SERVER['HTTP_USER_AGENT'];
?>
```

Natijasi:



Superglobal o'zgaruvchilar va xavfsizlik

Superglobal o'zgaruvchilardan foydalanganda juda ehtiyotkor bo'lish lozim. Chunki, xakerlar aynan $_POST, $_GETYoki boshqa superglobal o'zgaruvchilari orqali saytga xujum qilishlari mumkin. Ya’ni superglobal o'zgaruvchilarga zarar keltiruvchi uchun Yoki MySQL buyruqlarini yozishlari mumkin. Natijada hammaga oshkor qilish mumkin bo'lmagan ma'lumotlarni olishlari Yoki zarurlarni o'chirib tashlashlari mumkin bo'ladi.

Shuning uchun. superglobal o'zgaruvchilardan foydalanishda ularni boshlang'ich qayta ishlash lozim bo'ladi. Buning uchun PHP da bir qator funksiyalar mavjud. Masalan, **htmlspecialchars** funksiyasi barcha belgilarni HTML kodiga o'giradi. Ya’ni, "kichik" va "katta" (< va >) belgilarini < va > ga o'zgartiradi.

Superglobal o'zgaruvchilardan foydalanishdan oldin satrlar bilan ishlovchi funksiyalar va forma elementlarni tekshirish kabi bir qator mavzular bilan tanishib chiqing.

12-Laboratoriya ishi

Mavzu: GET va POST metodlari

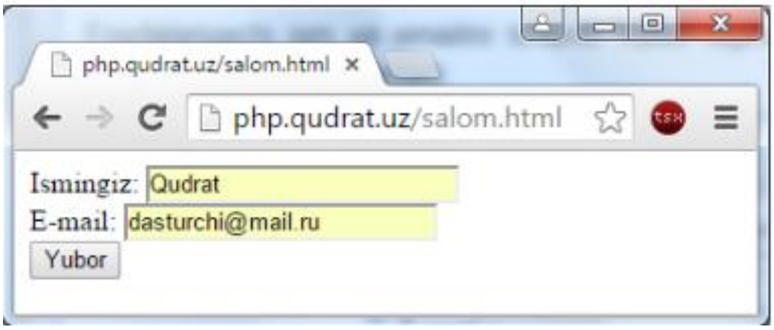
NAZARIY BO'LIM

Ishning maqsadi: PHP da GET va POST metodlari o'rganish va qo'llash;
Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:
Bilishi kerak – PHP da GET va POST metodlarining o'zoro farqini o'rganish va turli xil amallar bajarishni bilish;

$_GET va $_POST superglobal o'zgaruvchilari formalar bilan ishlash uchun ishlatiladi. Forma orqali ism va emailni yuboruvchi oddiy misolni ko'rib chiqamiz. salom.html faylini hosil qiling va unga quyidagilarni yozing.

Post metodidan foydalanish

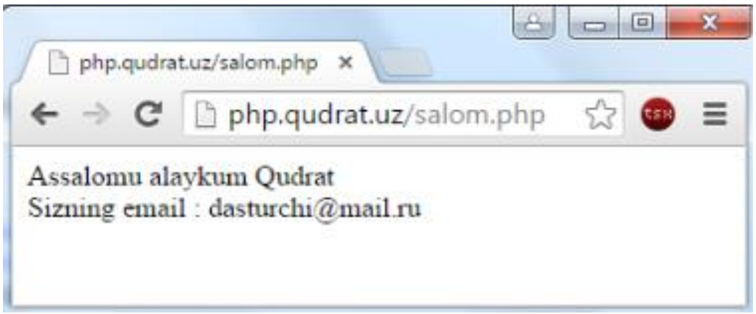
```
<html>
<body>
        <form action="salom-get.php" method="get">
        Ismingiz: <input type="text" name="name"><br>
        E-mail:   <input type="text" name="email"><br>
        <input type="submit" value="Yubor">
        </form>
</body>
</html>
<html>
<body>
        Assalomu alaykum <?php echo $_GET["name"]; ?><br>
        Sizning email : <?php echo $_GET["email"]; ?>
</body>
</html>
```



Foydalanuvchi ism va emailini to'ldirib Yubor tugmasini bosganida ma'lumotlar POST metodi orqali salom.php fayliga yuboriladi.

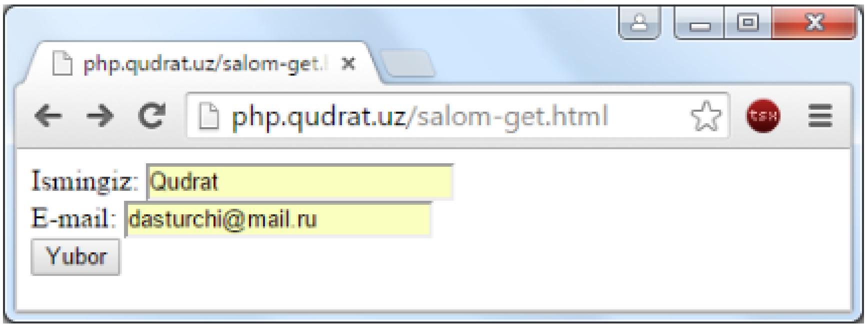
Ma'lumotlarni qabul qilib oluvchi salom.php fayli quyidagicha bo'ladi

```
<html>
<body>

        Assalomu alaykum <?php echo $_POST["name"]; ?><br>
        Sizning email : <?php echo $_POST["email"]; ?>

</body>
</html>
```

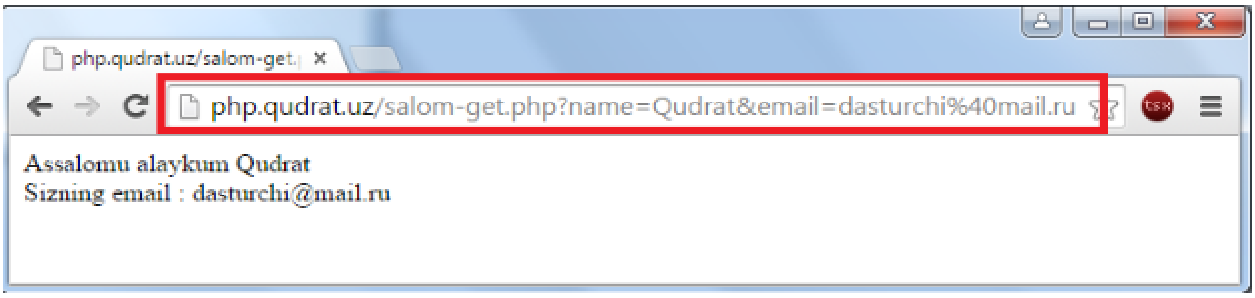


Ayni shu natijani GET metodi orqali olish quyidagicha bo'ladi. GET metodi orqali ma'lumot yuborish

```
<html>
<body>
        <form action="salom-get.php" method="get">
        Ismingiz: <input type="text" name="name"><br>
        E-mail:   <input type="text" name="email"><br>
        <input type="submit" value="Yubor">
        </form>
</body>
</html>
```



```
<html>
<body>

        Assalomu alaykum <?php echo $_GET["name"]; ?><br>
        Sizning email : <?php echo $_GET["email"]; ?>

</body>
</html>
```



Yuqorida keltirilgan namunalar tushuntirishi maqsadida sodda qilib keltirildi.

GET metodi orqali yuborilgan ma'lumotlarni istalgan odam ko'rishi mumkin.Chunki bu ma'lumotlar adreslar satrida yozilib turadi. Shuning uchun bu metod orqali parol va boshqa maxfiy ma'lumotlarni jo'natib bo'lmaydi. Bundan tashqari GET metodi orqali yuboriladigan ma'lumotlar hajmi chegaralangan, 2000 ta belgigacha jo'natish mumkin.

POST metodi orqali yuborilgan ma'lumotlarni boshqalar ko'ra olmaydi. Chunki bu ma'lumotlar URLda (adreslar satrida) yozilmaydi. Shuning uchun parol va boshqa maxfiy ma'lumotlar POST metod orqali serverga murojat POST metodi orqali odatda 8 MB ma’lumotni yuborish mumkin. Bu hajmni **php.ini** faylidagi **post_max_size** orqali o'zgartirish mumkin.

13-Laboratoriya ishi

Mavzu: PHP da forma elementarini tekshirish

NAZARIY BO'LIM

Ishning maqsadi: PHP da HTML formalari bilan ishlashni o'rganish va qo'llash;

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – Forma tashkillashtirish va PHP bilan bog'lashni o'rganish va amalda qo'llash;

Bu va bungan keyingi bir nechta mavzuda forma elemantlarim tekshirish masalalarini o'rganamiz.

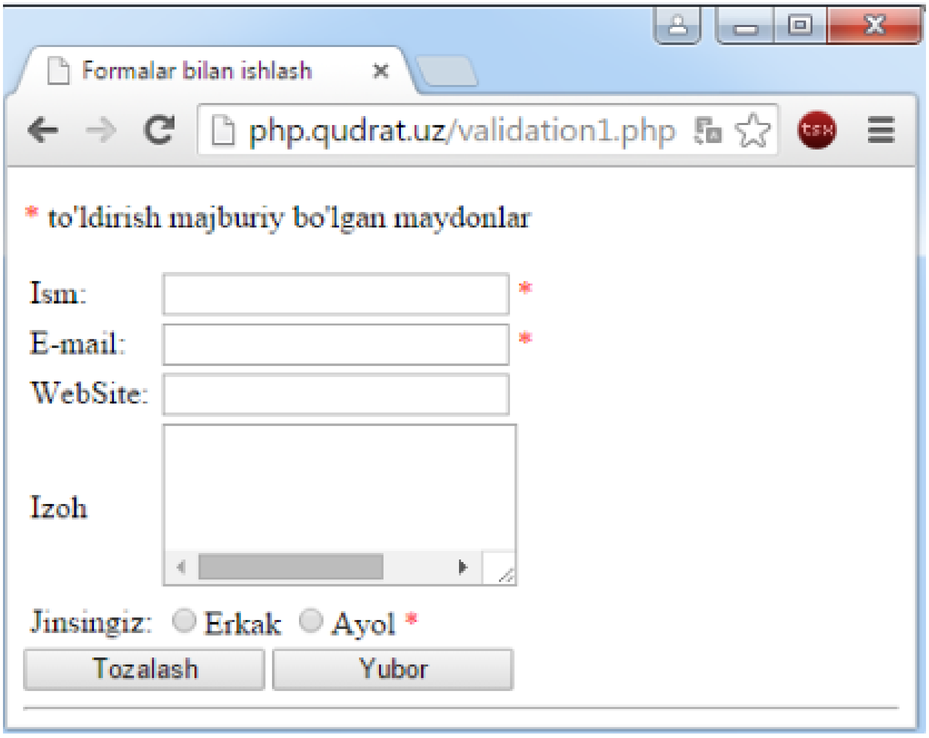
Yuqoridagi rasmda ko'rsatilgan formam hosil qiling. Buning uchun Ism, E-mail va Websitem kiritish uchun input tegining text xususiyatidan foydalanamiz. Izoh qoldirish uchun TEXTEREA ishlatiladi. Elementlar tekis chiqishi uchun iadvalga ioylashtiramiz.

Forma elementlarini tekshirish

```
<!DOCTYPE html>
<html>
<head>
	<title>Formalar bilan ishlash</title>
	<style type="text/css">
		.qizil{color: red;}
	</style>
</head>
<body>
	<?php
	// o'zgaruvchilarga boshlang'ich qiymat berish
	$nameErr = $emailErr = $genderErr = $websiteErr = "";
	$name = $email = $gender = $comment = $website = "";

	if ($_SERVER["REQUEST_METHOD"] == "POST") {

		if (empty($_POST["name"])) {
		$nameErr = "Ism kiritilmadi";
		} else {
		$name = test_input($_POST["name"]);
		}

		if (empty($_POST["email"])) {
		   $emailErr = "Email kiritilmadi";
```

```
                } else {
                  $email = test_input($_POST["email"]);
                }

                if (empty($_POST["website"])) {
                  $website = "";
                } else {
                  $website = test_input($_POST["website"]);
                }

                if (empty($_POST["comment"])) {
                  $comment = "";
                } else {
                  $comment = test_input($_POST["comment"]);
                }

                if (empty($_POST["gender"])) {
                  $genderErr = "Jins kiritilmadi";
                } else {
                  $gender = test_input($_POST["gender"]);
                }
        }

        function test_input($data) {
          $data = trim($data);
          $data = stripslashes($data);
          $data = htmlspecialchars($data);
          return $data;
        }
        ?>

    <form method="post" action="<?php
            echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

        <p><span class="qizil">*</span> to'ldirish majburiy
                    bo'lgan maydonlar</p>
        <table>
            <tr>
                <td>Ism:</td>
                <td><input type="text" name="name">
                    <span class="qizil">* <?php echo $nameErr;?>
</span></td>
            </tr>
            <tr>
                <td>E-mail:</td>
```

```
                    <td><input type="text" name="email">
                        <span class="qizil">* <?php echo
$emailErr;?></span></td>
                </tr>
                <tr>
                    <td>WebSite:</td>
                    <td><input type="text" name="website"></td>
                </tr>
                <tr>
                    <td>Izoh</td>
                    <td><textarea name="comment" rows="5"
cols="22"></textarea></td>
                </tr>
                <tr>
                    <td>Jinsingiz:</td>
                    <td> <input type="radio" name="gender"
value="male">Erkak
                        <input type="radio" name="gender"
value="female">Ayol
                        <span class="qizil">* <?php echo
$genderErr;?></span>
                    </td>
                </tr>
            </table>
            <input type="reset"  value="Tozalash" style="width:120px">
            <input type="submit" value="Yubor" style="width:120px">
    </form>
            <hr>

    <?php
            echo $name.'<br>';
            echo $email.'<br>';
            echo $website.'<br>';
            echo $comment.'<br>';
            echo $gender.'<br>';
    ?>

    </body>
    </html>
```


Formalar bilan ishlash
php.qudrat.uz/validation1.php
* to'ldirish majburiy bo'lgan maydonlar
Ism: *
E-mail: *
WebSite:
Izoh
Jinsingiz: Erkak Ayol *
Tozalash
Yubor

15-Laboratoriya ishi

Mavzu: Email va URLni tekshirish

Ishning maqsadi: PHP da HTML formalari bilan ishlashni o'rganish va Email va URLni tekshirish;

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – Forma tashkillashtirish va PHP bilan bog'lashni o'rganish va Email va URLni tekshirishni amalda qo'llash;

Bu mavzuda ism, email va sayt nomini to'g'ri kiritilganligini tekshirishni o'rganamiz. Ism faqat harflardan va probeldan iborat bo'lishi lozim. Quyidagi kod shuni tekshiradi

Email, ism, urlni tekshirish

```
<!DOCTYPE html>
<html>
<head>
	<title>Formalar bilan ishlash</title>
	<style type="text/css">
		.qizil{color: red;}
	</style>
</head>
<body>
<form method="post" action="<?php
		echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

	<p><span class="qizil">*</span> to'ldirish majburiy
				bo'lgan maydonlar</p>
	<table>
		<tr>
			<td>Ism:</td>
			<td><input type="text" name="name">
				<span class="qizil">*</span></td>
		</tr>
		<tr>
			<td>E-mail:</td>
			<td><input type="text" name="email">
				<span class="qizil">*</span></td>
		</tr>
		<tr>
			<td>WebSite:</td>
			<td><input type="text" name="website"></td>
		</tr>
```

```
                <tr>
                    <td>Izoh</td>
                    <td><textarea name="comment" rows="5" 
cols="22"></textarea></td>
                </tr>
                <tr>
                    <td>Jinsingiz:</td>
                    <td> <input type="radio" name="gender" 
value="male">Erkak
                        <input type="radio" name="gender" 
value="female">Ayol
                        <span class="qizil">*</span>
                    </td>
                </tr>
            </table>
            <input type="reset"  value="Tozalash" style="width:120px">
            <input type="submit" value="Yubor" style="width:120px">
        </form>
            <hr>
            <?php
            // o'zgaruvchilarga boshlang'ich qiymat berish
            $name = $email = $gender = $comment = $website = "";

            if ($_SERVER["REQUEST_METHOD"] == "POST") {
              $name       = test_input($_POST["name"]);
              $email      = test_input($_POST["email"]);
              $website    = test_input($_POST["website"]);
              $comment  = test_input($_POST["comment"]);
              $gender     = test_input($_POST["gender"]);

              echo $name.'<br>';
              echo $email.'<br>';
              echo $website.'<br>';
              echo $comment.'<br>';
              echo $gender.'<br>';
            }
            function test_input($data) {
              $data = trim($data);
              $data = stripslashes($data);
              $data = htmlspecialchars($data);
              return $data;
            }
            ?>
        </body>
        </html>
```

16-Laboratoriya ishi

Mavzu: Forma elementlari qiymati

Ishning maqsadi: PHP da HTML formalari bilan ishlashni o'rganish;

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – Forma tashkillashtirish va PHP bilan bog'lashni o'rganish va Email va URLni tekshirishni amalda qo'llash;

Bu mavzuda formaga kiritilgan qiymatlar o'chib ketmasdan o'z joyida qoldirishni o'rganamiz. Chunki bir nechta narsani kiritganda foydalanuvchi faqat bittasida xato qilishi mumkin. Lekin foydalanuvchidan hamma narsani qaytadan kiritishni talab qilish juda noqulay.

Demak scriptga kichkina kod qo'shamiz. Ya'ni input tegining value qiymatiga php kod yozamiz.

Elementlar qiymatlarini saqlab qolish

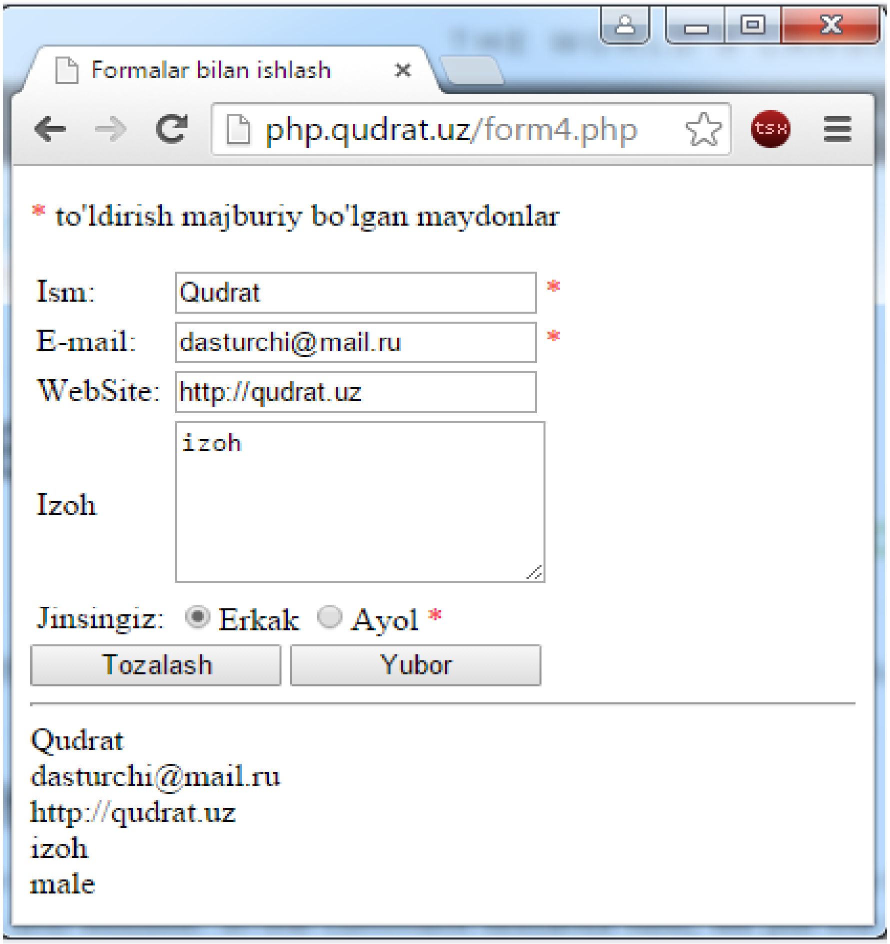
PHPda fayllarni yuklash

PHPda fayllarni serverga yuklash ancha oson. Faqat osonlik bilan birga xavf ham birga keladi. Birinchi o'rinda php.ini fayllda file_uploads hususiyat yoqilgan bo'lishi lozim.

```
<!DOCTYPE html>
<html>
<head>
	<title>Formalar bilan ishlash</title>
	<style type="text/css">
		.qizil{color: red;}
	</style>
</head>
<body>
<form method="post" action="<?php
		echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

	<p><span class="qizil">*</span> to'ldirish majburiy
				bo'lgan maydonlar</p>
	<table>
		<tr>
			<td>Ism:</td>
			<td><input type="text" name="name">
				<span class="qizil">*</span></td>
		</tr>
		<tr>
			<td>E-mail:</td>
			<td><input type="text" name="email">
				<span class="qizil">*</span></td>
```

```
				</tr>
				<tr>
					<td>WebSite:</td>
					<td><input type="text" name="website"></td>
				</tr>
				<tr>
					<td>Izoh</td>
					<td><textarea name="comment" rows="5" 
cols="22"></textarea></td>
				</tr>
				<tr>
					<td>Jinsingiz:</td>
					<td> <input type="radio" name="gender" 
value="male">Erkak
						<input type="radio" name="gender" 
value="female">Ayol
						<span class="qizil">*</span>
					</td>
				</tr>
			</table>
			<input type="reset"  value="Tozalash" style="width:120px">
			<input type="submit" value="Yubor" style="width:120px">
		</form>
			<hr>
			<?php
			// o'zgaruvchilarga boshlang'ich qiymat berish
			$name = $email = $gender = $comment = $website = "";

			if ($_SERVER["REQUEST_METHOD"] == "POST") {
			  $name       = test_input($_POST["name"]);
			  $email      = test_input($_POST["email"]);
			  $website    = test_input($_POST["website"]);
			  $comment  = test_input($_POST["comment"]);
			  $gender     = test_input($_POST["gender"]);

			  echo $name.'<br>';
			  echo $email.'<br>';
			  echo $website.'<br>';
			  echo $comment.'<br>';
			  echo $gender.'<br>';
			}

			function test_input($data) {
			  $data = trim($data);
			  $data = stripslashes($data);
```

```
            $data = htmlspecialchars($data);
            return $data;
          }
          ?>
</body>
</html>
```



17-Laboratoriya ishi

Mavzu: PHPda fayllarni yuklash

Ishning maqsadi: PHP da fayllarni yuklashni o'rganish;

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – Forma tashkillashtirish va PHP bilan bog'lashni o'rganish va fayllarni yuklashni amalda qo'llash;

PHPda fayllarni serverga yuklash ancha oson. Faqat osonlik bilan birga xavf ham birga keladi. Birinchi o'rinda php.ini faylida **file_uploads** hususiyati yoqilgan bo'lishi lozim.

file_uploads =on

HTML forma hosil qilish

Dastlab foydalanuvchilar rasm yuklashi uchun forma hosil qilamiz.

```
< SDOCTYPE html>
<html>
<body>
<form action=" upload.php" method="post" enctype="multipart/form-data">
Serverga yuklash uchun rasm tanlang:
<input type= "File " name="fileToUpload" id="fileToUpload">
<br>
<input type="submit" value="Rasmni yuborish" name="submit">
</body>
</html>
```

Bu HTMLforma uchun quyidagi qoidalar mavjud. Bularsiz faylni serverga yuklab bo'lmaydi.

1. Fayllar POST metodi orqali yuklanadi:
2. Forma qo'shimcha **enctype="multipart/from-data"** hususiyatiga ega bo'ladi;

Yana shuni aytish lozimki input tegidagi **type="file"** hususiyati orqali foydalanuvchi

kompyuteridan rasm tanlash mumkin bo'lsa funksiya chaqiriladi.

Bu forma orqali ma'lumotlar **upload.php** fayliga yuboriladi. Bu fayl quyida keltirilgan.

```
<!DOCTYPE html>
<html>
<body>
<form action="upload.php" method="post" enctype="multipart/form-
data">
    Serverga yuklash uchun rasm tanlang:
    <input type="file" name="fileToUpload" id="fileToUpload"><br>
    <input type="submit" value="Rasmm yuborish" name="submit">
```

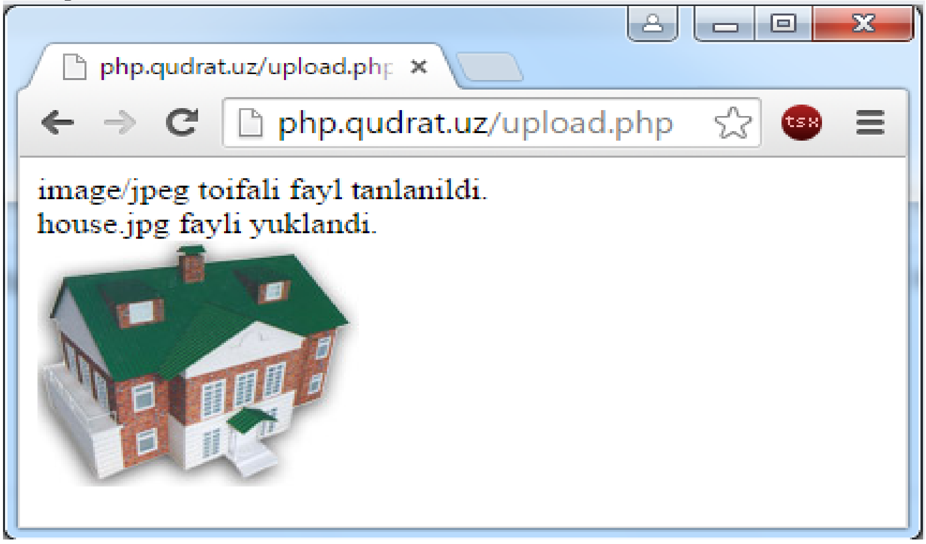
```
</form>
</body>
</html>
```

Upload fayli

```
<!DOCTYPE html>
<html>
<body>
<form action="upload.php" method="post" enctype="multipart/form-
data">
    Serverga yuklash uchun rasm tanlang:
    <input type="file" name="fileToUpload" id="fileToUpload"><br>
    <input type="submit" value="Rasmm yuborish" name="submit">
</form>
</body>
</html>
<?php
    $target_dir = "uploads/";
    $target_file =
$target_dir.basename($_FILES["fileToUpload"]["name"]);
    $uploadOk = 1;
    $imageFileType = pathinfo($target_file, PATHINFO_EXTENSION);

    if(isset($_POST["submit"])) {
        // faylni rasmligim tekshiramiz
        $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
        if($check != false) {
            echo $check["mime"] . " toifali fayl tanlamldi.";
            $uploadOk = 1;
        } else {
            echo "Siz tanlagan fayl rasm emas.";
            $uploadOk = 0;
        }
    }
    echo "<br>";
    // Bunday fayl oldindan bor yo'qligini takshirish
    if (file_exists($target_file)) {
        echo "Kechirasiz, bunday fayl bor.<br>";
        $uploadOk = 0;
    }

    // Fayl xajmim tekshirish
    // 500 kb dan katta bo'sa
    if ($_FILES["fileToUpload"]["size"] > 500000) {
        echo "Kechirasiz sizning fayl juda katta.<br>";
```

```
        $uploadOk = 0;
    }
    // Faylga amq formatm belgilash
    if($imageFileType != "jpg" && $imageFileType != "png"
    && $imageFileType != "jpeg" && $imageFileType != "gif" ) {
        echo "Kechirasiz, faqat JPG, JPEG, PNG & GIF
        kengaytmali fayllarni yuklash mumkin.<br>";
        $uploadOk = 0;
    }
        // yuklashda biror xatolik sodir bo'lgan bo'lsa
    if ($uploadOk == 0) {
        echo "Kechirasiz, sizning fayl yuklanmadi.<br>";
    // agar xammasi joyida bo'lsa, faylni yuklashga harakat qilarniz
    } else {
        if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$target_file)) {
            echo basename($_FILES["fileToUpload"]["name"]). " fayli
yuklandi.<br>";
            echo "<img src='".$target_file."'>";
        } else {
            echo "Kechirasiz faylni yuklashda xatolik sodir bo`ldi.";
        }
}
?>
```

Natijasi:



18-Laboratoriya ishi

Mavzu: PHP da include va require funksiyalar

<u>Ishning maqsadi</u>: PHP da include va require funksiyalarni qo’llashni o’rganish;

<u>Ishning natijasida talaba quyidagilarga ega bo’lishi kerak</u>:

Bilishi kerak – include va require funksiyalarini farqini o’rganish va amalda qo’llash;

Nazariy qism

Server Side Incudes (SSI)

Server biror PHP faylini ishga tushirishidan oldin, boshqa PHP fayllarini shu faylga qo'shish mumkin. Buning uchun **include**() Yoki **require**() funksiyalari ishlatiladi. Bu funksiyalar bir xil vazifani bajaradi. Faqat xatoliklarni ushlashda farq qiladi.

1.**include**() funksiyasida xatolik haqida xabar chiqariladi va sicript kelgan joyidan davom etadi.

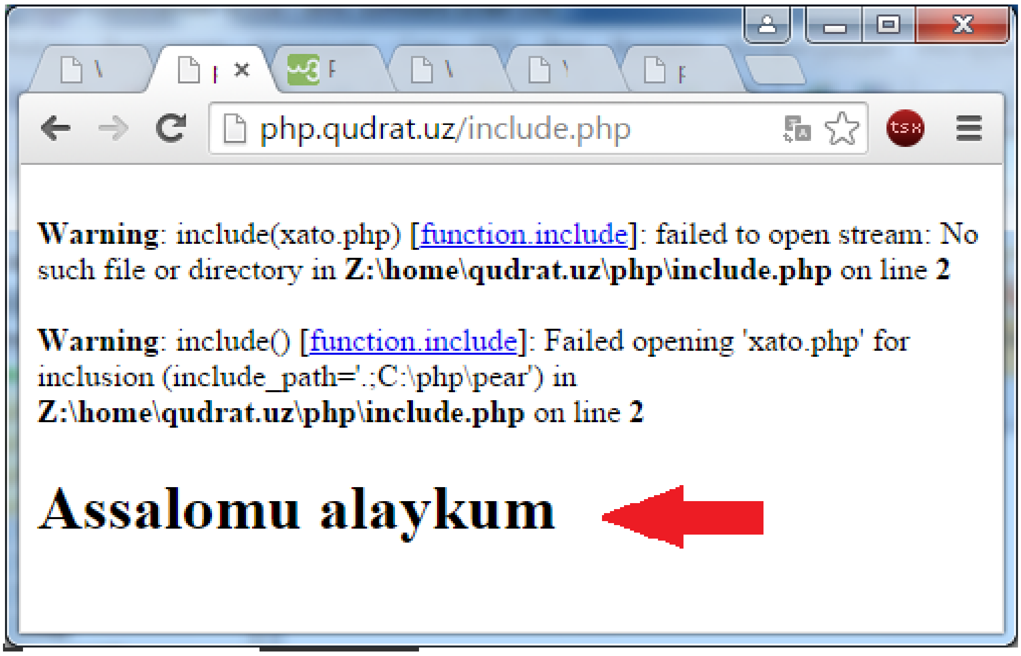
2.**require()** funksiyada fatal xatolik sodir bo'ladi va script bajarilishi to'xtatiladi.

Harikkala tunksiya ham funksiyalar hosil qilish va har xil sahifalarda chaqirish, header, footer, menu va boshqalarni hosil qilishda ishlatiladi.

Buning ma'nosi, siz saytingiz uchun standart header, footer va menu hosil qilasiz. Agar header faylini yangilash lozim bo’lsa, bir faylini o'zgartirish kifoya.

Include() funksivasiga na’muna

```
<?php
	include ("xato.php");
	echo "<h1>Assalomu alaykum</h1>";
?>
```



Include() funksiyasi orqali **xato.php** fayliga murojaat qilinadi. Bu fayl asli yo'q. Shuning uchun xatolik so dir bo'ladi. Xatolik sodir bo'lganligi (fayl topilmaganlig) haqida xabar chiqariladi. Lekin script ishlashda davom etib, include() funksiyasidan keyingi echo funksiyasi orqali "Assalomu alaykum"yozuvi chiqariladi.

Require() funksiyasiga na'muna

```
<?php
	require ("xato.php");
	echo "<h1>Assalomu alaykum</h1>";

?>
```

require() funksiyasi orqali **xato.php** fayliga munojaat qilinadi. Bu fayl asli yo'q. Shuning uchun xatolik sodir bo'ladi. Xatolik sodir bo'lganligi (fayl topilmaganligi) haqida xabar chiqariladi va script ishlashi tugatiladi. Ya'ni, require() funksiyasidan keyingi echo funksiyasi bajarilmaydi. ("Assalomu alaykum" yozuvi chiqarilmaydi) soddaroq qilib aytganda include - faylni qo'shib olish, require esa talab qilish. Ya'ni require faylni to'g'ri i to'liq ishlashini talab qiladi.

include_once() - funksiyasi

Siz o'zingiz uchun zarur bo'lgan funksiyalarni to'plab kutubxona tashkil qilishingiz mumkin. Masalan, funksiyalar to'plarnini mylib.php deb nomladingiz. Bu faylni istalgan php fayliga include funksiyasi orqali qo'shib funksiyalaridan foydalanishingiz mumkin. Lekin siz bir nechta faylda mylib.php ga murojaat qilsangiz va bu fayllar o'z navbatda boshqa php faylni include orqali qo'shib olishga harakat qilinsa xatolik sodir bo'ladi. Chunki, bu yerda funksiyalar 2 - marta takror e'lon qilinadi. Bunday o'rinlarda **include()** funksiyasi o'rniga **include_once()** funksiyasi ishlatiladi. Ya'ni, include_once() funksiyasi faylni bir

marta qo'shish uchun ishlatiladi. include() funksiyasi o’rniga doim include_once() funksiyasini ishlatish ham tavsiya etladi. Shunda faylni qayta qo'shish xatoligi sodir bo'lmaydi. require_once() - funksiyasi ham shunday ishlaydi.

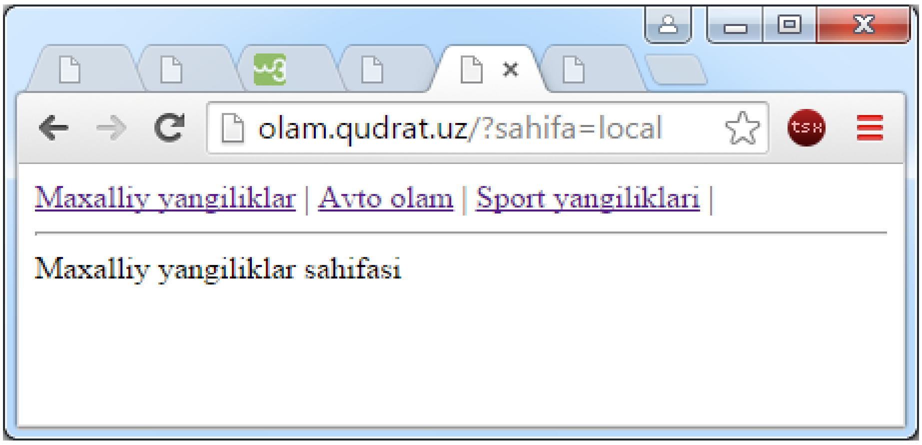
include va get funsiyalarining birga qo’llanilishi

Maxalliy yangiliklar, avto olam va sport yangiliklari menyusiga ega bo'lsa kichik bir namunani ko'rib chiqamiz. Buning uchun index.php, local.php, car.php va sport.php fayllarini ishlab chiqamiz index.phpfayli orqali kerak bo'lgan faylni include() funksiyasi orqali yuklab olarniz

index.php fayli

```
	<!DOCTYPE html>
<html>
<head>
	<title>Yangiliklar olarni</title>
</head>
<body>
	<a href="?sahifa=local">Maxalliy yangiliklar</a> |
	<a href="?sahifa=car">Avto olam</a> |
	<a href="?sahifa=sport">Sport yangiliklari</a> |
	<hr>

	<?php

		if (isset($_GET['sahifa']))
		{
			$sahifa = $_GET['sahifa'];

			if ($sahifa == 'local')
				include "local.php";

			elseif ($sahifa == 'car')
				include "car.php";

			elseif ($sahifa == 'sport')
				include "sport.php";
		}
	?>
</body>
</html>
```

Natijasi:


olam.qudrat.uz/?sahifa=local
Maxalliy yangiliklar | Avto olam | Sport yangiliklari |
Maxalliy yangiliklar sahifasi

Qo’shimcha o’rganish uchun mavzular

Mavzu: PHPda Cookielar bilan ishlash

HTTP protokoli uzoq ulanishlarga ega emas. Ya’ni har bir sahifani chaqirish uchun server bilan alohida ulanish sodir bo'ladi. Shuning uchun sahifalar umumiy parametriga ega bo'la olmaydi va bir biri bilan bog'lana olmaydi. Saytning umumiy parametrlari qiymatini saqlashning 3 xil usuli bor:

1. Cookie -foydalanuvchi kompyuterida saqlanuvchi fayl.
2. Session - sessiyalar.
3. Web dasturchi tomonidan mustaqil ravishda ishlab chiqiladi. Zarur parametrlar ma'lumotlar bazasida saqlanadi *va* foydalanuvchiga bog'lab qo'yiladi. Yoki formalar orqali bir sahifadan boshqasiga qiymatlar yuboriladi.

Cookie ikki xil bo'lishi mumkin.

1.1. Vaqtinchalik - bunda parametrlar foydalanuvchining kompyuterida server bilan ma'lum bir bo’glanish olarig'ida saqlaniladi.

1.2. Doimiy- bunda parametrlar foydalanuvchining kompyuterida web dasturchi tomonidan ko'rsatilgan vaqtgacha saqlaniladi.

Cookie qanday hosil qilinadi?

Setcookie() - funksiyasi orqali cookie o'matiladi. Bu funksiyani <HTML> tegidan oldin yozish shart. Aks holda xatolik sodir bo'ladi.

int setcookie(cookiename, value, lifetime, path, domain, secure);
string cookiename - cookie nomi
string value - qiymati
integer lifetime - yashash vaqti sekundlarda yoziladi
string path - yo'l
string domain - domen
integer secure - xafvsizlik parametri. Odatda nolga teng bo'ladi. Agar bir qo'yilsa, HTTP prortokoli orqali murojaat qilib bo'lmaydi.

10 sekund yashaydigan cookie fayli

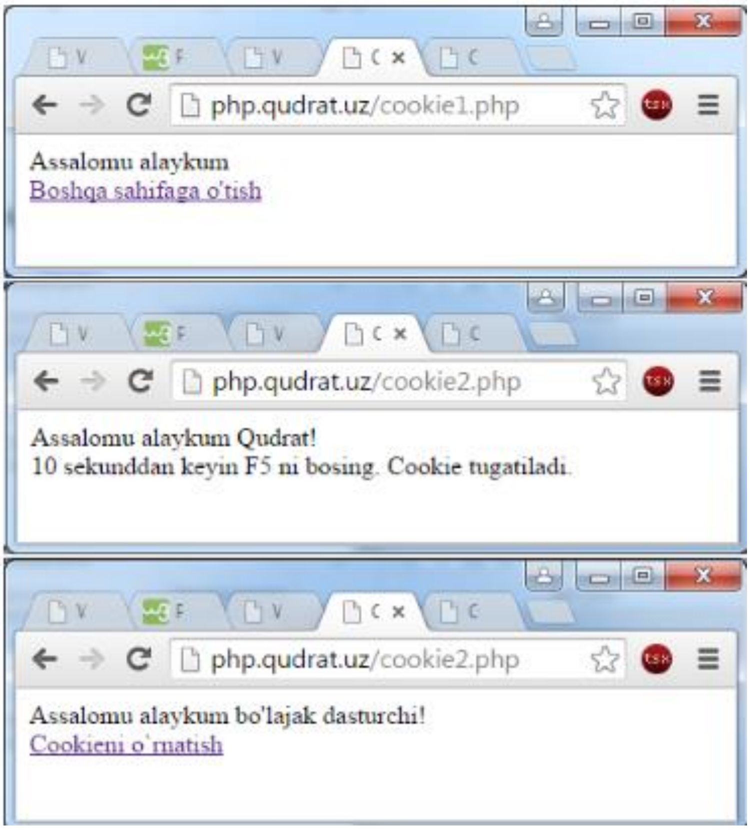
```
<?php
        $lifetime = time() + 10; // 10 sekund yashaydi
        setcookie("user", "Qudrat", $lifetime);
?>
<!DOCTYPE html>
<html>
<head>
        <title>Cookiega misol</title>
</head>
<body>
        <?php
                echo "Assalomu alaykum ".$_COOKIE["user"];
        ?>

        <br>
```

```
<a href="cookie2.php">Boshqa sahifaga o'tish</a>
</body>
</html>
```

Hosil qilingan cookiega murojat qilarnish. Isset funksiyasi orqali cookie tugatilganligi tekshiriladi

```
<!DOCTYPE html>
<html>
<head>
	<title>Cookiega misol</title>
</head>
<body>
<?php
	if (isset($_COOKIE["user"]))
	{
		echo "Assalomu alaykum " . $_COOKIE["user"] . "!<br />";
		echo "10 sekunddan keyin F5ni bosing. Cookie tugatiladi.";
	}
	else
	{
		echo "Assalomu alaykum bo'lajak dasturchi!<br/>";
		echo '<a href="cookie1.php">Cookiem o`rnatish</a>';
	}
?>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
	<title>Cookiega misol</title>
</head>
<body>
	<?php
		echo "Assalomu alaykum ".$_COOKIE["user"];
	?>

	<br>
	<a href="cookie2.php">Boshqa sahifaga o'tish</a>
</body>
</html>
```



cookiem o'chirish

```
<?php
	// cookiem o'chirish uchun vaqtm kamaytirish lozim
	setcookie("user", "", time() - 3600);
	echo "cookie o`chirildi";
?>
```

Agar foydalanuvchi browserida cookie o'chirilgan bo'lsa. dasturchi mustaqil ravishda o'zgaruvchilarni sahifadan - sahifaga olib o'ishi mumkin. Buning uchun farmalardanfoydalaniladi.

Cookie xavfsizligi

Cookieda umuman xafvsizlik yo'q. Istalgan odam kompyutendagi cookielarni o'qishi va o'zgattirishi mumkin. Bu degani cookiedan umuman fcydalanmang degani emas. Har bir narsaning imkomyaidan boricha foydalanish kerak. Cookieda foydalanuvchining paroli Yoki boshqa biror maxfiy ma'lumoiarm saqlab bo'lmaydi. Lekin, boshqa birovlar ko'rganida va xatto o'zgarlirganida ham sayt xafvsizligiga taxdid solmaydigan Ma’lumotlarni saqlash mumkin. Masalan. foydalanuvchi o'zi uchun tanlagan rang Yoki temam saqlash mumkin. Bum boshqa birovbilganida hamhechmma o'zgarmaydi.

PHPda ma'lumotlarni filtirlash

PHP filtirlari kiruvchi ma'lumotlarni tekshirish va tozalash uchun ishlatiladi. *Validating* data - kiruvchi ma'lumotlarni to'g'riligini tekshirish *Samtizing* data - kiruvchi ma'lumotlarni ortiqcha kodlardan tozalash

filter_list - funksiyasi orqali filterlamshi mumkin bo'lgan elementlarni ko'rish mumkin

```
<table border="1">
  <tr>
    <td>Filter Name</td>
    <td>Filter ID</td>
  </tr>
  <?php
  foreach (filter_list() as $id =>$filter) {
      echo '<tr><td>' . $filter . '</td><td>'.
      filter_id($filter) . '</td></tr>';
  }
  ?>
</table>
```

Filter name	Filter ID
INT	257
BOOLEAN	253
FLOAT	259
VALIDATE_REGEXP	272
VALIDATE URL	273
VALIDATE_EMAIL	274
VALIDATE_IP	275
STRING	513
STRIPPED	513
ENCODED	514
SPECIAL_CHARS	515
FULL_SPECIALCHARE	522
UNSAFE RAW	*516*
EMAIL	517
URL	518
NUMBER INT	519
NUMBER_FLOAT	520
MAGIC_QUOTES	521
CALLBACK	1024

Filterlar nima uchun ishlatiladi?

Katta saytlarda tashqi kiruvchi ma'lumotlar bo'ladi. Kiruvchi ma'lumotlar quyidagicha bo'lishi mumkin.

- Foydalanuvchilar forma orqali kiritishi

- Cookies
- Webxizmatlari ma'lumotlari
- Server
- o'zgaruvchilari
- Ma'lumotlar bazasi so'rovlari

➢ Siz har doim kiruvchi ma'lumotlarni to’g’riligini tekshirishingiz va tozalashingiz shart.

➢ Noto'g'ri kiritilgan ma'lumotlar sayt xavfsizligi muammolarini keltirib chiqarishi va saytni buzishi mumkin.

➢ PHP filterlardan foydalangan holda to’g’ri ma'lumotlarni kiritishingiz mumkin.

➢ **Filter_var() funksiyasi ma'lumotlarni tozalash va tekshirish uchun ishlatiladi.**

filter orqali satrdagi barcha html kodlarni o'chirish

```
<?php
        $str = "<h1>Hello World!</h1>";
        // firter orqali barcha html kodlarni o'chirish
        $newstr = filter_var($str, FILTER_SAMTIZE_STRING);
        echo $str;
        echo $newstr;

?>
```

Butun son kiritilganligini tekshirish

```
<?php
  if (isset($_GET['a'])) {
        $int = $_GET['a'];

        if (!filter_var($int, FILTER_VALIDATE_INT) === false) {
           echo("Butun son kiritildi");
        } else {
           echo("Butun son kiritilmadi");
        }
  }
  else  {
        echo "<a href='?a=5'>5ni kiritish</a>";
  }

?>
```

Ipni tekshirish

```
<?php
```

```
	$ip = "127.0.0.1";

	if (!filter_var($ip, FILTER_VALIDATE_IP) === false) {
	   echo("$ip is a valid IP address");
	} else {
	   echo("$ip is not a valid IP address");
	}
?>
```

e-mailm tekshirish

```
<?php
	$email = "dasturchi@mail.ru";

	// emaildan barcha ortiqcha belgilarni o'chirish
	$email = filter_var($email, FILTER_SAMTIZE_EMAIL);

	// e-mailm tekshirish
	if (!filter_var($email, FILTER_VALIDATE_EMAIL) === false) {
	   echo("$email to`g`ri email");
	} else {
	   echo("$email noto`g`ri email");
	}
?>
```

Urlni tekshirish

```
<?php
	$url = "http://jasur.uz";

	// urldan barcha ortiqcha belgilarni o'chirish
	$url = filter_var($url, FILTER_SAMTIZE_URL);

	// urlni tekshirish
	if (!filter_var($url, FILTER_VALIDATE_URL) === false) {
	   echo("$url to`g`ri URL");
	} else {
	   echo("$url no to`g`ri URL");
	}

?>
```

19-Laboratoriya ishi

Mavzu: MySQL ga ulanish

Ishning maqsadi: PHP da ma’lumotlar bazasini hosil va ulanishni qilishni o’rganish;

Ishning natijasida talaba quyidagilarga ega bo’lishi kerak:

Bilishi kerak – My SQL so’rovlarini o’rganish va amalda qo’llash;

Nazariy qism

PHP 5 va undan yuqori versiyalarida MySQL bilan ishlashning 2 xil usuli bor

1. MySQLi (improved) -takomillashgan MySQL
2. PDO (PHP Data Objecte}

Bundan oldingi versiyalarda MySQL bilan ishlovchi boshqa metodlar mavjud. PHP 7 dan boshlab eskirgan ko'pgina funksiyalar olib tashlaniladi. Shuning uchun biz MySQLi bilan ishlashni o'rgarish bilan kifoyalanamiz.

PDO orqali 12 ta ma'lumotlar bazasi bilan ishlash mumkin. MySQLi da esa faqat MySQL bilan ishlash mumkia

MyS QLi orqali MySQL bilan ishlashning 2 xil usuli bor.

1. MySQLi (object-oriented)
2. MySQLi (procedural)

MySQLi object-oriented connection

```
<?php
	$dbhost  = 'localhost';    // o'zgartirish shart emas
	$dbname  = 'dars';                // Ma'lumotlar bazasi nomi
	$dbuser  = 'root';                // user nomi
	$dbpass  = '';                    // parol

	// server bilan aloqa o'rnatish
	$connection = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

	if ($connection->connect_error) {
	   die("MySQLga ulanishda xatolik sodir bo`ldi: ".
		$connection->connect_error);
	}
	else
		echo "MySQL bilan aloqa o`rnatildi";

	// aloqani uzish
	$connection->close();

?>
```

MySQL prosedurali connection

```
<?php
	$dbhost  = 'localhost';    // o'zgartirish shart emas
	$dbname  = 'dars';                 // Ma'lumotlar bazasi nomi
	$dbuser  = 'root';                 // user nomi
	$dbpass  = '';                     // parol

	// server bilan aloqa o'rnatish
	$connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);

	if (!$connection) {
	    die("MySQLga ulanishda xatolik sodir bo`ldi: ".
	         mysqli_connect_error());
	}
	else
	         echo "MySQL bilan aloqa o`rnatildi";

	mysqli_close($connection);

?>
```

20-Laboratoriya ishi

Mavzu: MySQLda ma'lumotlar bazasini hosil qilish

Ishning maqsadi: PHP da ma’lumotlar bazasini hosil qilishni o’rganish;

Ishning natijasida talaba quyidagilarga ega bo’lishi kerak:

Bilishi kerak – My SQL so’rovlarini o’rganish va amalda qo’llash;

Nazariy qism

Php MyAdmin orqali MySQL ma'lumotlar bazasini boshqarish, jadvallarni hosil qilish. o'chirish, yangilash va hokazolarni siz bilan o'rganamiz.

Ma'lumotar bazasi bir yoki bir nechta jadvallardan tashkil topishi mumkin. Quyidagi misol myDB nomli ma'lumotar bazasini hosil qiladi.

database. php-fayli

```
<?php
        $dbhost  = 'localhost';    // o'zgartirish shart emas
        $dbuser  = 'root';                  // user nomi
        $dbpass  = '';                      // parol

        // server bilan aloqa o'rnatish
        $connection = new mysqli($dbhost, $dbuser, $dbpass);

        if ($connection->connect_error) {
           die("MySQLga ulanishda xatolik sodir bo`ldi: ".
                $connection->connect_error);
        }
        else
                echo "MySQL bilan aloqa o`rnatildi<br>";

        // myDB nomli bazani hosil qilish
        $sorov = "CREATE DATABASE myDB";
        if ($connection->query($sorov) === TRUE) {
           echo "Ma`lumotlar bazasi hosil qilindi";
        } else {
           echo "Ma`lumotlar bazasini hosil qilishda xatolik sodir bo`ldi:".
           $connection->error;
        }

        // aloqani uzish
        $connection->close();
?>
```

21-Laboratoriya ishi

Mavzu: MySQL da jadval hosil qilish

Ishning maqsadi: PHP da ma'lumotlar bazasini bilan HTML da yaratilgan jadvalni bog'lash va tahrirlashni o'rganish;

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – HTML da jadvalni hosil qilish va tahrirlashni bilish;

Nazariy qism

Ushbu mavzuda jadvalni CREATE TABLE orqali hosil qilishni o'nganamiz Quyidagi misolda E ta maydonli "MyGuests" nomli jadval hosil qilinadi. Jadval maydonlari "id", "firstnam e", "lastna me", "emair va "reg_date".

CREATE TABLE orqali jadval hosil qlish

```
<?php
        $dbhost  = 'localhost';    // o'zgartirish shart emas
        $dbname  = 'mydb';               // Ma'lumotlar bazasi nomi
        $dbuser  = 'root';               // user nomi
        $dbpass  = '';                   // parol
        // server bilan aloqa o'rnatish
        $connection = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
        if ($connection->connect_error) {
           die("MySQLga ulanishda xatolik sodir bo`ldi: ".
                $connection->connect_error);
        }
        else
                echo "MySQL bilan aloqa o`rnatildi<br>";
        // Jadval hosil qilish SQL so'rovi
        $sorov = "CREATE TABLE MyGuests (
        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP)";

if ($connection->query($sorov) === TRUE) {
           echo "MyGuests jadvali hosil qilindi";
        } else {
           echo "Jadvalni hosil qilishda xatolik: " .
           $conn->error;
        }

        // aloqani uzish
        $connection->close();
?>
```

22-Laboratoriya ishi

Mavzu: MySQL dan ma 'lumotlarni o'qish

Ishning maqsadi: PHP da ma’lumotlar bazasini bilan HTML da yaratilgan jadvalni bog’lash va tahrirlashni o’rganish;

Ishning natijasida talaba quyidagilarga ega bo’lishi kerak:

Bilishi kerak – HTML da jadvalni hosil qilish va tahrirlashni bilish;

Nazariy qism

Bu va bundan keyingi bir nechta darsda "dars" nomli ma'lumotlar bazasi bilan ishlaymiz. "dars" nomli ma'lumotlar bazasini phpMyAdmin orqali hosil qiling va unga *aforizm.sql* jadvalini import qiling. aforizm.sql fayli ushbu mavzu bilan bir xil bo'lgan katalogda joylashgan.

dastlab config.php fayli ni hosil qilib olamiz va unga keyingi mavzularda ham murojaat qilamiz.

config.php faylida server bilan aloqa, so'rovlarni amalga oshirish uchun *queryMysql* funksiyasi va kiruvchi ma'lumotlarni tekshirish uchun *test_input* funksiyasi berilgan.

Keyingi mavzularda require_once(`config.php`); deganda aynan ushbu fayl nazarda tutiladi. Bu fayl keyingi mavzuga takror qo'yilmaydi.

config.php fayli

```
<?php
	$dbhost  = 'localhost';		// o'zgartirish shart emas
	$dbname  = 'dars';			// Ma'lumotlar bazasi nomi
	$dbuser  = 'root';			// user nomi
	$dbpass  = '';			// parol

	// server bilan aloqa o'rnatish
	$connection = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
	if ($connection->connect_error) {
	   die("MySQLga ulanishda xatolik sodir bo`ldi: " . $connection->connect_error);
	}

	function queryMysql($query){
		global $connection;

	   $result = $connection->query($query);
	   if (!$result)
		die("So`rovda xatolik bor. ".$connection->error);

	   return $result;
	}
```

```
        function test_input($data) {
          $data = trim($data);
          $data = stripslashes($data);
          //$data = htmlspecialchars($data, ENT_QUOTES);
          return $data;
        }
?>
```

Mysql da ma'lumotlarni o'qish

```
<?php
        require_once('config.php');

        $sorov = 'select * from aforizm';

        $result = queryMysql($sorov);

        // yozuvlar sonini aniqlash
        $soni = $result->num_rows;

        /*
                MYSQLI_BOTH
                MYSQLI_ASSOC
                MYSQLI_NUM
        */
        for ($i = 0; $i < $soni; $i++) {
                $row = $result->fetch_array(MYSQLI_ASSOC);

                print_r($row);
        }

        // aloqani uzish
        $connection->close();

?>
```

While sikl orqali ma'lumotlarni o'qish

```
<?php
        require_once('config.php');
        $sorov = 'select * from aforizm';
        $result = queryMysql($sorov);
        while($row = $result->fetch_array(MYSQLI_ASSOC)){
        echo "<h3>".$row['matni'].' ('.$row['muallif'].')</h3>';
        }
        $connection->close();
?>
```

Aforizmni tasodifiy tanlovchi sicript

```
<!DOCTYPE html>
<html>
<head>
	<title>Aforizmlar</title>
</head>
<body>
<?php
		require_once('config.php');

		$muallif = $matni = '';
		get_aforizm();

		echo '<h1>'.$matni.'</h1>';
		echo '<h3>'.$muallif.'</h3>';
		echo '<br>Yana o`qish uchun F5 - tugmasini bosing.';

	function get_aforizm(){

		global $muallif, $matni;

		// eng katta ID ni aniqlash
		$query 	= 'select max(id_aforizm) from aforizm';
		$result = queryMysql($query);
		$row  = $result->fetch_array(MYSQLI_BOTH);
		$kattasi =  $row[0];

		// tasodiriy rasvishda aforizm ID sini olish
		$id 	= rand(1, $kattasi);

		$query = "select * from aforizm where `id_aforizm` = $id";
		$result = queryMysql($query);

		if ($result->num_rows) {
			$row  = $result->fetch_array(MYSQLI_ASSOC);
			$muallif = $row['muallif'];
			$matni 	= $row['matni'];
		}
		// o'chirigan id tanlangan bo'lsa, qayta tanlAyniiz
		else get_aforizm();
	}
?>
</body>
</html>
```

Aforizmlar ro’yhatini chiqarish

```
<!DOCTYPE html>
<html>
<head>
	<title>Aforizmlar</title>
</head>
<body>
	<form method="post" action="<?php
		echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

	<input type="hidden" name="id" value="<?php echo $id; ?>">
	<table>
		<tr>
			<td>Aforizm:</td>
			<td><textarea name="matni" rows="7"
cols="50"><?php echo $matni ?></textarea>
		</tr>
		<tr>
			<td>Muallif:</td>
			<td><input type="text" name="muallif" value="<?php
echo $muallif; ?>">
		</tr>
	</table>
	<input type="reset"  value="Tozalash" style="width:120px">
	<input type="submit" value="Yubor" style="width:120px">
</form>

</body>
</html>
```

23-Laboratoriya ishi

Mavzu: MySQL ga ma'lumot qoshish

Ishning maqsadi: PHP da ma'lumotlar bazasini bilan HTML da yaratilgan jadvalni bog'lash va tahrirlashni o'rganish;

Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:

Bilishi kerak – HTML da jadvalni hosil qilish ma'lumot qo'shishni bilish;

Ma'lumotlar bazasi va jadvalni hosil qilgandan keyinjadvakja ma'lumot la mi qo'shish mumkin bo'ladi.

Quyida MySQLga ma'lumotlarni qo'shishning ba'zi qoidalari ketiriligan:

1.PHPda SQL so'rovlari qo'shtimoq ichida bo'lishi lozim;

2. SQL so'rovidagi satrli ma'lumotlar qo'shtimoq ichida bo'lishi lozim:

3.SQLso'rovidagi sonli qiymatlar qo'shtirnoqsizyoziladi;

4. NULL qiymati ham qo'shtimoqsizyoziladi.

MySQLga ma'lumot qo'shishning umumiy ko'rinishi

INSERT INTO table_name (column1, column2, column3 . . .)
VALUES (value1, value2, value3…)

Insertga misol

Quyidagi misolda POST metodi orqali muallif va aforizm matni yuboriladi. Qabul qilingan ma'lumotlar *test_input* funksiyasi orqali to'g'irlanadi va bazaga qo'shiladi.

Agar ma'lumotlarni qo'shishda xatolik sodir bo'lsa, bu haqida xabar chiqariladi.

```
if (isset($_POST['muallif']) &&
        isset($_POST['matni']) ) {

        $muallif = test_input($_POST['muallif']);
        $matni        = test_input($_POST['matni']);

        $sorov = 'INSERT INTO  `dars`.`aforizm` (`matni` , `muallif`)';
        $sorov .= "VALUES ('$matni', '$muallif')";
        $result = queryMysql($sorov);

        if (!$result) {
                echo 'Aforizmni qo`shishda xatoli sodir bo`ldi'.
                $connection->error;
        }
    }
```

Aforizm jadvalga yangi ma'lumot qo'shish *insert.php* fayli

```
<!DOCTYPE html>
<html>
<head>
	<title>Aforizmlar</title>
</head>
<body>
	<?php
	require_once('config.php');
	$muallif = $matni = '';

	if (isset($_POST['muallif']) &&
		isset($_POST['matni']) ) {

		$muallif = test_input($_POST['muallif']);
		$matni		= test_input($_POST['matni']);

		$sorov = 'INSERT INTO  `dars`.`aforizm` (`matni` , `muallif`)';
		$sorov .= "VALUES ('$matni', '$muallif')";
		$result = queryMysql($sorov);

		if (!$result) {
			echo 'Aforizmni qo`shishda xatoli sodir bo`ldi'.
			$connection->error;
		}
	}

	$connection->close();
?>

<form method="post" action="<?php
		echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

	<table>
		<tr>
			<td>Aforizm:</td>
			<td><input type="text" name="matni" value="<?php
echo $matni ?>">
		</tr>
		<tr>
			<td>Muallif:</td>
			<td><input type="text" name="muallif" value="<?php
echo $muallif; ?>">
```

```
            </tr>
        </table>
        <input type="reset"  value="Tozalash" style="width:120px">
        <input type="submit" value="Yubor" style="width:120px">
</form>
        <hr>
        <?php
                echo $matni.'<br>';
                echo $muallif.'<br>';
        ?>
</body>
</html>
```

24-Laboratoriya ishi

Mavzu: MySQL ma'lumotlarni o'chirish

Ishning maqsadi: PHP da ma’lumotlar bazasini bilan HTML da yaratilgan jadvaldagi ma’lumotlarni o’chirishni o’rganish;

Ishning natijasida talaba quyidagilarga ega bo’lishi kerak:

Bilishi kerak – Ma’lumotlar bazasidagi ma’lumotlarni o’chirishni bilish;

SQL so'rovlarida DE LE TE buyrug'i orqali ma'lumotlar bazasidan yozuvlar o'chiriladi. SQL so'rovining umumiy ko’rinishi quyidagicha:

id_booktype	type
1	Family
2	Romance
3	Lifestyles
4	History
5	Sport
6	Fun & Humour
7	Crime & thrillers
8	Adventure
9	Science-fiction
10	Horror
11	Other

```
DELETE FROM table_name
WHERE some_column = some_value
```

Malumotla mi o'chirishda diqqatli bo'ling!

Ma'lumotoimi o'chirishda juda extiyot bo’lishingiz lozim bo'ladi. **SQL** so'rovidagi **WHERE** xizmatchi so'zi qaysi yozuv o'chirilishi lozimligini bildiradi. Agar **WHERE** tushirib qoldirifsa, Yoki hamma yozuvlanga tegishli shart yozilsa. jadvaldagi hamma ma'lumotlar o'chib ketadi. Agar ma'lumotlar bazasi nusxasi olib qo'yilniagan bo'lsa, qayta tiklash imkoniyat bo'lmaydi.

Biror narsari o'chirishdan oldin nimani o'chirayotganinggizni aniq bilishingiz lozim bo'ladi. Masalan yuqoridagi jadvaldan "History" ni o'chirmoqchi bo’1sangiz, quyidagicha bo'ladi

```
DELETE FROM `book_type`   WHERE  ` id_booktype` = 4
```

'book_type' jadval nomi

Aforizm jadvalidan ID si korsatilganini o'chirish

```php
<?php
	require_once('config.php');

	if (isset($_GET['id'])) {
		if (is_numeric($_GET['id'])) {

			$id = $_GET['id'];
			$sorov.= "DELETE FROM `aforizm` WHERE
`id_aforizm`=$id";

			$result = queryMysql($sorov);
			if ($result) {
				echo "o`chirildi";
			}
		}
	}

	$connection->close();

?>
```

25-Laboratoriya ishi

Mavzu: PHPda sessiyalar bilan ishlash

Ishning maqsadi: PHP sessiyalar bilan ishlashni o'rganish;
Ishning natijasida talaba quyidagilarga ega bo'lishi kerak:
Bilishi kerak – Ma'lumotlar bazasidagi ma'lumotlarni yangilash bilish;

Sessiyalardan foydalanish uchun dastlab session_start() funksiyasini chaqirish lozim. Bu funksiya har qanday HTML tegdan oldin yozilishi shart. Aks xolda xatolik sodir bo'ladi. Sessiyada cookiedan farqli ravishda fayilar serverda joylashadi.

Sessiyaga qiymat berish $_SESSION global o'zgaruvchisi orqali amalga oshiriladi. Sessiyadan foydalanishga quyidagicha misol keltiramiz 4 ta fayl hosil qilamiz.

1) sessiom, php faylida color va ammal nomli o'zgaruvchi hosil qilamiz.
2) session2.php faylida colorva ammal o'zgaruvchisi qiymatlarini chiqaramiz
3) session3.php faylida o'zgaruvchilar qiymatini ozgartiramiz.
4) session4.php faylida sessiyani o'chiramiz
5) pages.php faylida sahifalarga havolalarga o'rnatamiz

Sessiyadan foydalarnishga misol

1) session 1 .php fayli da color va animal nomli o'zgaruvchi hosil qilamiz.

```
<?php
    // sessiyam boshlash
    session_start();
?>
    <!DOCTYPE html>
    <html>
    <body>
    <?php
        // sessiyaga qiymat berish
        $_SESSION["color"] = "green";
        $_SESSION["ammal"] = "cat";

        echo "Sessiya ma'lumotlari o'rnatildi";
        include "pages.php";
    ?>
</body>
</html>
```

Session2 php faylida color va ammal o’zgaruvchi qiymatlarini chiqaramiz.

```
<?php
	session_start();
?>
<!DOCTYPE html>
<html>
<body>
	<?php
		// Oldingi sahifada o'rnatilgan sessiya ma'lumotlari
		echo "Favorite color is " . $_SESSION["color"] . ".<br>";
		echo "Favorite ammal is " . $_SESSION["ammal"] . ". <br>";
		print_r($_SESSION);
		include "pages.php";
	?>
</body>
</html>
```

Session3 php faylida o’zgaruvchi qiymatlarini o’zgartiramiz.

```
<?php
	session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
	// sessiya ma'lumotlarim o'zgartirish
	$_SESSION["color"] = "yellow";
	$_SESSION["ammal"] = "dog";

	echo "Favorite color is " . $_SESSION["color"] . ".<br>";
	echo "Favorite ammal is " . $_SESSION["ammal"] . ". <br>";

	print_r($_SESSION);
	include "pages.php";
?>
</body>
</html>
```

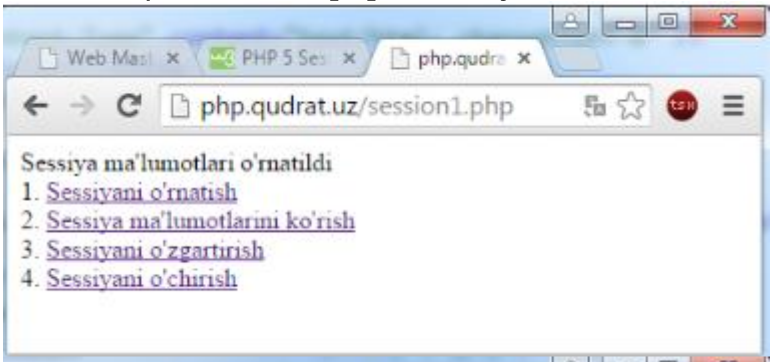
Session4 php faylida sessiyam o’chiramiz.

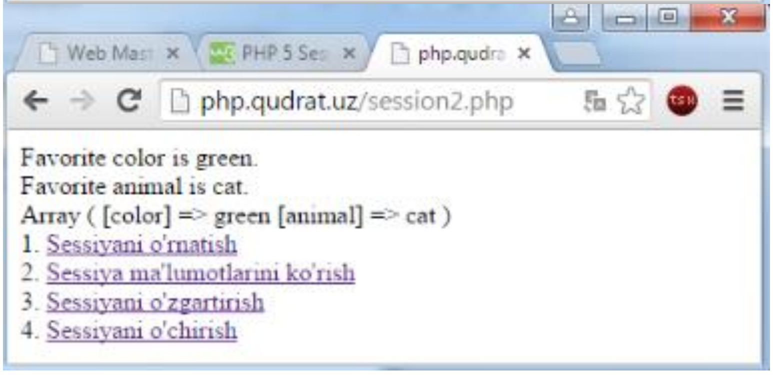
```
<?php
	session_start();
?>
<!DOCTYPE html>
```

```
<html>
<body>
        Sessiya tugatildi.
<?php
        // Sessiyaning barcha o'zgaruvchilarim o'chirish
        session_unset();
        // sessiyam tugatish
        session_destroy();

        include "pages.php";
?>
</body>
</html>
```

Pages.php faylida sessiyam tugatamiz

```
<br>
        1. <a href="session1.php">Sessiyam o'rnatish</a> <br>
        2. <a href="session2.php">Sessiya ma'lumotlarim ko'rish</a> <br>
        3. <a href="session3.php">Sessiyam o'zgartirish</a> <br>
        4. <a href="session4.php">Sessiyam o'chirish</a> <br>
```





Sessiyaga vaqt o'rnatish

Ba'zan sessiyaga ma’lum vaqt oralig'ini o'rnatish kerak bo'ladi. Buning uchun ini_set() funksiyasidan foydalaniladi. Quyidagi misolda sessiya bir kunga o'rnatiladi.

ini_set(‘session.gc_maxlifetime’**,** 60 * 60 * 24);

Agar joriy sessiyaning yashash vaqtini aniqlash kerak bolsa quyidagi funksiyadan foydalanish mumkin.

echo ini_get(‘session.gc_maxlifetime’);

```
<?php
	echo im_get('session.gc_maxlifetime');
	echo '<p> Bir kunga o`rnatish </p>';
	im_set('session.gc_maxlifetime', 60 * 60 * 24);
	echo im_get('session.gc_maxlifetime');

?>
```

26-Laboratoriya ishi

Mavzu: MySQL ma'lumatlarni yangilash (tahrirlash)

Ishning maqsadi: PHP da ma’lumotlar bazasini ma'lumatlarni yangilashni (tahrirlash) o’rganish;

Ishning natijasida talaba quyidagilarga ega bo’lishi kerak:

Bilishi kerak – Ma’lumotlar bazasidagi ma’lumotlarni yangilash bilish;

SQL so'rovlarida UPDATE buyrug'i orqali ma'lumotlar bazasidan yozuvlar o'chiriladi. SQL so'rovining umumiy korinishi quyidagicha:

id_booktype	type
1	Family
2	Romance
3	Lifestyles
4	History
5	Sport
6	Fun & Humour
7	Crime & thrillers
8	Adventure
9	Science-fiction
10	Horror
11	Other

update table_name
set column1=va1ue, column2=valu2,. . .
where some_column=some_value

Ma'lumotlarni yangilashda diqqatli bo’ling!

Ma’lumotlarni yangilashda ham o'chirishdagi kabi juda ehtiyot bo'lishingiz lozim bo'ladi. SQL so'rovidagi WHERE xizmatchi so’zi qaysi yozuv yangilanishi lozimligini bildiradi. Agar WHERE tushirib qoldirilsa, Yoki hamma yozuvlarga tegishli shart yozilsa, jadvaldagi hamma ma'lumotlar yangilanadi. Ya'ni qiymati o'zgarib ketadi. Agar ma'lumotlar bazasi nusxasi olib qo'yilmagan bo'lsa, qayta tiklash imkoniyati bo'lmaydi.

Biror narsani o'zgartirishdan oldin nimani o’zgartirayotganingizni aniq bilishingiz lozim bo'ladi. Masalan yuqoridagi jadvaldan "History" ni "Fable" ga o’zgartirmoqchi bo'lsangiz, quyidagicha bo'ladi.

Aforizm jadvalidagi ma'lumotlarni tahrirlash

<!DOCTYPE html>

```
<html>
<head>
        <title>Aforizmlar</title>
</head>
<body>
<?php
        require_once('config.php');

        $muallif = $matni = '';

        if (isset($_POST['muallif']) &&
                isset($_POST['matni'])) {

                $muallif = test_input($_POST['muallif']);
                $matni = test_input($_POST['matni']);

                $sorov = 'INSERT INTO  `dars`.`aforizm` (
                                        `matni` ,
                                        `muallif`)
                VALUES ("'.$matni.'", "'.$muallif.'");';

                $result = queryMysql($sorov);

                if (!$result) {
                 die('Aforizm qo`shilmadi'.$connection->error);
                }
        }
 ?>

<form method="post" action="<?php
                echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

        <table>
                <tr>
                        <td>Aforizm:</td>
                        <td><input type="text" name="matni" value="<?php
echo $matni ?>">
                </tr>
                <tr>
                        <td>Muallif:</td>
                        <td><input type="text" name="muallif" value="<?php
echo $muallif; ?>">
                </tr>
        </table>
        <input type="reset"  value="Tozalash" style="width:120px">
```

```
		<input type="submit" value="Yubor" style="width:120px">
	</form>
		<hr>
		<?php
			echo $matni.'<br>';
			echo $muallif.'<br>';
		?>
	</body>
	</html>
```

Aforizm jadvalidagi ma'lumotlarni tahrirlash

```
	<?php
		require_once('config.php');

		$id = 0;
		$matni = $muallif = "";

		if (isset($_GET['id'])) {
			if (is_numeric($_GET['id'])) {

				$id = $_GET['id'];
				$sorov = 'select * from aforizm ';
				$sorov.= 'where id_aforizm = '.$id;

				$result = queryMysql($sorov);
				$row = $result->fetch_array(MYSQLI_ASSOC);

				$matni = $row['matni'];
				$muallif = $row['muallif'];
			}
		}

		if (isset($_POST['id']) &&
			isset($_POST['matni']) &&
			isset($_POST['muallif'])) {

			$id	= test_input($_POST['id']);
			$matni		= test_input($_POST['matni']);
			$muallif = test_input($_POST['muallif']);

			$sorov = "UPDATE `dars`.`aforizm` SET `matni` = '$matni',
`muallif`='$muallif' ";
			$sorov.= "where `aforizm`.`id_aforizm` = $id";
```

```
                    $result = queryMysql($sorov);

                    if ($result) {
                            echo "<h1>$matni</h1>";
                            echo "<h3>$muallif</h3>";
                            echo "<p>Yangilandi</p>";
                            echo "<a href=list.php> Ro`xtga o`tish</a>";
                            die();
                     }
                     else {
                            die("Xatolik sodir bo`ldi");
                      }
            }

            $connection->close();
    ?>

    <!DOCTYPE html>
    <html>
    <head>
            <title>Aforizmlar</title>
    </head>
    <body>
            <form method="post" action="<?php
                    echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

            <input type="hidden" name="id" value="<?php echo $id; ?>">
            <table>
                    <tr>
                            <td>Aforizm:</td>
                            <td><textarea name="matni" rows="7"
cols="50"><?php echo $matni ?></textarea>
                    </tr>
                    <tr>
                            <td>Muallif:</td>
                            <td><input type="text" name="muallif" value="<?php
echo $muallif; ?>">
                    </tr>
            </table>
            <input type="reset"  value="Tozalash" style="width:120px">
            <input type="submit" value="Yubor" style="width:120px">
    </form>

    </body>
    </html>
```

GLOSSARIY

№	Terms	Translate
1.	http	Protocol
2.	Client-server	Klient-server
3.	www	World wide web
4.	Tag(code)	Teg(kodlar)
5.	Port	port
6.	address	adres
7.	IP	Internet protocol
8.	Web page	Web sahifa
9.	Animation	animasiya
10.	Apache	Apache Software Foundation
11.	Web server	Web server
12.	Web site	Web sayt
13.	browser	brouser
14.	CGI	Common gateway interface
15.	Cookie	Cookie fayl
16.	CSS	Cascading style sheet
17.	Database	database
18.	Dynamic IP	Dynamic Internet protocol
19.	E-mail	E-mail
20.	Flash	flash
21.	frame	frame
22.	Form	forma
23.	FrontPage	Front page
24.	FTP	File transport protocol
25.	GIF	Gif fayl
26.	gateway	shlyuz
27.	Host	host
28.	hosting	hosting
29.	HTML	Html text markup language
30.	HTML Editor	HTML editor
31.	Html form	HTML formasi
32.	Html page	HTML sahifasi
33.	Html tags	HTML teglari
34.	Hyperlink	giperishorat
35.	font	shrift
36.	Internet Explorer	Internet explorer

37.	Intranet	intranet
38.	IP address	IP address
39.	Java	JAVA
40.	JavaScript	JavaScript
41.	JScript	JScript
42.	JSP	JSP
43.	Navigate	navigator
44.	Perl	Perl tili
45.	php	Personal home page tili
46.	THTTPD	oddiy, kichik hajmdagi, tez va xavfsiz veb server.
47.	CMS	Content Management System – Kontentlarni Boshqarish Tizimi
48.	WordPress	foydalanish uchun juda oson, qulay va eng asosiysi 100% bepul bo’lgan dastur.
49.	Joomla	CMS asosan katta saytlar va portallar uchun
50.	Blog	bu internet kundalik hisoblanadi, bu kundalikni yurituvchi shaxsbloger deyiladi.
51.	header	sahifa nomi uchun teg
52.	footer	aytning eng pastki qismi(podval) uchun zarur bo’lgan teg.
53.	SSD	bu qattiq diskning(nakopitel) bir turi bo‘lib, kontroller va mikrosxemalardan tashkil topgan bo‘lad
54.	ICT	axborot-kommunikasiya texnologiyalari
55.	website optimization	veb-saytni optimallash
56.	webpage title	veb-sahifa sarlavhasi
57.	webmaster	Web master
58.	web-hosting	Web-hostin
59.	web-interface	veb-interfeys
60.	web-document	Web-hujjat
61.	web-colors	Web-ranlari
62.	webcam	Web kamera
63.	web template	veb-shablon
64.	gateway page	geytvey
65.	web-design	veb-dizayn

Xulosa

Shunday qilib, biz PHP dasturlash tili haqida ma’lumotga ega bo’ldik, u qanday dunyoga kelgan va tarqalgan, uni qanday va qayerda foydalanilishini o’rgandik, dasturiy vositani o’rnatdik hamda uni ishlashi uchun barcha sozlashlarni bajardik va *php*-dastur nimalardan tashkil topishini angladik, asosiy sintaksislarini ko’rib chiqdik hamda bir qancha amaliy masalalarni hal etdik. Hozir kunda webga yo’naltirilgan dasturlash tillari ichida PHP ning muhim ahamiyat kasb etishini amalda ko’rib chiqdik, ya’ni PHP dasturlash tilisiz ma’lumotlar bazasini to’laqonli hosil qilish va amalga oshirishning iloji yo’q. Qolaversa murakkab bo’lgan jarayonlarni tashkillashtirish qidiruv tizimini amalga oshirish juda qulayligini amalda ko’rdik. MySQL ma’lumotlar bazasini ishlashni o’rgandik va PHP bilan bog’ladik. PHP va MySQL web saytlar tayyorlashda muhim vosita hisoblanishini amalda ko’rib chiqdik.

Foydalaniladigan adabiyotlar ro'yxati

Asosiy adabiyotlar

1. Karimov I.A. Yuksak ma'naviyat - yengilmas kuch. - T.: Ma'naviyat, 2008.
2. M.Aripov, M.Fayziyeva, C.*floTToeB*.Web dasturlash. Ukuv qo'llanma.T.; "Faylasuflar jamiyati". 2013.
3. R.Xamdamov va boshkalar. Ta'limda axborot texnologiyalari. Uslubiy qo'llanma.Toshkent. Uzbekistan milliy ensiklopediyasi" 2010
4. Robin Nixon. Learning PHP, MySQL&JavaScript width JQuery, CSS & HTML5, 4th Edition. O'Reilly Media. United States of America. 2015.
5. Лебедов С В..Wеб-дизайн..Учеб.пос.Москва, ЗАО «Издателский дом Алянс пресс», 2004
6. Jamsil K. Lalani, S., S. Weekly Programming Web professionals Trans, from English. A.I.Panasyuk - Mn. LLC "Potpourri", 1997. - 632 p.: III.
7. Malchuk EV. HTML and CSS. self-teacher

Qo'shimcha adabiyotlar:

1. M.Mamarajabov, S.Tursunov. Kompyuter grafikasi va Web dizayn. Darslik.T.:"Cho'lpon"\ 2013

2.U. Yuldashev, M.Mamarajabov, S.Tursunov. Pedagogik Web dizayn. Ukuv qo'llanma.T: "Voris"\ 2013

3.VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.

6.R. Darnell. "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.

7.Dmitriev M. JavaScript. Quick start. - St. Petersburg.: BHV St. Pereburg, 2002.

Internet saytlar:

5. http://www.intuit.ru
6. http://etuit.uz/
7. www.tuit.uz
8. www.ziyonet.uz