

Xayatov Xurshidjon Usmanovich, Murodov Shaxriddin Qahramonovich

xml tilini php dashturlash tilida ishlatish

(o'quv-uslubiy qo'llanma)

BUXORO 2018 yil

Ushbu qo`llanmada XML tili va uning imkoniyatlar, PHP tili orqali XML dan foydalanish haqidagi boshlang`ich tushunchalar keltirilgan. Web dastrulash bilan mustaqil o`rganuvchilar uchun mo`ljallangan. Bundan tashqari XML bilan ishlashni o`rganishni xohlovchilarga mo`ljallangan bo`lib, ular mazkur qo`llanmadan foydalanishi maqsadga muvofiqdir.

Mualifflar:

X. U. Xayatov – Buxoro davlat universiteti
“Axborot texnologiyalari” kafedrası” katta
o`qituvchisi

Sh. Q. Murodov - Buxoro davlat universiteti
Kasb ta`limi (Informatika va axborot
texnologiyalari) ta`lim yo`nalishi 4-kurs
talabasi

Taqrizchilar:

O.I. Jalolov – Buxoro davlat universiteti
“Axborot texnologiyalari” kafedrası” dotsenti

Z. Sh. Jo`rayev – Buxoro davlat universiteti
“Axborot texnologiyalari” kafedrası” katta
o`qituvchisi

MUNDARIJA

Kirish	4
I – bob. Web serverlar va texnologiyalar	5
1.1 Web serverlar haqida.....	5
1.2 Web texnologiyalar haqida	11
1.3 Php tili va undan foydalanish	12
II – bob. XML va undan foydalanish.....	28
2.1 XML tili va uning sintaksisi.....	28
2.2 XML da XPath bilan ishlash	38
2.3 XML da XQuery bilan ishlash	39
III – bob. PHP da XML bilan ishlash.....	40
3.1 PHP orqali XML ma`lumotlari bilan ishlash	40
3.2 PHP orqali XML ma`lumotlarini o`zgartirish, qo`shish, o`chirish.....	65
3.3 PHP orqali XML ma`lumotlariga XPath, XQuery orqali ma`lumotlarini tanlash.....	68
Foydalanilga adabiyotlar	71

KIRISH

Jahonda axborot texnologiyalari bo'yicha juda rivojlanib bormoqda. Endilikda dasturlar tuzish bir yoki ikki shaxs tomonidan emas balki guruh yoki kompaniyalar orqali tuzilmoqda. Katta imkoniyatli dasturlar tuzishda, berilganlar bazasi, XML fayllarida ma'lumotlar saqlanmoqda. Bu ma'lumotlar bilan ishlaydigan dasturlar uch xil texnologiya asosida yaratilmoqda. Desktop (yakka tartibda) turdagi dasturlar. Mijoz-server texnologiyasi asosida dasturlar. Web texnologiyalar asosida tuzilgan dasturlar.

Hozirgi paytda eng qulay texnologiya web texnologiya bo'lib, unda barcha dasturiy ta'minotlarni serverda joylashgan bo'ladi. Mijozning kompyuterida esa atigi web brouzer bo'lishi yetarli. Qolgan web-ilova dasturlarining hammasi serverga joylashgan bo'ladi.

Ushbu o'quv – uslubiy qo'llanma “Xml tilini php dashturlash tilida ishlatish” deb nomlangan. Bunda XML tili, uning sintaksisi va undan foydalanish haqida ma'lumotlar keltirilgan. XML ma'lumotlar ustida ishlash va PHP orqali undan foydalanish bo'yicha ma'lumotlar ham keltirilgan. Bu ma'lumotlardan Web dasturlash faniga mos keladigan mavzulariga doir bo'lgan mavzularida foydalanish va mustaqil shug'ullanuvchilar uchun mo'ljallangan.

I bob. Web serverlar va texnologiyalar

1.1 Web serverlar haqida

Bugungi kunda internet jadal suratda rivojlanib bormoqda. Shu bilan bir qatorda internetga oid xar-xil termin, atamalar ham ortib bormoqda. Masalan sayt, server, web-sahifa, portal,.. va hk. Bu terminlarni o'rganish, hozirgi hunda juda muhim hisoblanadi. Biror inson bilan internet haqida suhbatlashsangiz, bu terminlarni bilmasangiz, boshqa tildagi inson bilan suhbatlashayotganday bo'lasiz va uni tushunmaysiz ham. Bu rejamda shu terminlardan biri hisoblangan web-server haqida to'xtalib o'taman.

Biror bir internet saytga kirish uchun, brouzerga manzil yozsangiz, brouzer shu sayt joylashgan kompyuterga, kompyuterdagi ma'lumotlarni ko'rsatish uchun so'rov yuboradi. Kompyuter oddiy termin bo'lgani uchun bu so'rovni tushunmaydi. Shuning uchun, ya'ni bu so'rovni tushunish uchun, kompyuterga dastur o'rnatiladi. Mana shu dastur web dastur deyiladi. Bu kompyuter esa, web-serverga aylanadi. Bugungi kunda esa buni web dastur deyilmasdan web-server deb atalmoqda.

Demak, web-server — foydalanuvchidan kelayotgan so'rovlarni qabul qilib, tegishli javobni jo'natadigan dasturdir. Foydalanuvchi sifatida web brouzer(http, https protokoli orqali), mobil telefon(wap pratokoli orqali), ba'zi dasturlar (antiviruslar) bo'lishi mumkin. Web-server orqali butun sayt boshqariladi. Ya'ni sayt katologlariga ruxsatlarni beradi, xavfsilik jihatdan IP manzillarini blokka oladi. Foydalanuvchilarni ro'yxatdan o'tkazish va ko'plab ishlarni amalga oshiradi. Web-serverni siz uyingizda ham yaratishingiz mumkin. Buning uchun bitta kompyuter ajratilib, unga biror web dastur o'rnatasiz. Bu dsturni yaxshilab sozlashingiz kerak bo'ladi. So'ng o'z saytingizni shu serverga joylashtirasiz va biror domen sotib olasiz. Bundan kelib chiqadiki, saytingiz ishlashi uchun siz ajratgan kompyuter doimo yoqilgan holatda bo'lishi kerak.

Web-serverga o'rnatiladigan web dasturlarning turlari anchagina. Hozirda eng zamonaviy va eng ommabop dastur bu – albatta “Apache”dir. Apache Software Foundation korxonasi tamonidan ishlab chiqarilgan. Quyida ba'zi web-serverlar nomini keltirish mumkin:

- Apache - Apache Software Foundation
- IIS – Microsoft mahsuloti hisoblanadi. Server operatsion tizimlariga o'rnatilgan(Windows server 2003, Windows server 2008,..).
- Nginx – ochiq web-server, katta hajmdagi web-saytlar uchun.
- Lighttpd – ochiq web-server.
- Google web server – Google tomonidan ishlab chiqilgan.
- Cherokee – web interfeys orqali boshqariladi.
- Rootage – java tilida yozilgan web-server.

- THTTPD – oddiy, kichik hajmdagi, xavfsiz web-server.

Apache Software Foundation (ASF) – Apache dasturiy ta’minoti loyihalarni rivojlantirishga ko’maklashuv tashkilot-jamg’armasi va boshqalar bilan birga, HTTP web-serveri ham kiradi. ASF 1999-yil Delaverdagi, AQSH Apache Group guruhidan tashkil qilingan.

Ba’zi ma’lumotlarga qaraganda, 2011-yil yanvar oyiga kelib, tarmoqda 273301445 ta sayt faoliyat yuritgan. Apache web-serveri ancha vaqtdan beri yetakchi o’rinlarni egallab kelmoqda. Keyingi vaqtlarda web-serverlar bozorida o’rni 59,13 foizini tashkil qiladi. ASF hamjamiyati jahonning turli joylarida yashovchi va turli Open Source loyihalarida ishtirok qiluvchi loyihachi tashabbuskorlardan tashkil qilingan.

Apache o’ziga xos tomoni – bu kodlarni birgalikda ishlab chiqish va ochiq, pragmatik – Apache Software License litsenziyasining mavjudligidir.

Apache HTTP-server - bepul tarqatiluvchi web-server hisoblanib, Shimoliy Amerikadagi Apaches qabilasi nomiga qo’yilgan. Bundan tashqari, ingliz tilidan “A tasodifiy” server deb tarjima qilinadi, odatda rus foydalanuvchilar orasida apache buzib A pathy deb yuritiladi. Apache dasturiy ta’minoti Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS operatsion tizimlari qo‘llab-quvvatlaydigan kross-platforma hisoblanadi. Apachening asosiy afzalliklari ishonchliligi va konfiguratsiya moslashuvchanligidadir. Bu xato xabarlarni, ma’lumotlar bazasi foydalanuvchilarga aniqlashda, o‘zgartirish uchun ishlatiladigan ma’lumotlarni taqdim qilish uchun tashqi modulni ulash imkonini beradi.

Server 1995 yilda yozilgan deb hisoblanadi va uning kulgili nomi «a tasodifiy», u keyin mashhur World Wide Web Server NCSA httpd 1.3 xatolari topilgan. Keyinchalik, versiya 2.x, server qayta yozilgan va hozir kodi NCSA o‘z ichiga oladi. Ayni paytda, rivojlanish versiyasiyasi 2.2 yaratilmoqda va 1.3 va 2.0 versiyalarida xavfsizlikdagi xatolar qaytadan yozib chiqilmoqda. Ayni paytda, 2.4 - 2.4.3 (21 avgust 2012) eng oxirgi versiyasi ommaga taqdim etilgan.

1996 yilning Aprel oyidan hozirgi zamon Internetida eng ommabop HTTP-server hisoblanadi. Netcraftning Apache foydalanish haqida quyidagi statistik ma’lumotlarni ko‘rish mumkin:

2007 yil avgust oyida, u barcha web-serverlarida 51% ishlagan;

2009 yil may oyida 46% da ishlagan;

2011 yilning yanvar oyida 59%, ya'ni - 160 milliondan ortiq saytlarda ishlagan.

Apache yadrosi o'zining asosiy funksiyalari, konfiguratsiya fayllarini tahlil qilish, HTTP protokoli va tizimda modulni yuklash imkoniyatiga ega. Apache yadrosi uchinchi tomon dasturchilarni jalb qilmasdan butunlay Apache Software Foundation tomonidan ishlab chiqilgan.

Apache yadrosi to'liq C dasturlash tilida yozilgan.

Tizim konfiguratsiya

Apache tizim konfiguratsiya matn konfiguratsiya fayllariga asoslangan. Bu an'anaviy konfiguratsiyaning uch darajasi bor:

- Server konfiguratsiya (httpd.conf).
- Virtual host konfiguratsiya (httpd.conf, 2.2 versiyadan boshlab qo'shimcha httpd-vhosts.conf)
- Papka konfiguratsiya darajasi (.htaccess).

Bu dasturda papka bloklariga asoslangan o'zining til konfiguratsiya fayllar bor. Deyarli barcha yadro parametrlarni konfiguratsiya fayllar o'zgartirish mumkin, bu orqali MPM(Multiprotsessor modeli)ni boshqarish mumkin. Modelning katta qismi o'z parametrlarni ega. Model operatsion tizimda ish davomida ba'zi konfiguratsiya fayllaridan foydalanadi. (masalan / etc / passwd va / etc / hosts). Bundan tashqari, parametrlarni buyruq qatori orqali o'rnatishingiz mumkin.

Multiprotsessorli model

Apache web serveri uchun juda ko'p multiprotsessorli modellar mavjud.

Tizim moduli

Apache HTTP server modulini qo'llab quvvatlaydi. Turli vazifalarni bajaruvchi 500 dan ortiq modullar bor. Ulardan bazilari Apache Software

Foundation jamoasi tamonidan yaratilgan, lekin asosiy qismi Open Source jamoasi tamonidan ishlab chiqilgan.

Apache virtual xosting ichki mexanizmi hisoblanadi. U butun boshli ko'p saytlarga bitta IP manzil orqali xizmat ko'rsaradi va ularning har birini alohida ko'rsatadi.

Har bir virtual xost uchun o'z yadro va modul konfiguratsiya ko'rsatishimiz butun sayt yoki fayllarga kirishlarni cheklash mumkin. Ayrim MPM lar, masalan: Apache-ITK httpda har bir virtual xost uchun alohida uid va guid identifikatorlarini ishlashini talab qiladi.

Boshqa dasturlar va dasturlash tillari bilan integratsiyasi

Apache turli xil dasturlash tillar va tizimni kuchaytirishni qo'llab-quvvatlash uchun ko'pgina qo'shimcha modullar bor. Bular:

PHP (mod_php).

Python (mod_python mod_msgi).

Ruby (apache_ruby).

Perl (mod_Perl).

ASP (apache-asp).

TCL (chega)

Bundan tashqari Apache C, C++, Lua, Java deyarli barcha dasturlash tillarini ishlatish imkonini beradi. U CGI va FastC mexanizmlarini qo'llab-quvvatlaydi.

IIS – Internet Information Service.

MS Internet Information Server ommaviylikiga ko'ra ikkinchi hisoblanadi. NetCraft kompaniyasi tomonidan to'plangan ma'lumotlarga asoslanib IIS Web-serveri jami tugunlarning 35% ga xizmat qiladi. IIS 7.0 versiyasining chiqarilishi 2001 yil Windows NT 4.0 tarkibida chiqarilgan 1-versiyasining 10 yilligiga to'g'ri keldi. 4 yildan so'ng IIS5.0 internetda eng keng tarqalgan server bo'ldi, biroq bir necha oy o'tishi bilan birga mashhur cherv viruslari-CodeRed va Nimdanning qurboniga aylandi. Windows Server 2003 tarkibidagi 6-versiyasi barcha urinishlar xavfsizlikni ishonchlilikni va samaradorlikni oshirishga qaratilgan jiddiy qayta ishlangan versiyasi hisoblanadi. Shundan boshlab IIS 6.0 o'zini yuqori darajada himoyalagan Web-serverlar ekanligini yuqori ishonchlilik xavfsizlikni ta'minlovchi ekanligini ko'rsatib kelmoqda. U ishlab chiqarishdan buyon uzoqlashgan rejimda foydalanish mumkin bo'lmagan xavfsizlik bo'yicha xatolikni to'g'rilovchi bittayangilash chiqarildi. IIS 7.0 IIS6 asosiy kodidan tezlik, ishonchlilik va xavfsizlikni meros qilib olib uni yuqori darajada kengayadigan va boshqaradigan Web-server platformasiga aylantirishi, zamonaviy Web-dasturlarni etarlicha quvvat bilan bajaruvchi vositaga aylantirishi lozim edi. Natijada IIS ning eng katta miqdordagi arxitektura yangiliklarini jamlagan istiqbolli MS versiyasi yaratildi. IIS ning asosida to'liq modulli versiyasi yotib, o'zida kompakt versiyaga birlashtirilgan, dastur topologiyasida kerakli optimallashtirilgan 40 dan ortiq komponentalarini jamlagan. Bu komponentalar yangi kengayuvchanlik qatlamida yaratilgan bo'lib, dastur tuzuvchilarga serverning istalgan funksiyasini mashina kodida yoki MS.NerFrameWork yordamida kengaytirish imkonini beradi. IIS 7.0 bajarish, boshqaruv va ishchi komponentalari kengaytirishni taklif qilish bilan birga konkret himollarga mos ravishda kompleks yechimlarni yaratishni yengillashtiradi. IIS platformasida IIS 6. ga nisbatan serverni boshqarish va foydalanish bilan bog'liq ko'plab muammolarni hal etadi. U tugunlarni boshqarishda huquqlarni to'liq ta'minlovchi va yakunda Web-dasturlarni xcopy dan foydalanib yechishni reallashtiradigan prinsipial jihanda yangi sozlash tizimiga ega.

Boshqaruv va diagnostik komponentalaridan iborat yangi API interfeyslari yoyish ,adminstrasiyalash va serverning nosozliklarini bartaraf etish protseduralarini oldingilaridan sodda va qulay ko‘rinishga olib keldi.

IIS Web-serveri ,Web-dasturlar yaratishning bir nech xil turli texnologiyalarni quvvatlaydi.

-ASP.Net Windows tizimlari uchun bugungi kunda Web-dasturlar va Web-xizmatlar yaratishning asosiy vositasi hisoblanadi. ASP.Net ni quvvatlash IIS 6.0 dan kiritilgan. Undan oldingi versiyalari uchun alohida NetFrameWork ni ko‘chirib olib o‘rnatiladi.

-ASP-senariy asosida dinamik Web-sahifalar yaratishning eskirgan texnologiyasi.IIS 3.0 dan boshlab Web-server tarkibiga kirutilgan.

-CGI-dinamik Web-sahifalar yaratishning standart texnologiyasi. ISAPI-Windows tizimlari uchun IIS ning barcha komponentalarida to‘liq foydalanish imkonini beruvchi kuchli texnologiya. CGI vaISAPI yordamida IIS Web-serverga Web-dasturlarni quvvatlovchi vositalar ,masalan ,PHPva Perl ulanishi mumkin. IIS 7.0 dagi qiziqarli va istiqbolli yangiliklardan biri IIS Media pack hisoblanadi. Ikkita qo‘shimcha bepul modul Web-serverni zamonaviy mediaeshittirish asbobiga aylanishga imkon berdi.MS ning yangi texnologiyalari media ma’lumotlar raqamli oqimini optimallashtirishga va savodli boshqarishga imkon beradi. Server quyidagi formatli ma’lumotlarni eshittirishga imkon beradi:

ASF, AVI, FLV, MLV, MOV, MP3, MP4, FM,RHVM,WMA,WMV.

Mijoz-server dasturlarini quyidagilarni hisobga olish lozim:

-yaratilayotgan axborot tizimi qanday foydalanuvchilarga mo‘ljallangan;

-xavfsizlikka qanday talab qo‘yiladi.

1.2 Web texnologiyalar haqida

ASP (Active Server Pages) ASP- ma'lumot bazalarini tashkil etish va ular bilan ishlash vazifalarini bajarishda juda moslashuvchan, qulay vositadir. ASP vositalari server tomonida ishlaydi. HTML kod va ssenariylar kabi fayllarni qayta ishlaydi. ASP texnologiyasi VBScript, Java va JavaScript tillarini qo'llab-quvvatlaydi. ASP-kod joylashgan web-sahifa fayllari kengaytmasi .asp bo'ladi. ASP texnologiyasi Windows NT va Microsoft IIS web-servariga mo'ljallangan hisoblanib, imkoniyatlari va samaradorligi yuqori bo'lganligi bois ko'pgina kompaniyalar o'z vositalariga ASP ni qo'llab-quvvatlash imkoniyatlarini kiritmoqda. ASP vositalarini ishlab chiqarish bo'yicha yirik kompaniya Chillsoft Lider hisoblanadi. Ko'pgina HTML-muharrirlar, masalan Adobe GoLive ham ASPni qo'llab-quvvatlaydi.

ASP texnologiyasi qulayliklari quyidagilarni o'z ichiga oladi:

HTML-hujjatni dinamik generatsiyalaydi, formalarini qo'llab-quvvatlaydi, ma'lumotlar bazasiga ruhsatni tashkil etadi va u bilan ishlay oladi. ASP – dasturlash tili ham ilova ham emas, u interaktiv web-sahifa hosil qilish texnologiyasi.

PHP – bu serverda qayta ishlanuvchi ssenariylar tilidir. ASP kabi PHP kodlar ham bevosita HTML-hujjatning tarkibiga qo'shiladi/Ushbu tilning nomi Personal Home Page Tools so'zining qisqartmasidan olingan. PHP da C va Perl tillarida uchragan bir qator muammolar hal etilgan, bundan tashqari, PHP ma'lumotlar bazasi bilan ishlash uchun ham juda qulay vositadir. Umuman olganda PHP ochiq tizimli til hisoblanadi. Ularni dasturchilar modernizatsiyalashtira oladi.

JSP (JavaServerPage) texnologiyasi o'zining funksional imkoniyatlariga ko'ra ASP ga o'xshashdir. Asosiy farqi shundaki, bunda VBScript va JavaScript bilan birga Java tili ham qo'llana oladi. Shunga qaramay JSP Javadan oldinroq qo'llanilgan va ushbu texnologiya mukammal web-ilovalar yartish uchun yetarli imkoniyatga ega.

SSI (Server Side Include) vositasi dastlab HTML-faylni dastlab serverda qayta ishlaydi va undan so'ng uni klientga uzatadi. Dastlabki qayta ishlash vaqtida hujjatga dinamik generatsiya qilingan ma'lumotlar qo'shiladi. Masalan: joriy vaqt haqidagi ma'lumot. Umuman olganda SSI texnologiyasi HTML-faylning tarkibiga qo'shimcha qo'llanmalar qo'shishga mo'ljallangan, HTMLning qismiga hisoblanadi.

1.3 Php tili va undan foydalanish

PHP haqida qisqacha ma'lumot.

PHP (Hypertext Preprocessor) — eng ko'p tarqalgan dasturlash tillaridan biri bo'lib, Web — dasturlar yaratishda ishlatiladi. Hozirda PHP juda ko'p hosting-provayderlar tomonidan foydalaniladi, bu esa uni harqanday internet- projeklarni (oddiygina saytdan tortib, to yirik portallargacha) yaratishda ishlatiladigan, sal kam, eng asosiy tilga aylantirdi. PHP da juda ko'p miqdorda alohida skriptlar, shuningdek, forum, kontent tizimi boshqaruvi kabi tugallangan projektlar yozilgan.

PHP tarixi. 1994 yili php tilinig yaratuvchisi Rasmus Lerdorf o'zinig saytiga mehmonlar kirishini hisoblash uchun Perl/ CGI(skriptlar to'plami)da maxsus qobiq yozib, amalda qo'lladi va uni «Personal Home Page»(PHP nomi shundan kelib chiqqan) deya nomladi. Ammo Perl — sezilarli darajadagi sekin ishlovchi interpretator, tez orada uning ishlash unumdorligi yetmay qola boshladi, shu sababli, Rasmus «C» dasturlash tilida yangi interpretator yaratdi va uni PHP/FI(Personal Home Page / Form Interpreter) deb nomladi. Yangi interpretator Perlga juda o'xshardi, masalan, o'zgaruvchini nomi oldidan dollar(\$) belgisini qo'yish. O'shandayoq zamonaviy PHP ning poydevori qo'yila boshlandi (HTML ga kiritish kabilar). 1997-yilda ikkinchi qismi yaratildi — PHP/FI 2.0. U taxminan 50 ming serverga o'rnatilgan edi. O'ylab qarasa, bu juda katta raqam, lekin bu internet-serverlarning umumiy sonining atigi 1% ni tashkil etadi. 1998-yilda PHP 3.0 ishlab chiqildi. Aynan shu versiyadan boshlab PHP rekursiv akronim (dasturning qisqartirilgan nomi uning kengaytmasida ham ishlatilishi) sifatida qo'llana boshlandi, ya'ni PHP bu — PHP: Hypertext Preprocessor. Uchinchi

versiya ikkinchisiga nisbatan ancha omadliroq chiqdi, u barcha internet-serverlarning 10% iga o'rnatilindi, bu esa «ancha- muncha»! PHP 3.0 ning yadrosining sekin ishlashi qisman tanqidga uchradi, chunki berilgan masalalar ketma- ketligi yetarlicha tezlikda bajarilmayotgan edi. Shu sababli PHP 4 ni yadroni qayta ishlashdan boshlashdi. To'rtinchi versiya (PHP 4) 2000- yilning May oyida ishga tushdi. PHP ning ushbu versiyasi 2007-yilning ohirigacha yangilanib kelinadi(ОБНОВЛЕНИЕ). 2008-yilning Avgust oyidan boshlab bu ish to'xtatildi. PHP ning beshinchi versiyasi 2004-yil Iyul oyida chiqdi. Zend yadrosi qayta ishlab chiqilgan edi, bu esa interpretatorni unumdorligini oshirdi. Endi PHP o'z yo'nalishiga ega bo'lgan, to'laqonli dasturlash tili hisoblanar edi, uning obyekt modeli esa Java'ning modeli bilan juda ko'p jihatdan o'xshash. PHP 1.0 — 1995-yil, 8- iyun. Rasmiy nomi “ Personal Home Page Tools (PHP Tools)”. Bu “PHP” qisqartmani birinchi marta ishlatilinishi. PHP 2.0 — 1996-yil, 16- aprel. Veb sahifalarni “Tez va oson ishlaydigan qurol” deb e'lon qilingan mahsulot. PHP 3.0 — 1998-yil, 6- iyun ,Bir kishi ishlab chiqaruvchi loyihadan ko'p ishlab chiqaruvchilar loyihasiga aylandi. Endi Gutmans va Ziv Suraski kodni boshqatdan yozib chiqishdi. PHP 4.0.0 — 2000-yil, 22-may. Yangi yadro “ Zend Engine” asosida tuzilgan PHP 4.1.0 — 2001-yil, 10- dekabr. Til konstruktsiyasiga superglobal imkoniyatlar qoshilgan (\$_GET, \$_POST, \$_SESSION, etc.) PHP 4.2.0 — 2002-yil, 22- aprel. Xavfsizlikni oshirish maqsadida tizimning sozlash “register_globals” parametri ochirilgan PHP 4.3.0 — 2002-yil, 27- dekabr. “CGI” (Common Gateway Interface)ga qo'shimcha “CLI” (command line interpreter) texnologiyasi kiritilgan. PHP 4.4.0 — 2005-yil, 11- iyul. phpize va php-config skriptlari uchun qo'llanma varaqlari qo'shilgan. PHP 5.0.0 — 2004-yil, 13- iyul. Zend Engine II – yangi obyekt modeli kiritilgan. PHP 5.1.0 — 2005-yil, 24- noyabr. PHP Engine ishlash unumdorligi yaxshilangan. PHP 5.2.0 — 2006-yil, 2- noyabr. Kengaytiradigan filtr modullarining php tizimida ishlatish imkoniyatlari ochiq holati o'rnatilgan (default). PHP 5.3.3 — 2010-yil, 22- iyul. Axborot xavfsizligi kuchaytirilgan.

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 C, 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 C va Perl dasturlash tiliga o'xshash. Biroq keltirilgan dastur C va Perl dasturlash tilidagi dasturdan ancha katta farq qiladi. Bu erda 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 C 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 mijoz kompyuteridagina bajariladi. Mijoz 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 protsessori qayta ishlaydigan qilib sozlab olishingiz ham mumkin. YA'ni mijozlar oddiy HTML-faylni qabul qildimi yoki skript natijasini ko'rdimi buni bila olmaydi. Agarda skript mijoz kompyuterida qayta ishlansa (masalan, JavaScript tilidagi dastur), u holda mijoz skript kodidan iborat HTMLsahifani 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.

PHP dasturlash tili faylni qayta ishlayotganda u oddiy matnni PHP kod interpretatsiya 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 interpretatsiyalaydi. Ya'na 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

Instruktsiyani 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.

Biz ko'p hollarda misollarda `<?php ?>`

variant o'rniga qisqartirilgan `<? ?>` teglardan foydalanamiz.

Instruktsiyalarni ajratilishi.

PHP dasturlash tilidagi dastur(ixtiyoriy dasturlash tilidagi) - bu buyruqlar (instruksiyalar) to'plamidir. Dasturni qayta ishlash uchun bir buyruqni boshqa buyruqdan farqini bilish kerak. Buning uchun maxsus simvollar - ajratgichlardan foydalaniladi. PHP dasturlash tilida instruktsiyalarni xuddi C yoki Perl dasturlash tillari kabi ajratiladi, ya'ni har bir ifoda nuqtali vergul (";") bilan tugaydi. «?>» yopiladigan teg ham instruktsiyani tugashini anglatadi, 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 anglatadi, /* va */ belgilar esa mos ravishda ko'p satrli kommentariylarning boshlanish va tugashini anglatadi.

Misol: PHP dasturlash tilida kommentariyning qo'llanilishi

```
<?php
```

```
echo "Meni ismim Alisher";
```

```
// Bu bir satrli kommentariy
```

```
// C++ dasturlash tili stilidagi
```

```
echo "Meni familiyam Murodov";
```

```
/* Bu ko'p satrli kommentariy. Bu erga bir qancha satr yozish mumkin.
```

Dastur bajarilish jarayonida bu erdagi barcha yozuvlar (kommentariylangan), o'qilmaydi.

```
*/ echo "Men PHP dasturlash tilini o'rganyapman";
```

```
# Bu kommentariy
```

```
# Unix qobig'idagi kommentariy. ?>
```

O'zgaruvchilar, o'zgarmlar va operatorlar

Har bir dasturlash tilida muhim elementlardan biri bu o'zgaruvchilar, o'zgarmlar va ular qo'llaniladigan operatorlardir. 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.

Misol.

```
<?php
```

```
$a = "Hello, world";
```

```
echo ($a);
```

```
// natija sifatida "Helli, world" texti chiqadi.
```

?>

O'zgarmlar.

Skript bajarilish jarayonida o'zgarmlaydigan qiymatli kattaliklarni saqlash uchun o'zgarmlarlardan foydalaniladi. Bunday kattaliklar matematik o'zgarmlar, parollar, fayllarning yo'llari va boshqalar bo'lishi mumkin. O'zgarmlarlarning 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'zgarmlar oldida dollar belgisi qyilmaydi hamda uni oddiy qiymat o'zlashtirish kabi qarash mumkin emas. O'zgarmlar qanday aniqlanadi? Buning uchun maxsus `define()` funksiyasi mavjud, uning sintaksisi quyidagichadir:`define("O'zgarmlar nomi", "O'zgarmlar qiymati", [registrga sezgirligi kichik])`

O'zgarmlar nomi registrga segirligi katta. Har bir o'zgarmlarlarda uni o'zgartirish mumkin, ya'ni registrga sezgirligi kichik argumentni qiymati sifatida `True` qiymati ko'rsatiladi. O'zgarmlar nomi har doim katta registr bilan yozishga kelishib olingan. O'zgarmlarni qiymatini bilish uchun uni nomini ko'rsatish kerak.

O'zgaruvchidan farqi o'zgarmlar nomi oldiga `$` belgi qo'yilmaydi. Bundan tashqari o'zgarmlarni qiymatini bilish uchun konstanta nomi bilan parametr sifatida `constant()` funksiyasidan foydalanish mumkin.

Misol. PHP dasturlash tilida o'zgarmlar.

```
<?php
// o'zgarmlarni aniqlaymiz
PASSWORD
define("PASSWORD","qwerty");
// registrlanmagan PI o'zgarmlarni qiymatini aniqlaymiz
3.14 define("PI","3.14", True);
// PASSWORD o'zgarmlar qiymatini olamiz, ya'ni
qwerty echo (PASSWORD); // bu ham qwerty ni
```

```

chiqaradi echo constant("PASSWORD"); echo
(password);
/* password ni chiqaradi va biz registrlangan o'zgarma PASSWORD ni
kutgandik.*/ echo pi;
// 3.14 ni chiqaradi, chunki o'zgarma PI registrlanmagan va
aniqlangan. ?>

```

Amallar.

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

1.3.1-jadval. PHPda arifmetik amallar.

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

1.3.2-jadval. O'zlashtirish amallari.

O'zlashtirish amallari.			
Belgila nishi	Nomlan ishi	Izox	Misol
=	O'zlashtirish	<i>Operator</i> dan o'ng tomonda turgan	$\$a = (\$b = 4) + 5;$ ($\$a$ 9 ga teng, $\$b$ 4

		o'zgaruvchilar ustida bajarilgan amallardan hosil bo'lgan natija qiymati o'zlashtiriladi.	ga teng)
+=		Qisqartirish. O'zgaruvchiga son qo'shiladi va keyin natija o'zlashtiriladi.	$\$a += 5;$ ($\$a = \$a + 5$ ifodaga ekvivalent;)
.=		O'zlashtirish va konkatenatsiya amallari kombinatsiyasini qisqartirilgan shakli(dastavval satrlar qo'shiladi, so'ngra hosil bo'lgan satr o'zgaruvchiga o'zlashadi).	$\$b = \text{"Hammaga "};$ $\$b .= \text{"salom"};$ ($\$b = \$b . \text{"salom"}$ ifodaga ekvivalent;) Natijasi: $\$b = \text{"Hammaga salom"}$

1.3.3-jadval. Mantiqiy amallar.

Mantiqiy amallar.			
Belgilanishi	Nomlanishi	Izox	Misol
and	VA	$\$a$ va $\$b$ rost (True)	$\$a$ and
&&	VA		$\$b$ $\$a$ &&

or	YOKI	$\$a$ yoki $\$b$ o'zgaruvchilardan hech bo'lmaganda bittasi rost bo'lsa (ikkalasi ham rost bo'lishish mumkin).	$\$b$ $\$a$ or $\$b$
	YOKI		$\$a$ $\$b$ $\$a$
xor	Inversiya YOKI	O'zgaruvchilardan bittasi rost bo'lsa. Agarda ikkalasi ham rost bo'lsa inversiyalanadi.	xor $\$b$
!	Inversiya (NOT)	Agarda $\$a=True$, u holda $!\$a=False$ va aks holda teskarisi bo'ladi.	! $\$a$

1.3.4-jadval. Taqqoslash amali.

Taqqoslash amallari			
Belgilanishi	Nomlanishi	Izox	Misol
==	Tenglik	O'zgaruvchilar	$\$a == \b $\$a ===$
===	Ekvivalentlik	qiymatlari teng	$\$b$ $\$a != \b
!=	Tengsizlik	O'zgaruvchilar	
	Tengsizlik	qiymatlari va	
<>	Noekvivalentlik	tiplari teng	
!==	Kichik	O'zgaruvchilar	
<		qiymatlari teng emas	
		O'zgaruvchilar ekvivalent emas	$\$a < \b $\$a !==$ $\$b$

			$\$a < \b
>	Katta		$\$a > \b
<=	Kichik yoki teng		$\$a < = \b
>=	Katta yoki teng		$\$a > = \b

1.3.5-jadval. Inkrement va dekrement amallari.

Inkrement va dekrement amallari.		
Belgilanishi	Nomlanishi	Izox
++\$a	<i>Preinkrement</i>	\$a qiymati birga oshiriladi va \$a qiymati qaytariladi
\$a++	<i>Postinkrement</i>	\$a qiymati qaytariladi va so'ngra \$a qiymati birga oshiriladi
--\$a	<i>Predekrement</i>	\$a qiymati birga kamaytiriladi va \$a qiymati qaytariladi
\$a--	<i>Postdekrement</i>	\$a qiymati qaytariladi va so'ngra \$a qiymati birga kamaytiriladi

Ma'lumotlar tiplari

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

To'rttasi skalyar *tiplar*: . *boolean* (mantiqiy); . *integer* (butun); . *float* (nuqtasi siljiydigan); . *string* (satrli).

Ikkitasi aralish *tiplar*: . *array* (massiv); . *object* (obyekt).

PHP dasturlash tilida o'zgaruvchilar tiplari oshkora e'lon qilinmaydi. Ko'pincha o'zgaruvchi qo'llanilgan kontekstdan, 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 tiplarini* birma-bir ko'rib chiqamiz.

Boolean tipi(Bul yoki mantiqiy tip).

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

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

Misol. Mantiqiy tip. `<?php
$test =True; ?>`

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

Misol. Mantiqiy tiplarning qo'llanilishi.

```
<?php
// '==' amal tenglikka tekshiradi mantiqiy qiymatni qaytaradi
if ($know == False) { // agar $know qiymat false bo'lsa
echo "PHP dasturlash tilini o'rgan!";
}
if (!$know) { // xuddi yuqoridagidek $know qiymati false bo'ladi
echo " PHP dasturlash tilini o'rgan!";
}
/* == amal $action o'zgaruvchi qiymati bilan "PHP dasturlash tilini o'rganish!"
satrni
ustma-ust tushishini tekshiradi. Agar ustma-ust tushsa true qiymatni qaytaradi,
boshqa
```

holda false ni qaytaradi. Agar true ni qaytarsa figurali qavs ichidagi amallar bajariladi. */

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

Integer (butun) tipi.

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

Agar siz sakkizlik sanoq sistemasidan foydalanayotgan bo'lsangiz, oldindan 0 (no1) 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 sonni o'lchami platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati ikki milliard (bu ishorali 32 bitli qiymat) atrofida bo'ladi. Ishorasiz *butun sonni PHP* dasturlash tili qo'llab quvvatlamaydi.

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

PHP dasturlash tilida butun sonlarni bo'lish amali mavjud emas. $1/2$ ifoda qiymati *qo'zg'aluvchan vergulli son* 0.5 ga teng. Siz natijangizni butun tipga standart qoida asosida yoki `round()` funksiyasidan foydalangan taqdirda o'zgartirishingiz mumkin. O'zgaruvchini aniq bir tipga o'zgartirish uchun uning oldiga qavs ichida kerakli tipni yozish kerak bo'ladi. Masalan, $\$a=0.5$ o'zgaruvchini butun tipga o'zgartirish uchun `(integer)(0.5)` yoki `(integer) $a` ko'rinishda yoki qisqartirilgan `(int)(0.5)` ko'rinishda yozish kerak bo'ladi. Bunday oshkora yangi tipga o'tish imkoniyati barcha *ma'lumotlar tiplari* uchun o'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 kontekstdan mustaqil ravishda o'zi bu ishlarni bajaradi. **Float (Qo'zg'aluvchan vergulli son) tipi.**

Qo'zg'aluvchan vergulli sonlar (ular ikki karra aniqlik yoki haqiqiy sonlardir) quyidagi sintaksislar yordamida aniqlanishi mumkin: `<?php $a = 1.234; $b = 1.2e3; $c = 7E-10; ?>`

Qo'zg'aluvchan vergulli sonni o'lchami ham platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati $\sim 1.8e308$ aniqlik bilan 14 xonali raqam atrofida bo'ladi.

Resource (resurslar) tipi.

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

Null tipi.

Maxsus *NULL* qiymati *o'zgaruvchini* qiymatga ega emasligi haqida ogohlantiradi. *O'zgaruvchi NULL* qiymat qabul qiladi, agarda: `.` unga *h zgarmas NULL* (`$var = NULL`) o'zlashtirilgan bo'lsa; `.` unga ech qanday qiymat berilmagan bo'lsa; `.` u *unset()* funksiya yordamida tozalangan bo'lsa.

NULL tipli faqat bitta qiymati mavjud - registrga sezgirligi kichik *NULL* kalit so'zidir.

PHP tilining boshqaruv operatorlar (if, switch, for, while,?).

if operatori shart operatori hisoblanadi. Uning sintaksisi quyidagicha:

if(**shart**) ifoda1 *else* ifoda2.

```
<?php
```

```
$a=1;
```

```
$b=2;
```

```
If ($a>$b) {echo($a)
```

```
Else echo($b);} ?>
```

Switch tanlash shart operatori. Umumiy sintaksisi

```
switch(expression) // qiymat
```

```
{
```

```
  case value1: // 1-ifoda
```

```
    statements; // operator bloki
```

```
  break;
```

```
  case value2: // 2-ifoda
```

```
    statements; opertor bloki
```

```
  break;
```

```
  default:
```

```
    statements; operator bloki.
```

Misol:

```
<?php
```

```
switch($number)
```

```
{
```

```
  case 1:
```

```
    echo ("one ");
```

```
  case 2: case 3:
```

```
    echo ("free");
```

```
  case 4: case 5:
```

```
    echo ("five");
```

```
  case 6: case 7:
```

```
    echo ("seven");
```

```
  case 8: case 9:
```

```
    echo ("nine");
```

```
  break;
```

```
  default:
```

```
    echo ("This isn't number or number is > 9 or < 1");
```

```
}
```

```
?>
```

for takrorlash operatori. Bu operatorning berilish sintaksisi quyidagicha:

```
for(expression1;expression2;expression3)
```

```
{
```

```
  statements;
```

```
}
```

Misol:

```
<?php
$var = 5;
$i = 0;
for ($i = 0; $i <= $var; $i++)
{
    echo($i);
    echo('<br>');
} ?>
```

while sikl operatori. Bu operator ishlatilishi C++ dagi kabi ishlatiladi.

Shart chin qiymat qabul qilganda berilgan amal bajariladi. While ga quyidagi misolni keltirib o'tamiz:

```
<?php
$var = 5;
$i = 0;
while(++$i <= $var)
{
    echo($i); echo('<br>');
}
?>
```

PHP da funksiya va sinflar. PHPda funksiyalarning berilishi quyidagicha:

Function <funksiya nomi> (parametr) {operator}

Misol: oddiy funksiya.

```
<? function summ( ) {
    $a=5;
    $b=6;
    $c=$a+$b;
    return $c;
    echo (summ( ));
}
?>
```

Endi parametrli funksiyaga misol:

```
<? function summ($a,$b) {
    $c=$a+$b;
    echo ($c);
}
summ(5,3);
?>
```

II bob. XML va undan foydalanish

2.1 XML tili va uning sintaksisi

XML (eXtensible Markup Language) – Kengaytirilgan formatlash tili hisoblanadi. XML HTMLga o'xshash formatlashtiruvchi tildir. HTMLdan farqli tarafi shundaki bunda dasturchi o'zi yangi teg yaratib undan foydalanib borish imkoniyatini beradi. Bu til ma'lumotlarni saqlash va tahrirlash uchun juda ham qulay hisoblanadi. XML tilining rivojlanishi 1996-yildan boshlangan. Bu ish bilan W3C (World Wide Web Consortium) tashkiloti shug'ullangan. Bu tashkilotning rasmiy web-manzili www.w3.org. Aynan XML uchun berilgan sahifasi esa quyidagidan iborat: www.w3.org/XML/.

XML sintaksisi qoidalari oson va ularni o'rganish juda ham qulay. Demak, XML hujjat sintaksisi qoidalari bilan tanishishni boshlaymiz.

<?...?> xml belgisi va versiyasi ifodalanishi shart. U quyidagicha beriladi:

```
<? xml version="1.0">
```

Bundan tashqari, xml da kodirovkani ham berish maqsadga muvofiq bo'ladi. Odatda kodirovka hujjatda **UTF-8** xalqaro standart bo'yicha yoziladi.

```
<? xml version="1.0" encoding=UTF-8>
```

XMLda kommentarilar esa quyidagicha ishlatiladi:

```
<!-- bu izoh -->
```

1-misol:

```
<? xml version="1.0" encoding="UTF-8"?>
```

```
<note>
```

```
<to>Tove</to>
```

```
<from>Jani</from>
```

```
<heading>Reminder</heading>
```

```
<body>Don't forget me this weekend!</body>
```

```
</note>
```

Bu misolning natijasini tekshirib ko'ramiz:



2.1.1-rasm.

2-misol:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<breakfast_menu>
```

```
<food>
```

```
<name>Belgian Waffles</name>
```

```
<price>$5.95</price>
```

```
<description>
```

Two of our famous Belgian Waffles with plenty of real maple syrup

```
</description>
```

```
<calories>650</calories>
```

```
</food>
```

```
<food>
```

```
<name>Strawberry Belgian Waffles</name>
```

```
<price>$7.95</price>
```

```
<description>
```

Light Belgian waffles covered with strawberries and whipped cream

```
</description>
```

```
<calories>900</calories>
```

```
</food>
```

```
<food>
  <name>Berry-Berry Belgian Waffles</name>
  <price>$8.95</price>
  <description>
    Belgian waffles covered with assorted fresh berries and whipped cream
  </description>
  <calories>900</calories>
```

```
</food>
```

```
<food>
  <name>French Toast</name>
  <price>$4.50</price>
  <description>
    Thick slices made from our homemade sourdough bread
  </description>
  <calories>600</calories>
```

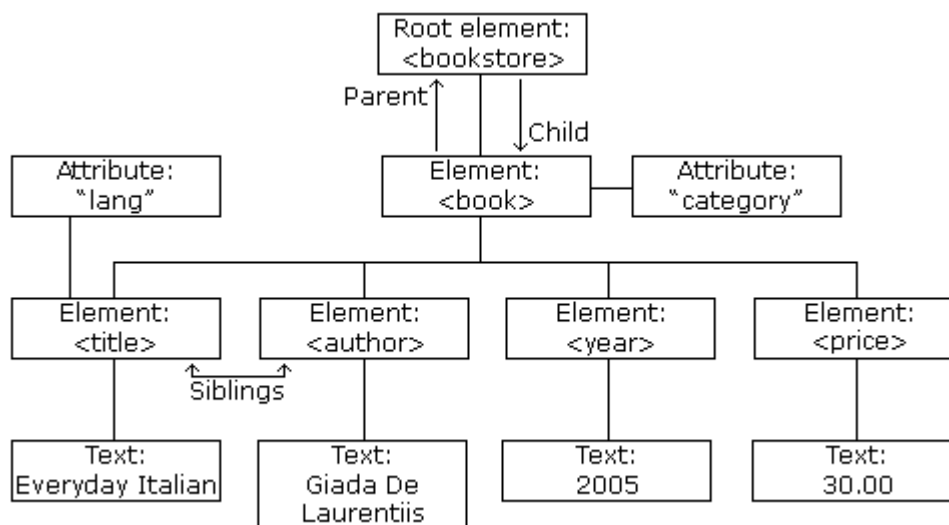
```
</food>
```

```
<food>
  <name>Homestyle Breakfast</name>
  <price>$6.95</price>
  <description>
    Two eggs, bacon or sausage, toast, and our ever-popular hash browns
  </description>
  <calories>950</calories>
```

```
</food>
```

```
</breakfast_menu>
```

XML daraxt tuzilishi:



2.1.2-rasm.

Yuqoridagi rasm asosida tuzilgan XML formatdagi kitob haqida ma'lumot beradigan misolni ko'rib chiqamiz.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<bookstore>
```

```
  <book category="cooking">
```

```
    <title lang="en">Everyday Italian</title>
```

```
    <author>Giada De Laurentiis</author>
```

```
    <year>2005</year>
```

```
    <price>30.00</price>
```

```
  </book>
```

```
  <book category="children">
```

```
    <title lang="en">Harry Potter</title>
```

```
    <author>J K. Rowling</author>
```

```
    <year>2005</year>
```

```
    <price>29.99</price>
```

```
  </book>
```

```
  <book category="web">
```

```
    <title lang="en">Learning XML</title>
```

```
    <author>Erik T. Ray</author>
```

```
    <year>2003</year>
```

```
    <price>39.95</price>
```

```
  </book>
```

```
</bookstore>
```

Demak, XML daraxtning tuzulishi:

XML - hujjatlari element daraxtlari sifatida shakllanadi.

XML - daraxti ildiz elementi va filialni ildizdan pastki elementlarga boshlaydi.

Barcha elementlar ichki qismlari (bolalari) bo'lishi mumkin:

```
<root>
```

```
<child>
```

```
<subchild>.....</subchild>
```

```
</child>
```

```
</root>
```

XML teglari katta / kichik harflarga sezgir. <Letter> yorlig'i <letter> tegidan farq qiladi.

XML o'z-o'zini tushuntirib beruvchi sintaksisini juda ko'p ishlatadi.

Keyingi qator hujjatning asosiy elementi:

```
<bookstore>
```

Keyingi satrda <book> elementi boshlanadi:

```
<book category = "cooking">
```

<Book> elementlarida 4 ta kichik element mavjud: <title>, <author>, <year>, <price>.

```
<title lang = "en"> Kunlik italyancha </ title>
```

```
<Muallif> Giada De Laurentis </ author>
```

```
<year> 2005 </ year>
```

```
<price> 30.00 </ price>
```

Keyingi satrning book elementini tugatadi:

```
</ book>
```

Ushbu misoldan XML hujjatiga kitob do'konidagi kitoblar haqida ma'lumot kiritilganini ko'rishimiz mumkin.

XML sintaksisi qoidalari

XML hujjatlari ildiz elementiga ega bo'lishi kerak.

XML hujjatlari barcha boshqa elementlarning asosiy elementlaridan biri bo'lishi kerak:

```
<root>
```

```
<bola>
```

```
<subchild> ..... </ subchild>
```


</ child>

</ root>

Ushbu misolda <note> - ildiz element:

<? xml version = "1.0" kodlash = "UTF-8"?>

<eslatma>

<to> Tove </ to> ga bosing

<dan> Jani </ from>

<sarlavhasi> eslatma </ heading>

<body> Meni bu hafta oxiri unutmang! </ body>

</ note>

XMLda yopilish yorlig'ini bekor qilish noqonuniy hisoblanadi. Barcha elementlar yopilish yorlig'iga ega bo'lishi kerak:

<p> Bu xat. </ p>

Kirish va yopish teglari bir xil holatda yozilishi kerak:

<message> Bu to'g'ri </ message>

"Ochish va yopish teglari" ko'pincha "Boshlash va tugatish belgilar" deb nomlanadi.

Eslatma: XML prologida yopilish yorlig'i yo'q! Bu xato emas. Prolog XML hujjatining bir qismi emas.

XML elementlari to'g'ri joylashtirilgan bo'lishi kerak. XML formatida barcha elementlar bir-biriga to'g'ri joylashtirilgan bo'lishi kerak:

 <i> Ushbu matn qalin va kursivdir </ i> </ b>

Yuqoridagi misolda "to'g'ri kiritilgan" oddiygina <i> elementi ichida elementi ochilganligi tufayli element ichida yopilishi kerakligini bildiradi.

XML elementlari HTML-da bo'lgani kabi nom / qiymat jufti bilan atributlarga ega bo'lishi mumkin. Faqat, XML formatida atribut qiymatlari har doim keltirilishi kerak:

<note date="12/11/2018">

<to>Tove</to>

<from>Jani</from>

</note>

Bu yerda date- atribut sifatida qo'llanilgan.

XMLda bir nechta oq bo'shliqlarni qisqartirmaydi (HTMLda esa bir necha oq bo'shliqlarni bitta bo'shliqqa qisqartiradi):

XML:	Hello	Tove
HTML:	Hello Tove	

XML xususiyatlari(attributlar).

XML elementlari xuddi HTML kabi atributlarga ega bo'lishi mumkin. Xususiyatlar muayyan elementga tegishli ma'lumotlarni o'z ichiga olishi uchun yaratiladi. Albatta, XML da atribut qiymatlari ko'rsatilgan bo'lishi kerak. Bunda ular bitta yoki juft tirnoqlar bilan yoziladi. Bir kishining ma'lumoti uchun <person> elementi shunday yozilishi mumkin:

```
<person gender="female">
```

yoki shunga o'xshash:

```
<person gender='female'>
```

Agar atribut qiymati o'zi juft tirnoqlardan iborat bo'lsa, ushbu misol kabi bitta tirnoqlardan foydalanishingiz mumkin:

```
<gangster name='George "Shotgun" Ziegler'>
```

bu ham xato hisoblanmaydi.

XML elementlari va atributlardan iborat quyidagi misollarni ko'rib chiqamiz:

```
<person gender="female">
```

```
<firstname>Anna</firstname>
```

```
<lastname>Smith</lastname>
```

```
</person>
```

```
<person>
```

```
<gender>female</gender>
```

```
<firstname>Anna</firstname>
```

```
<lastname>Smith</lastname>
```

```
</person>
```

Birinchisida female atribut sifatida kelga, ikkinchisida esa element bo'lib kelgan.

Atributlardan foydalanishda e'tiborga olinadigan ayrim narsalarni keltirib o'tamiz:

- Xususiyatlar bir nechta qiymatlarni o'z ichiga olmaydi.
- atributlar daraxt tuzilmalarini o'z ichiga olmaydi.
- Xususiyatlar kengaytirilishi mumkin emas.

Metadata uchun XML xossalari.

Ba'zan identifikatsiya ma'lumotnomalari elementlarga beriladi. Ushbu identifikatorlar XML elementlarini identifikatsiyalash HTML-da bir xil tarzda identifikatsiyalash uchun ishlatilishi mumkin. Ushbu misolda shu narsa keltirib o'tilgan:

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

XML DOM

DOM nima?

DOM(Document Object Model) - hujjatlarga kirish va ularni qayta ishlash uchun standartni belgilaydi.

HTML DOM HTML-hujjatga kirish va ularni qayta ishlashning standart usulini belgilaydi. U daraxt tuzilishi sifatida HTML hujjatini ifodalaydi. DOMni tushunish HTML yoki XML bilan ishlaydigan har bir kishi uchun muhim hisoblanadi. Barcha HTML elementlarga HTML DOM orqali kirish mumkin. Ushbu misol id = "demo" bilan HTML elementining qiymatini o'zgartiradi:

```
<h1 id="demo">This is a Heading</h1>
```

```
<button type="button"
onclick="document.getElementById('demo').innerHTML='Hello World!'">
```

Click Me!

```
</button>
```

HTML fayli esa quyidagicha:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1 id="demo">This is a Heading</h1>
```

```
<button type="button"
```

```
onclick="document.getElementById('demo').innerHTML = 'Hello
```

World!'">Click Me!

```
</button>.
```

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

Agar sizga JavaScriptni o'rganishda HTML DOM ko'proq o'rganishini xohlasangiz https://www.w3schools.com/js/js_htmlDOM.asp manzil orqali olishingiz mumkin.

Demak, barcha xml elementlariga XML DOM orqali kirish mumkin:

Books.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<bookstore>
```

```
<book category="cooking">
```

```
<title lang="en">Everyday Italian</title>
```

```
<author>Giada De Laurentiis</author>
```

```
<year>2005</year>
```

```
<price>30.00</price>
```

```
</book>
```

```
<book category="children">
```

```
<title lang="en">Harry Potter</title>
```

```
<author>J K. Rowling</author>
```

```
<year>2005</year>
```

```
<price>29.99</price>
```

```
</book>
```

```
</bookstore>
```

Ushbu kod XML hujjatidagi birinchi <title> elementining matn qiymatini chiqaradi:

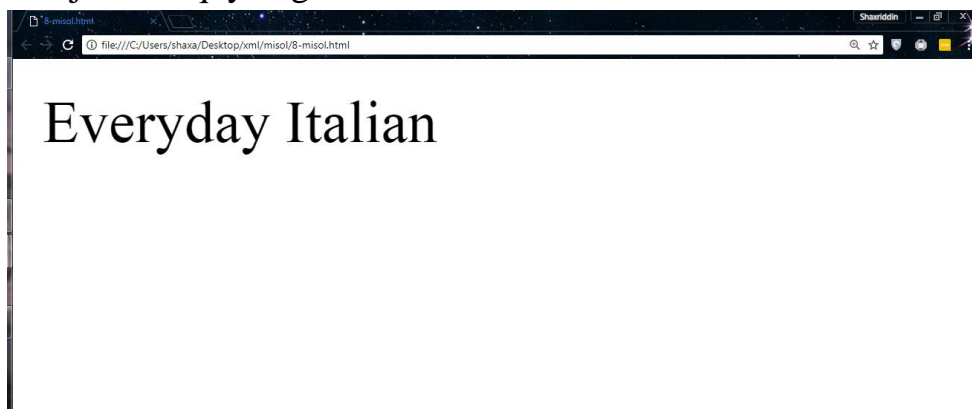
```
txt=xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
```

XML DOM XML elementlarini olish, o'zgartirish, qo'shish va o'chirish uchun ham ishlatiladi.

Ushbu misol, matnli satrlarni XML DOM ob'ektiga yuklaydi va undan JavaScript dan foydalanib ma'lumotlarni chiqaradi:

```
<html>
<body>
<p id="demo"></p>
<script>
var text, parser, xmlDoc;
text = "<bookstore><book>" +
"<title>Everyday Italian</title>" +
"<author>Giada De Laurentiis</author>" +
"<year>2005</year>" +
"</book></bookstore>";
parser = new DOMParser( );
xmlDoc = parser.parseFromString(text,"text/xml");
document.getElementById("demo").innerHTML =
xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
</script>
</body>
</html>
```

Natijasi esa quyidagicha:

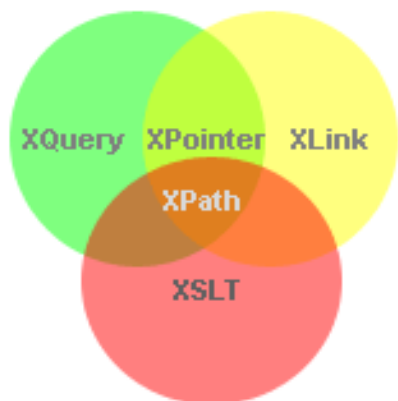


2.1.3-rasm.

XML DOM haqida mustaqil yanada ko'proq o'rganmoqchi bo'lsangiz https://www.w3schools.com/xml/dom_intro.asp manzilidan foydalanishingizni tavsiya qilamiz.

2.2 XML da XPath bilan ishlash

XPath (**Path**–“yo‘l, so‘qmoq”) XSLT standartidagi asosiy element hisoblanadi. XPath XML hujjatidagi elementlar va atributlar bo'yicha harakat qilish uchun ishlatilishi ham mumkin.



2.2.1-rasm.

- XPath XML hujjat qismlarini aniqlash uchun sintaksis hisoblanadi.
- XPath, XML hujjatlarini yurish uchun yo'l ifodalari ishlatadi.
- XPath standart funksiyalar kutubxonasini o'z ichiga oladi.
XPath XSLT va XQuery-da muhim hisoblanadi.

XPath iboralari JavaScript, Java, XML sxemasi, PHP, Python, C va C++ da ishlatilishi mumkin. Bundan tashqari XPath XSLTda ishlatiladi. XPath XSLT standartidagi asosiy element hisoblanadi. Siz XPathni mukammal o'rgana olsangiz XLS dan keng qamrovda foydalana olasiz.

Quyidagi jadvalda ba'zi XPath ifodalari va ifodalarning natijalarini keltirilingan.

2.2.1-jadval. XPath ifodalari.

XPath ifodasi	Natija
/bookstore/book[1]	Kitob egasi elementining bolasi bo'lgan birinchi kitob elementini tanlaydi
/bookstore/book[last()]	Kitob egasi elementining bolasi bo'lgan oxirgi kitob elementini tanlaydi
/bookstore/book[last()-1]	Kitob egasi elementining bolasi bo'lgan oxirgi kitob elementini tanlaydi
/bookstore/book[position()<3]	Kitob egasining bolalari bo'lgan birinchi ikki kitob elementini tanlaydi
//title[@lang]	“Lang” nomli atributga ega bo'lgan barcha nom unsurlarini tanlaydi
//title[@lang='en']	"En" qiymatiga ega "lang" xususiyati bo'lgan barcha nom unsurlarini tanlaydi
/bookstore/book[price>35.00]	35.00dan kattaroq narxdagi element maqomiga ega kitob do'konining barcha elementlarini tanlaydi

/bookstore/book[price>35.00]/title	35.00dan kattagina narx elementi bo'lgan kitob sotuvchisi elementlarining kitob elementlarining barcha nom qismlarini tanlaydi
------------------------------------	--

XPath haqida ko'proq ma'lumotlarni [XPath Tutorial](#) dan olishingiz mumkin.

2.3 XML da XQuery bilan ishlash

XQuery XML ma'lumotlarini so'rov qilish uchun mo'ljallangan. XML uchun XQuery ma'lumotlar bazalari uchun SQL kabi hisoblanadi.

XQuery, XPath so'zlarida yaratilgan.

XQuery barcha asosiy ma'lumotlar bazalari tomonidan qo'llab-quvvatlanadi.

XQueryga misol:

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title.
```

XQuery XML hujjatidan elementlar va atributlarni topish va chiqarish uchun tildir. XQuery XML hujjatidan elementlar va atributlarni topish va chiqarish uchun tildir.

Bu yerda XQuery qanday natija bo'lishi mumkinligi haqidagi misol keltirilgan:

"Cd_catalog.xml da joylashgan CD to'plamidan \$ 10 dan kam bo'lgan CD-yozuvlarni tanlang".

XQuery va XPath

XQuery 1.0 va XPath 2.0 bir xil ma'lumot modelini ulashadi va bir xil funktsiyalarni va operatorlarni qo'llab-quvvatlaydi. Agar siz allaqachon XPathni o'rgangan bo'lsangiz, XQueryni tushunishida hech qanday muammo bo'lmaydi.

- XQuery quyidagilarni bajarishi mumkin:
- Veb-xizmatidan foydalanish uchun ma'lumotlarni chiqarish;
- Xulosa hisobotlarini tuzish;
- XML ma'lumotlarini XHTMLga o'tkazish;
- Tegishli ma'lumot uchun veb-hujjatlarni qidirish;

III bob. PHP da XML bilan ishlash

3.1 PHP orqali XML ma'lumotlari bilan ishlash

PHP XML Parsers

XML hujjatini o'qish va yangilash, yaratish va boshqarish uchun sizga XML-parserga kerak bo'ladi.

PHP-da XML-parsersning ikkita asosiy turi mavjud:

- Daraxt asosidagi parser;
- Voqeaga asoslangan parser;

Daraxt asosidagi parser. Daraxt asosidagi parserlar butun hujjatni ushlab turadilar. Barcha hujjatni tahlil qiladi va daraxt elementlariga (DOM) kirish imkonini beradi.

Ushbu turdagi ayristirıcı, kichik XML hujjatlari uchun yanada yaxshi tanlovdir, lekin katta XML hujjati uchun emas.

Daraxt asosidagi parserlarga misol:

- SimpleXML;
- DOM;

Voqeaga asoslangan parserlar. Voqeaga asoslangan parserlar butun hujjatni xotirada saqlamaydilar, buning o'rniga ular bir vaqtning o'zida bitta tugunni o'qiydi va real vaqt bilan o'zaro muloqot qilish imkonini beradi. Keyingi tugunga o'tib ketganingizdan so'ng, eski nusxa ko'chiriladi. Ushbu turdagi parsers, katta XML hujjatlari uchun juda mos keladi. U tezroq ajralib chiqadi va kamroq xotirani egallaydi.

Voqeani asoslangan parserlarga misol:

XMLReader;
XML Expat Parser;

PHP SimpleXML Parser

SimpleXML bizni osongina manipulyatsiya qilish va XML ma'lumotlarini olish imkonini beruvchi PHP kengaytmasi hisoblanadi. SimpleXML daraxt asosidagi parserdir. SimpleXML, XML hujjatining tuzilishini yoki tartibini bilsangiz, elementning nomi, atributlari va matn tarkibini olishning osongina imkon beradi. Bundan tashqari, SimpleXML XML hujjatini bir qator ma'lumotlar majmuasiga o'xshash ma'lumotlar strukturasi aylantiradi. DOM yoki Expat parseriga nisbatan SimpleXML elementdan matnli ma'lumotlarni o'qish uchun kodning kamroq satrini oladi.

PHP SimpleXML - Stringdan o'qish.

PHP `simplexml_load_string()` funksiyasi XML ma'lumotlarini magistraldan o'qish uchun ishlatiladi.

```
$myXMLData =
"<?xml version='1.0' encoding='UTF-8'?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>";
```

Quyidagi misol `simplexml_load_string()` funksiyasidan XML ma'lumotlarini magistraldan o'qish uchun qanday ishlatilishini ko'rsatadi:

```
<?php
$myXMLData =
"<?xml version='1.0' encoding='UTF-8'?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>";
```

```
$xml=simplexml_load_string($myXMLData) or die("Error: Cannot create
object");
```

```
print_r($xml);
?>;
```

Yuqoridagi kodning chiqishi quyidagicha bo'ladi:

```
SimpleXMLElement Object ( [to] => Tove [from] => Jani [heading] =>
Reminder [body] => Don't forget me this weekend! );
```

Misol:

```
<?php
libxml_use_internal_errors(true);
$myXMLData =
"<?xml version='1.0' encoding='UTF-8'?>
<document>
<user>John Doe</wronguser>
<email>john@example.com</wrongemail>
</document>";
```

```
$xml = simplexml_load_string($myXMLData);
```

```

if ($xml === false) {
    echo "Failed loading XML: ";
    foreach(libxml_get_errors() as $error) {
        echo "<br>", $error->message;
    }
} else {
    print_r($xml);
}
?>;

```

Yuqoridagi kodning chiqishi quyidagicha bo'ladi:

Failed loading XML:

Opening and ending tag mismatch: user line 3 and wronguser

Opening and ending tag mismatch: email line 4 and wrongemail.

PHP SimpleXML - Fayldan o'qish. PHPda simplexml_load_file() funksiyasi XML ma'lumotlarini fayldan o'qish uchun ishlatiladi. Endi "Note.xml" deb nomlangan XML fayli mavjudligini ko'raylik.

```

<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

```

Quyidagi misol simplexml_load_file() funksiyasidan qanday qilib fayldan XML ma'lumotlarini o'qish uchun ishlatilishini ko'rsatadi:

```

<?php
$xml=simplexml_load_file("note.xml") or die("Error: Cannot create object");
print_r($xml);
?>.

```

Yuqoridagi kodning chiqishi quyidagicha bo'ladi:

```

SimpleXMLElement Object ( [to] => Tove [from] => Jani [heading] =>
Reminder [body] => Don't forget me this weekend! )

```

PHP SimpleXML - Get Node/Attribute Values.

"Note.xml" faylidan tugun qiymatlarini olish:

```

<?php
$xml=simplexml_load_file("note.xml") or die("Error: Cannot create object");
echo $xml->to . "<br>";
echo $xml->from . "<br>";
echo $xml->heading . "<br>";
echo $xml->body;

```

?>;

Yuqoridagi kodning chiqishi quyidagicha bo'ladi:

“Tove

Jani

Reminder

Don't forget me this weekend!”

Boshqa bir XML faylida shu "Books.xml" deb nomlangan XML fayli mavjudligini ko'rib chiqaylik.

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en-us">XQuery Kick Start</title>
    <author>James McGovern</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="WEB">
    <title lang="en-us">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

PHP SimpleXML - ma'lum elementlarning tugun qiymatlarini qabul qilish. Quyidagi misol, "books.xml" faylidagi birinchi va ikkinchi <book> elementlaridagi <title> elementining tugun qiymatini oladi:

```
<?php
```

```

$xml=simplexml_load_file("books.xml") or die("Error: Cannot create
object");
echo $xml->book[0]->title . "<br>";
echo $xml->book[1]->title;
?>

```

Natijasi esa quyidagicha:

Everyday Italian

Harry Potter.

Quyidagi misol, "books.xml" fayliidagi barcha <book> elementlari orqali pastadir va <title>, <author>, <year> va <price> elements ning tugun qiymatlarini oladi:

```

<?php
$xml=simplexml_load_file("books.xml") or die("Error: Cannot create
object");
foreach($xml->children( ) as $books) {
    echo $books->title . ", ";
    echo $books->author . ", ";
    echo $books->year . ", ";
    echo $books->price . "<br>";
}
?>

```

Natijasi esa:

Everyday Italian, Giada De Laurentiis, 2005, 30.00

Harry Potter, J K. Rowling, 2005, 29.99

XQuery Kick Start, James McGovern, 2003, 49.99

Learning XML, Erik T. Ray, 2003, 39.95.

PHP SimpleXML - Atribut qiymati olish.

Quyidagi misolda birinchi <book> elementining «kategoriya» xususiyati va ikkinchi <book> elementidagi <title> elementining "lang" atributining atribut qiymati olinadi:

```

<?php
$xml=simplexml_load_file("books.xml") or die("Error: Cannot create
object");
echo $xml->book[0]['category'] . "<br>";
echo $xml->book[1]->title['lang'];
?>;

```

Quyidagi misol, "books.xml" faylidagi <title> elementlarining atribut qiymatlarini oladi:

```

<?php

```

```

$xml=simplexml_load_file("books.xml") or die("Error: Cannot create
object");
foreach($xml->children( ) as $books) {
    echo $books->title['lang'];
    echo "<br>";
}
?>;

```

Natijasi esa quyidagicha:

```

en
en
en-us
en-us.

```

The XML Expat Parser

Ichki XML Expat Parser, PHP-da XML hujjatlari bilan ishlashga imkon beradi. XML Expat parser – tartib asoslangan parser.

XML Expat Parserni ishga tushirish.

```

<?php
// Initialize the XML parser
$parser=xml_parser_create( );

// Function to use at the start of an element
function start($parser,$element_name,$element_attrs) {
    switch($element_name) {
        case "NOTE":
            echo "-- Note --<br>";
            break;
        case "TO":
            echo "To: ";
            break;
        case "FROM":
            echo "From: ";
            break;
        case "HEADING":
            echo "Heading: ";
            break;
        case "BODY":
            echo "Message: ";
    }
}
}

```

```

// Function to use at the end of an element
function stop($parser,$element_name) {
    echo "<br>";
}

// Function to use when finding character data
function char($parser,$data) {
    echo $data;
}

// Specify element handler
xml_set_element_handler($parser,"start","stop");

// Specify data handler
xml_set_character_data_handler($parser,"char");

// Open XML file
$fp=fopen("note.xml","r");

// Read data
while ($data=fread($fp,4096)) {
    xml_parse($parser,$data,feof($fp)) or
    die (sprintf("XML Error: %s at line %d",
    xml_error_string(xml_get_error_code($parser)),
    xml_get_current_line_number($parser)));
}

// Free the XML parser
xml_parser_free($parser);
?>.

```

Misoldagi funksiyalarni quyidagicha tushuntirish mumkin:

1. XML parserni `xml_parser_create ()` funksiyasi bilan ishga tushirish;
2. Har xil voqea ishlovchilarida ishlatish uchun funksiyalar yaratish;
3. Agar ajralish oynasi ochilganda va yopilish belgilariga duch kelganda, `Xml_set_element_handler ()` funksiyasini qo'shing;
4. Agar ajraluvchi belgi ma'lumotlari bilan to'qnashsa, `Xml_set_character_data_handler ()` funksiyasini qo'shing;
5. "Note.xml" faylini `xml_parse ()` funksiyasi bilan ajrating;
6. Agar xato bo'lsa, `xml_error_string ()` funksiyasini XML xatosini metinsel tavsifga aylantirish uchun qo'shing;

7. Xml_parser_create () funksiyasi bilan ajratilgan xotirani chiqarish uchun xml_parser_free () funksiyasiga murojaat qiling;

PHP XML DOM Parser

Ichki DOM parser XML fayllarini PHP-da ishlashga imkon beradi. DOM parser daraxt asosidagi parserdir. Quyidagi XML hujjat qismiga qarang:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<from>Jani</from>
```

DOM yuqoridagi XML-ni daraxt tuzilishi deb biladi:

1-darajali: XML hujjati

2-darajali: Ildiz element: <from>

3-darajali: Matn elementi: "Jani".

Yuklab olish va output XML

Demak, biz XML parserni ishga tushiramiz, XML-ni yuklaymiz va uni chiqaramiz:

```
<?php
```

```
$xmlDoc = new DOMDocument();
```

```
$xmlDoc->load("note.xml");
```

```
print $xmlDoc->saveXML( );
```

```
?>;
```

Yuqoridagi kodning chiqishi quyidagicha bo'ladi:

Tove Jani Reminder Don't forget me this weekend!

DOMDocument-ob'ekti va unga "note.xml" dan XML yuklaydi. SaveXML () funksiyasi ichki XML hujjatini satrga qo'yadi, shuning uchun uni chiqarishimiz mumkin.

Biz XML parserni ishga tushirib, XML va loopni <note> elementining barcha elementlari orqali yuklashni ko'ramiz:

```
<?php
```

```
$xmlDoc = new DOMDocument();
```

```
$xmlDoc->load("note.xml");
```

```
$x = $xmlDoc->documentElement;
```

```
foreach ($x->childNodes AS $item) {
```

```
    print $item->nodeName . " = " . $item->nodeValue . "<br>";
```

```
}
```

```
?>;
```

Yuqoridagi kodning chiqishi quyidagicha bo'ladi:

```
#text =
```

```
to = Tove
```

```
#text =
```

```
from = Jani
```

```
#text =
```

```

heading = Reminder
#text =
body = Don't forget me this weekend!
#text =;

```

Yuqoridagi misolda siz har bir element orasida bo'sh matnli tugunlar mavjudligini ko'rasiz.

XML yaratilganda, u odatda tugunlar orasidagi bo'shliqlarni o'z ichiga oladi. XML DOM parserlari ularni oddiy elementlar deb hisoblaydi va siz ulardan xabardor bo'lmasangiz, ular ba'zan muammolarga olib keladi.

PHP 5 libxml vazifalari

Libxml funksiyalari va doimiylari SimpleXML, XSLT va DOM funksiyalari bilan birgalikda ishlatiladi. Ushbu vazifalar libxml paketini talab qiladi. <http://www.xmlsoft.org/downloads.html> da yuklab oling.

PHP libxml funksiyalari

libxml_clear_errors()	Libxml xato bufferini tozalash
libxml_get_errors()	Ketma-ketlikdagi xatolarni aytarib olish
libxml_get_last_error()	Oxirgi xatoni libxml-dan olish
libxml_set_streams_context()	Keyingi libxml hujjatiga yuklash yoki yozish uchun oqimlar kontekstini belgilash
libxml_use_internal_errors(user_errors)	Libxml xatosini o'chirib qo'yish va foydalanuvchi kerak bo'lganda xato ma'lumotlarini olishiga imkon berish

3.1.1-jadval. PHP libxml funksiyalari.

PHP: funktsiyani qo'llab-quvvatlaydigan PHPning eng oxirgi versiyasini ko'rsatadi:

PHP libxml_clear_errors () funktsiyasi. Libxml_clear_errors () funktsiyasi libxml xato buferini tozalaydi.

Sintaksis: libxml_clear_errors ();

Misol: <?php

libxml_clear_errors()

?>;

PHP libxml_get_errors () funktsiyasi. Libxml_get_errors () funktsiyasi libxml xato buferidan xatoliklarni oladi. Ushbu funktsiya bir qator xato moslamalarni va libxml xato buferida xatolik bo'lmasa, bo'sh qatorni qaytaradi.

Sintaksis: libxml_get_errors ();

Misol: <?php


```
libxml_get_errors();
```

```
?>;
```

PHP `libxml_get_last_error()` funksiyasi. `Libxml_get_last_error()` funksiyasi oxirgi xatoni `libxml` xato buferidan oladi.

Sintaksis: `libxml_get_last_errors()`;

Misol: `<?php`

```
libxml_get_last_errors();
```

```
?>;
```

`Libxml_use_internal_errors()` funksiyasi standart `libxml` xatosini o'chirib qo'yadi va foydalanuvchi xatolarini boshqarish imkonini beradi. Bu funksiya `use_lib_errors` parametrining oldingi qiymatini qaytaradi.

Sintaksis: `libxml_use_internal_errors(user_errors)`;

Misol: `<?php`

```
libxml_use_internal_errors(user_errors);
```

```
?>;
```

PHP 5 SimpleXML vazifalari.

Funksiya	Tavsifi
<code>__construct()</code>	Yangi SimpleXMLElement ob'ektini yaratadi.
<code>addChild()</code>	SimpleXML elementini bir kichik element qo'shib qo'yadi.
<code>asXML()</code>	SimpleXML ob'ektidan yaxshi shakllangan XML stringini (XML versiyasi 1.0) qaytaradi.
<code>attributes()</code>	Bir elementning atributlari / qiymatlarini qaytaradi
<code>children()</code>	Belgilangan tugunning bolalarini qaytaradi
<code>count()</code>	Belgilangan tugunning bolalarini hisoblaydi
<code>getDocNamespaces()</code>	Hujjatda DECLARED nom maydonlarini qaytaradi
<code>getName()</code>	SimpleXML elementi tomonidan havola qilingan XML tagining nomini qaytaradi
<code>getNamespaces()</code>	Hujjatda ishlatilgan nom maydonlarini qaytaradi
<code>registerXPathNamespace()</code>	Keyingi XPath so'rovi uchun nom

	maydoni kontekstini yaratadi
saveXML () asXML ()	nomini o'zgartirish
simplexml_load_file ()	XML faylini SimpleXMLElement ob'ektiga o'zgartiradi
simplexml_load_string ()	XML dizisini SimpleXMLElement ob'ektiga o'zgartiradi
xpath ()	XML ma'lumotlari bo'yicha XPath so'rovini ishga tushiradi
simplexml_import_dom ()	DOM tugunidan SimpleXMLElement ob'ektini qaytaradi

3.1.2-jadval. PHP 5 SimpleXML vazifalari.

PHP 5 SimpleXML iteratsiya funksiyalari.

Funksiya	Tavsifi
current ()	Joriy elementni qaytaradi
getChildren ()	Joriy elementning pastki elementlarini qaytaradi
hasChildren ()	Joriy element bolaligini tekshiradi
key ()	Joriy kalitni qaytadi.
next ()	Keyingi elementga o'tadi
rewind()	Birinchi elementga qaytarish
current ()	Joriy element joriy yoki yo'qligini tekshirish

3.1.3-jadval. SimpleXML iteratsiya funksiyalari.

__construct () - __construct () funksiyasi yangi SimpleXMLElement ob'ektini yaratadi. Sintaksisi:

`__construct(data,options,data_is_url,ns,is_prefix);`

data - Majburiy. Data_is_url TRUE bo'lsa, yaxshi hosil qilingan XML string yoki XML hujjatiga yo'l yoki URL belgilaydi.

options – Majburiy emas. Qo'shimcha Libxml parametrlarini belgilaydi. Variantni belgilash bilan belgilanadi va 1 yoki 0 (TRUE yoki FALSE, masalan, LIBXML_NOBLANKS (1)).

data_is_url - Majburiy emas. TRUE ma'lumotlar ma'lumoti o'rniga XML hujjatiga yo'l / URL ekanligini bildiradi. Standart holatda FALSE turadi.

ns - Optional. Specifies a namespace prefix or URI.

is_prefix - Majburiy emas. Boolean qiymati bildiradi. Agar ns - bu prefiks bo'lsa, TRUE. Agar ns URI bo'lsa, FALSE. Standart u ham FALSE turadi.

addChild () - SimpleXML elementini bir kichik element qo'shib qo'yadi. Sintaksisi: `addAttribute(name,value,ns);` parametrlari - **name-** Majburiy.

Qo'shiladigan atribut nomini bildiradi. **value** - Majburiy emas. Xususiyatning qiymatini belgilaydi. **ns** - Majburiy emas. Atribut uchun nom maydoni bildiradi.

Misol

Ildiz elementiga (<note>) va <body> elementiga bir atribut qo'shish:

```
<?php
$xml = <<<XML
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
XML;
```

```
$xml = new SimpleXMLElement($note);
$xml->addAttribute("type","private");
$xml->body->addAttribute("date","2014-01-01");
```

```
echo $xml->asXML();
?>;
<?php
$xml = <<<XML
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
XML;
```

```
$xml = new SimpleXMLElement($note);
$xml->addAttribute("type","private");
$xml->body->addAttribute("date","2014-01-01");
```

```
echo $xml->asXML();
?>;
```

AddChild () - funksiyasi SimpleXML elementiga bir kichik elementni qo'shadi. Sintaksis: `addChild(name,value,ns)`;

name - Majburiy. Qo'shiladigan pastki element nomini bildiradi.

value - Majburiy emas. Bola elementining qiymatini belgilaydi.

ns - Majburiy emas. Bola elementi uchun nom maydonini bildiradi.

Misol. Bola elementini <body> elementiga va yangi <footer> elementiga qo'shish:

```
<?php
$note=<<<XML
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
XML;
```

```
$xml = new SimpleXMLElement($note);
```

```
// Add a child element to the body element
$xml->body->addChild("date","2014-01-01");
```

```
// Add a child element after the last element inside note
$footer = $xml->addChild("footer","Some footer text");
```

```
echo $xml->asXML();
```

```
?>;
```

asXML() funksiyasi - sodda XML ob'ektini (XML versiyasi 1.0) SimpleXML ob'ektidan qaytaradi. Sintaksisi: asXML(filename); filename – Majburiy emas. Belgilangan bo'lsa, funktsiya bu ma'lumotni faylga yozib bo'lmaydi, lekin uni qaytaradi.

Misol. Formatlangan XML stringini (XML versiyasi 1.0) SimpleXML obyektidan qaytarish:

```
<?php
$note=<<<XML
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
XML;
```

```
$xml = new SimpleXMLElement($note);
```

```
echo $xml->asXML();
```

?>;

Atributs () funktsiyasi - XML yorlig'i ichidagi atributlar va qiymatlarni qaytaradi. Sintaksisi: atributs (ns, is_prefix);

ns - Majburiy emas. Qabul qilingan atributlar uchun nom maydonini bildiradi.

is_prefix - Majburiy emas. Boolean qiymati bildiradi. Agar ns - bu prefiks bo'lsa, TRUE . Agar ns URI bo'lsa, FALSE. Standart FALSE.

Misol. XML <body> elementi ichidagi atributlar va qiymatlarni qaytarish:

```
<?php
$note=<<<<XML
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body date="2014-01-01" type="private">Don't forget me this weekend!</body>
</note>
XML;
```

```
$xml = simplexml_load_string($note);
foreach($xml->body[0]->attributes() as $a => $b)
{
    echo $a,'=', $b,"<br>";
}
?>;
```

Count () - funktsiyasi ma'lum tugunli bolalarni hisobga oladi. Sintaksisi: count();

Misol. Avtomobil tugunlarining bolalarini hisoblash:

```
<?php
$xml=<<<<XML
<cars>
    <car name="Volvo">
        <child/>
        <child/>
        <child/>
        <child/>
    </car>
    <car name="BMW">
        <child/>
        <child/>
    </car>
</cars>
```

XML;

```
$elem=new SimpleXMLElement($xml);
foreach ($elem as $car)
{
    printf("%s has %d children.<br>", $car['name'], $car->count());
}
?>
```

GetDocNamespaces () - funksiyasi XML hujjatida e'lon qilingan ismlarni qaytaradi. Sintaksisi: getDocNamespaces (recursive, from_root);

recursive - Majburiy emas. Boolean qiymati bildiradi. TRUE bo'lsa, ota-ona va chaqaloq tugunlarida e'lon qilingan barcha nomlar qaytariladi. FALSE bo'lsa, faqat ildiz tugunida e'lon qilingan ismlar qaytariladi. Standart FALSE turadi. **from_root** - Majburiy emas. Boolean qiymati bildiradi. ROLE XML doc ning ildizidan nomini tekshiring. FALSE, bola tugunidagi nom satrini tekshiring. Standart qiymati TRUE. Misol: XML hujjatining ildizida e'lon qilingan ismlarni qaytarish.

```
<?php
$xml=<<<<XML
<?xml version="1.0" standalone="yes"?>
<cars xmlns:c="http://w3schools.com/ns">
    <c:car id="1">Volvo</c:car>
    <c:car id="2">BMW</c:car>
    <c:car id="3">Saab</c:car>
</cars>
```

XML;

```
$sxe=new SimpleXMLElement($xml);
$ns=$sxe->getDocNamespaces();
print_r($ns);
?>
```

GetName () - funksiyasi XML element nomini qaytaradi. Sintaksis: getName();

Misol: XML elementini va bolalarni nomini qaytarish:

```
<?php
$xml=<<<<XML
<?xml version="1.0" standalone="yes"?>
<cars>
    <car id="1">Volvo</car>
    <car id="2">BMW</car>
    <car id="3">Saab</car>
</cars>
```

XML;

```

$sxe=new SimpleXMLElement($xml);
echo $sxe->getName() . "<br>";
foreach ($sxe->children() as $child)
{
    echo $child->getName() . "<br>";
}
?>

```

GetNamespaces () - funktsiyasi XML hujjatida ishlatiladigan nom bo'shligini qaytaradi. Sintaksisi: `getNamespaces(recursive)`; **recursive** - Majburiy emas. Boolean qiymati bildiradi. TRUE bo'lsa, ota-ona va chaqaloq tugunlarida e'lon qilingan barcha nomlar qaytariladi. FALSE bo'lsa, faqat ildiz tugunida e'lon qilingan ismlar qaytariladi. Standart FALSE bo'ladi.

Misol: XML hujjatida ishlatiladigan nom maydonlarini qaytarish.

```

<?php
$xml=<<<XML
<?xml version="1.0" standalone="yes"?>
<cars                                xmlns:c="http://w3schools.com/ns"
xmlns:a="http://w3schools.com/country">
    <c:car id="1">Volvo</c:car>
    <c:car id="2">BMW</c:car>
    <c:car id="3">Saab</c:car>
</cars>
XML;
$sxe=new SimpleXMLElement($xml);
$ns=$sxe->getNamespaces(true);
var_dump($ns);
?>

```

RegisterXPathNamespace () - funktsiyasi keyingi XPath so'rovi uchun nom maydoni kontekstini yaratadi. Sintaksisi: `registerXPathNamespace(prefix,ns)`; **prefix**- Majburiy. NSda berilgan nom maydoni uchun XPath so'rovida ishlatiladigan nom maydoni prefiksini bildiradi. **ns** - Majburiy. XPath so'rovi uchun ishlatiladigan nom maydonini bildiradi. Misol: Keyingi XPath so'rovi uchun nom maydoni kontekstini yaratish.

```

<?php
$xml=<<<XML
<book xmlns:chap="http://example.org/chapter-title">

```

```

<title>My Book</title>
<chapter id="1">
  <chap:title>Chapter 1</chap:title>
  <para>Donec velit. Nullam eget tellus...</para>
</chapter>
<chapter id="2">
  <chap:title>Chapter 2</chap:title>
  <para>Lorem ipsum dolor sit amet....</para>
</chapter>
</book>

```

XML;

```

$sxe=new SimpleXMLElement($xml);
$sxe->registerXPathNamespace('c','http://example.org/chapter-title');
$result=$sxe->xpath('//c:title');
foreach ($result as $title)
{
  echo $title . "<br>";
}
?>

```

Natija:

Chapter 1

Chapter 2.

Simplexml_import_dom() - funktsiyasi, DOM tugunidan

SimpleXMLElement ob'ektini qaytaradi. Sintaksisi:

simplexml_import_dom(node,classname); **node** - Majburiy. DOM elementi tugunni bildiradi. **classname** - Majburiy emas. Yangi ob'ekt sinfini bildiradi.

Misol: DOM hujjatidan tugunni olish va uni SimpleXML tuguniga aylantirish;

```
<?php
```

```
$dom=new domDocument;
```

```
$dom->loadXML("<note><to>Tove</to><from>Jani</from></note>");
```



```
$x=simplexml_import_dom($dom);
```

```
echo $x->from;
```

```
?>
```

Simplexml_load_file () funksiyasi ko'rsatilgan XML faylini

SimpleXMLElement ob'ektiga o'zgartiradi. Sintaksisi:

```
simplexml_load_file(file,classname,options,ns,is_prefix);
```

file - Majburiy. XML faylining yo'lini bildiradi. **classname** - Majburiy emas.

Yangi ob'ekt sinfini bildiradi. **options** - Majburiy emas. Qo'shimcha Libxml parametrlarini belgilaydi. Variantni belgilash bilan belgilanadi va 1 yoki 0 (TRUE yoki FALSE, masalan, LIBXML_NOBLANKS (1)).

LIBXML_COMPACT - Tugmalarni ajratishni optimallashtirishni faollashtirish (ilovani tezlashtirishi mumkin).

LIBXML_DTDATTR - Standart DTD atributlarini o'rnatish.

LIBXML_DTDLOAD - tashqi to'plamni o'rnatish.

LIBXML_DTDVALID - DTD bilan tasdiqlash.

LIBXML_NOBLANKS - bo'sh tugunlarni olib tashlash.

LIBXML_NOCDATA - CDATAni matnli tugunlar sifatida birlashtirish.

LIBXML_NOEMPTYTAG - DOMDocument-> save () va DOMDocument-> saveXML () funksiyalarida mavjud bo'lgan bo'sh teglar (masalan,
 dan
 ga) kengaytirilsin

LIBXML_NOENT - O'zgartirish ob'ektlari.

LIBXML_NOERROR - Xato haqida hisobot ko'rsatma.

LIBXML_NONET - Hujjatlarni yuklashda tarmoqqa ulanishni o'chirib qo'yish.

LIBXML_NOWARNING - ogohlantirish hisobotlarini ko'rsatma.

LIBXML_NOXMLDECL - Hujjatni saqlash vaqtida XML deklaratsiyasini qoldirish.

LIBXML_NSCLEAN - keraksiz nom maydoni bildirgilarini olib tashlash.

LIBXML_PARSEHUGE - XML_PARSE_HUGE bayrog'ini o'rnatadi, bu esa har qanday kodlashtirilgan chegarani ajralish moslamasidan yumshatadi. Bu hujjatning maksimal chuqurligi va matnli tugun kattaligi chegaralariga ta'sir qiladi.

LIBXML_XINCLUDE - XInclude o'zgarishini bajarish.

LIBXML_ERR_ERROR - Qayta tiklanadigan xatoliklarni qabul qilish.

LIBXML_ERR_FATAL - muhim xatoliklarni olish.

LIBXML_ERR_NONE - Hech qanday xatolik yo'q.

LIBXML_ERR_WARNING - oddiy ogohlantirishlarni olish.

LIBXML_VERSION - libxml versiyasini oling (m-n, 20605 yoki 20617).

LIBXML_DOTTED_VERSION - nuqta libxml versiyasini oling (m-n, 2.6.5 yoki 2.6.17).

ns - Majburiy emas. Nom maydonining oldingi yoki URI ni bildiradi. **is_prefix** - Majburiy emas. Boolean qiymati bildiradi. Agar ns - bu prefiks bo'lsa, TRUE. Agar ns URI bo'lsa, FALSE. Standart FALSE.

Misol

XML faylini SimpleXMLElement ob'ektiga, so'ngra chiqadigan kalitlarga va ob'ekt elementlariga aylantirish:

```
<?php
$xml=simplexml_load_file("note.xml");
print_r($xml);
?>
```

Natija: SimpleXMLElement Object ([to] => Tove [from] => Jani [heading] => Reminder [body] => Don't forget me this weekend!);

Xpath () - funktsiyasi XML hujjatida XPath so'rovini bajaradi. Sintaksisi: class SimpleXMLElement{string xpath(path)}; **path** - Majburiy. XML hujjatida nimani qidirish kerakligini belgilaydi. Misol XML fayl:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

PHP kodi:

```
<?php
```

```
$xml = simplexml_load_file("test.xml");
$result = $xml->xpath("from");
print_r($result); ?>
```

Yuqoridagi kodning chiqishi quyidagicha bo'ladi:

```
Array ([0] => SimpleXMLElement Object ([0] => Jani ) )
```

PHP XML Parser Funktsiyalari.

Utf8_decode () - funktsiyasi UTF-8 simini ISO-8859-1 kodi bilan kodlaydi.

Sintaksis: utf8_decode (string); **string** - Majburiy. Kodni kodlash uchun belgilaydi.

Xml_error_string () - funktsiyasi XML ajralish xatosi tavsifini oladi. Bu funktsiya xato tavsifini natijasiga qaytaradi, yoki xato yuz berganda FALSE ni qaytaradi. **Sintaksisi:** xml_error_string(errorcode); **errorcode** - Majburiy. Foydalanish uchun xato kodini bildiradi. Xato kodi xml_get_error_code () funktsiyasidan qaytish qiymati. Misol:

```
<?php
//invalid xml file
$xmlfile = 'test.xml';
$xmlparser = xml_parser_create();
// open a file and read data
$fp = fopen($xmlfile, 'r');
while ($xmldata = fread($fp, 4096))
{
// parse the data chunk
if (!xml_parse($xmlparser,$xmldata,feof($fp)))
{
die( print "ERROR: "
. xml_error_string(xml_get_error_code($xmlparser))
. "<br />"
. "Line: "
. xml_get_current_line_number($xmlparser)
```

```

. "<br />"
. "Column: "
. xml_get_current_column_number($xmlparser)
. "<br />");
}
}

```

```
xml_parser_free($xmlparser);
```

```
?>;
```

Kodni natijasi:

ERROR: Mismatched tag

Line: 5

Column: 41;

Xml_get_current_byte_index () - funktsiyasi XML Parser uchun bayt indeksini oladi. Ushbu funktsiya joriy bayt ko'rsatkichini muvaffaqiyatga qaytaradi yoki xato yuz berganda FALSEni qaytaradi. Sintaksisi: `xml_get_current_byte_index(parser);` **parser** - Majburiy. Foydalanadigan XML parserini belgilaydi. Misol:

```

<?php
//invalid xml file
$xmlfile = 'test.xml';
$xmlparser = xml_parser_create();
// open a file and read data
$fp = fopen($xmlfile, 'r');
while ($xmldata = fread($fp, 4096))
{ // parse the data chunk
if (!xml_parse($xmlparser,$xmldata,feof($fp)))
{ die( print "ERROR: "
. xml_error_string(xml_get_error_code($xmlparser))
. "<br />"
. "Line: "

```

```

. xml_get_current_line_number($xmlparser)
. "<br />"
. "Column: "
. xml_get_current_column_number($xmlparser)
. "<br />"
. "Byte Index: "
. xml_get_current_byte_index($xmlparser)
. "<br />"); } }

```

```
xml_parser_free($xmlparser);
```

```
?>;
```

Natijasi quyidagicha:

ERROR: Mismatched tag

Line: 5

Column: 41

Byte Index: 72;

xml_get_current_column_number () - funktsiyasi XML sintaksik tahlil qilish uchun mavjud ustun sonini oladi. Sintaksisi: `xml_get_current_column_number(parser);` **parser** - Foydalanadigan XML parserini belgilaydi. Misol:

```

<?php
//invalid xml file
$xmlfile = 'test.xml';
$xmlparser = xml_parser_create();
// open a file and read data
$fp = fopen($xmlfile, 'r');
while ($xmldata = fread($fp, 4096))
{ // parse the data chunk
if (!xml_parse($xmlparser,$xmldata,feof($fp)))
{ die( print "ERROR: "
. xml_error_string(xml_get_error_code($xmlparser))
. "<br />"

```

```

    . "Line: "
    . xml_get_current_line_number($xmlparser)
    . "<br />"
    . "Column: "
    . xml_get_current_column_number($xmlparser)
    . "<br />"); } }
xml_parser_free($xmlparser);
?>;

```

Natijasi:

ERROR: Mismatched tag

Line: 5

Column: 41.

xml_get_current_line_number () - funktsiyasi XML tekshiruvchi uchun joriy satr raqamini oladi. Sintaksis: `xml_get_current_line_number (parser);` **parser** - Majburiy. Foydalanadigan XML parserini belgilaydi. Misol:

```

<?php
//invalid xml file
$xmlfile = 'test.xml';
$xmlparser = xml_parser_create();
// open a file and read data
$fp = fopen($xmlfile, 'r');
while ($xmldata = fread($fp, 4096))
{ // ma'lumotlar to'plamini ajratib ko'rsatish
if (!xml_parse($xmlparser,$xmldata,feof($fp)))
{ die( print "ERROR: "
. xml_error_string(xml_get_error_code($xmlparser))
. "<br />" . "Line: "
. xml_get_current_line_number($xmlparser) . "<br />"
. "Column: "
. xml_get_current_column_number($xmlparser) . "<br />"); } }
xml_parser_free($xmlparser);?>.

```

Natijasi:

ERROR: Mismatched tag

Line: 5

Column: 41.

xml_get_error_code () - funktsiyasi XML ajratuvchi xato kodini oladi.

Sintaksis: `xml_get_error_code (parser)`; **parser** - Majburiy. Foydalanadigan XML parserini belgilaydi. Misol:

```
<?php
//invalid xml file
$xmlfile = 'test.xml';
$xmlparser = xml_parser_create();
// open a file and read data
$fp = fopen($xmlfile, 'r');
while ($xmldata = fread($fp, 4096))
{
// parse the data chunk
if (!xml_parse($xmlparser,$xmldata,feof($fp)))
{
die( print "ERROR: "
. xml_error_string(xml_get_error_code($xmlparser))
. "<br />"
. "Line: "
. xml_get_current_line_number($xmlparser)
. "<br />"
. "Column: "
. xml_get_current_column_number($xmlparser)
. "<br />");
} }

```

```
xml_parser_free($xmlparser); ?>
```

Natijasi:

ERROR: 76

Line: 5

Column: 41

xml_parse () - funksiyasi XML hujjatini ajratadi. Sintaksisi: `xml_parse (parser, xml, end)`; **parser** - Majburiy. Foydalanadigan XML parserini belgilaydi. **xml** - Majburiy. Tahrirlash uchun XML ma'lumotlarini belgilaydi. **end** - Majburiy emas. Agar bu parametr TRUE bo'lsa, "XML" parametridagi ma'lumotlar ushbu parse ichida yuborilgan so'nggi ma'lumotdir. Maslahat: XML parserini yaratish uchun `xml_parser_create ()` funksiyasidan foydalaning. Misol:

XML File

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

PHP code

```
<?php
$parser=xml_parser_create();
function char($parser,$data) { echo $data; }
xml_set_character_data_handler($parser,"char");
$fp=fopen("test.xml","r");
while ($data=fread($fp,4096))
{ xml_parse($parser,$data,feof($fp)) or
die (sprintf("XML Error: %s at line %d",
xml_error_string(xml_get_error_code($parser)),
xml_get_current_line_number($parser))); }
xml_parser_free($parser);?>
```

Natijasi:

Tove Jani Reminder Don't forget me this weekend!

3.2 PHP orqali XML ma`lumotlarini o`zgartirish, qo`shish, o`chirish

```
// malumot.xml faylining ma`lumoti
```

```
<?xml version='1.0'?>
<maxsulotlar>
  <maxsulot pid='314'>
    <nomi>Olma</nomi>
    <narxi>$1.00</narxi>
    <chegirma>3%</chegirma>
  </maxsulot>
  <maxsulot pid='315'>
    <nomi>O`rik</nomi>
    <narxi>$0.90</narxi>
    <chegirma>3%</chegirma>
  </maxsulot>
  <maxsulot pid='316'>
    <nomi>Anor</nomi>
    <narxi>$0.95</narxi>
    <chegirma>4%</chegirma>
  </maxsulot>
  <maxsulot pid='317'>
    <nomi>Nok</nomi>
    <narxi>$1.90</narxi>
    <chegirma>5%</chegirma>
  </maxsulot>
</maxsulotlar>
```

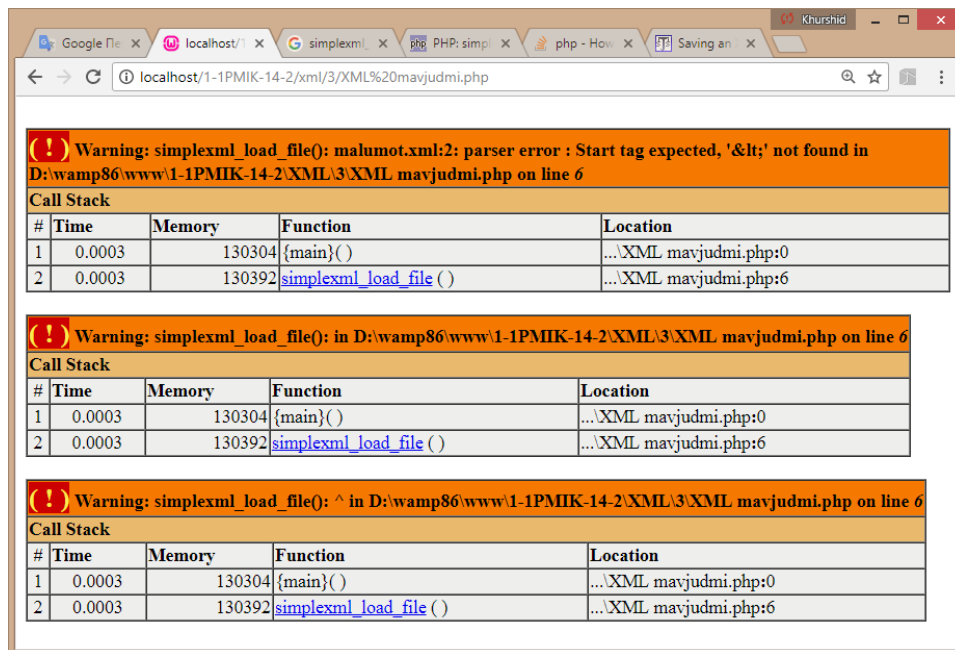
XML fayli ma`lumotini keltirdik endilikda shu malumot.xml bilan ishlaymiz. PHP orqali XML fayli mavjudligini tekshiramiz. Bunda agar xml fayli mavjud bo`lmasa fayl o`qilmadi deb aytadi. Ammo XML faylini ichida hech bo`lmaganda bitta element bo`lishi kerak.

```
//XML fayli mavjudligini tekshirish **XML mavjudmi.php**
```

```
<?php
// malumot.xml fayli mavjudmi
// XML-hujjatda hech bo`lmasa bitta elemen bor /[root]/title.

if (file_exists('malumot.xml')) {
  $xml = simplexml_load_file('malumot.xml');
  print_r($xml);
} else { exit('malumot.xml faylini o`qib bo`lmadi. '); } ?>
```

Agar faylni ichida element mavjud bo`lmasa quyidagi xatolik chiqadi



3.2.1 – rasm. Element mavjud bo`lmagandagi xatolik

Biz PHP orqali XML fayliga ma`lumot qo`shmoqchimiz. Buning uchun quyidagi misolni ko`rib chiqamiz.

```
// XML da element qo`shish.php
<?php
$xml=simplexml_load_file("malumot.xml"); // XML ni o`qish
$newchild = $xml->addChild("maxsulot");
//Qo`shiladigan yozuv parametrlari
$newchild->addAttribute("pid", 318);
$newchild->addChild("nomi", "Gilos");
$newchild->addChild("narxi", "$0.20");
$newchild->addChild("chegirma", "0.7%");
echo $xml->asXML(); // XML elementlarini chop etish
// $xml->asXML("malumot.xml"); XML fayliga saqlash
?>
```

Agar XML fayliga qo`shilgan ma`lumotni yozmoqchi bo`lsak \$xml->asXML("malumot.xml"); buyruqni o`zi yetarli. Saqlanmagan vaqtda faqat \$xml o`zgaruvchisiga qo`shilganligicha qoladi xolos.

PHP orqali XML fayliga ma`lumotini o`chirmoqchimiz. Buning uchun quyidagi misolni ko`rib chiqamiz.

```
// XML da element o`chirish.php
<?php
//O`rik maxsulotini o`chirish
$maxsulotId = 315;
$dom = new DomDocument;
$dom->load("malumot.xml");
```

```

// o`chirilishi kerak bo`lganlarni izlash
$xmlpath = new DOMXPath($dom);
$xmlodelist = $xmlpath->query("/maxsulotlar/maxsulot[@pid={$maxsulotId}]");
$xmloldnode = $xmlodelist->item(0);
// elementni o`chirish
$xmloldnode->parentNode->removeChild($xmloldnode);
echo $dom->saveXML();
// $dom->save("malumot.xml"); o`chirilgandan keyin saqlash
?>

```

PHP orqali XML faylidagi ma`lumotlarni o`zgartirimoqchimiz. Buning uchun quyidagi misolni ko`rib chiqamiz.

// XML da element almashtirish.php

```

<?php
    $maxsulotId = 314;
    $parent = new DomDocument;
    // yangi product maxsulotini yaratamiz
    $parent_node = $parent->createElement('maxsulot');
    // atribut qo`shamiz
    $attribute = $parent->createAttribute("pid");
    // qiymat o`rnatamiz
    $attribute->value = $maxsulotId;
    $parent_node->appendChild($attribute);
    // farzand (navda) elementlarini qo`shamiz
    $parent_node->appendChild($parent->createElement('nomi', "Uzum"));
    $parent_node->appendChild($parent->createElement('narxi', "$3.00"));
    $parent_node->appendChild($parent->createElement('chegirma', "1%"));
    // 'maxsulot' yaratilgan elementga qo`shamiz
    $parent->appendChild($parent_node);
    // XML formatni yukalaymiz
    $dom = new DomDocument;
    $dom->load("malumot.xml");
    // pid = 314 bo`lgan elementni topamiz
    $xmlpath = new DOMXPath($dom);
    $xmlodelist = $xmlpath->query("/maxsulotlar/maxsulot[@pid={$maxsulotId}]");

    $xmloldnode = $xmlodelist->item(0);

    // Yangi yaratilgan elementni import qilamiz
    $xmlnewnode = $dom->importNode($parent->documentElement, true);

    // Oldingi elementni yangisi bilan almashamiz
    $xmloldnode->parentNode->replaceChild($xmlnewnode, $xmloldnode);
    // XML ni saqlaymiz
    echo $dom->saveXML();
    // $dom->save("malumot.xml"); o`chirilgandan keyin saqlash

```

?>

3.3 PHP orqali XML ma`lumotlariga XPath, XQuery orqali ma`lumotlarini tanlash

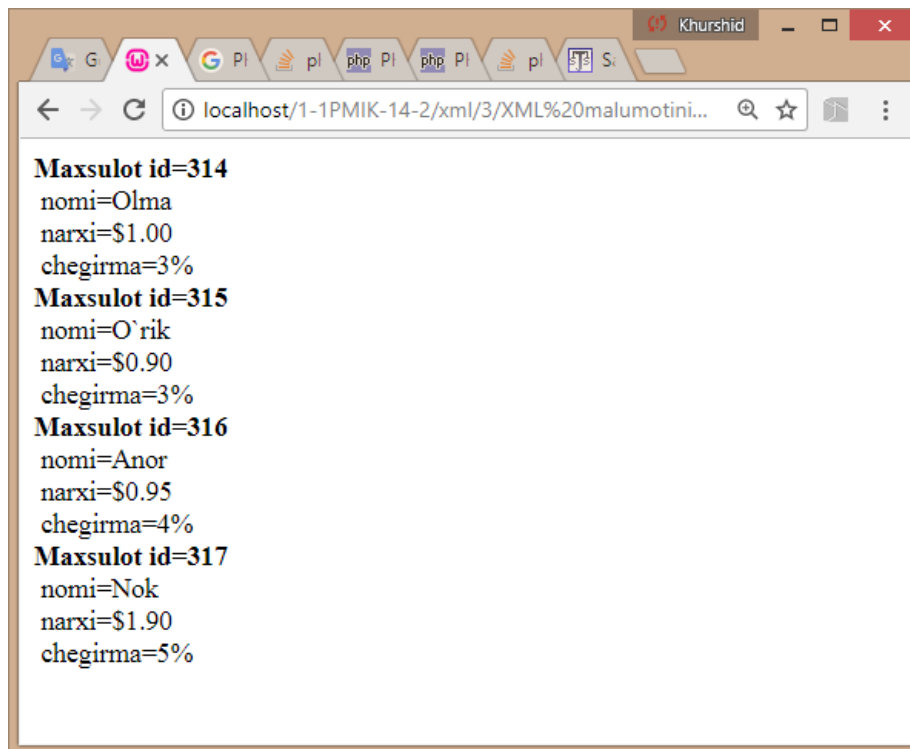
// malumot.xml faylining ma`lumoti

```
<?xml version='1.0'?>
<maxsulotlar>
  <maxsulot pid='314'>
    <nomi>Olma</nomi>
    <narxi>$1.00</narxi>
    <chegirma>3%</chegirma>
  </maxsulot>
  <maxsulot pid='315'>
    <nomi>O`rik</nomi>
    <narxi>$0.90</narxi>
    <chegirma>3%</chegirma>
  </maxsulot>
  <maxsulot pid='316'>
    <nomi>Anor</nomi>
    <narxi>$0.95</narxi>
    <chegirma>4%</chegirma>
  </maxsulot>
  <maxsulot pid='317'>
    <nomi>Nok</nomi>
    <narxi>$1.90</narxi>
    <chegirma>5%</chegirma>
  </maxsulot>
</maxsulotlar>
```

PHP orqali XML faylidagi ma`lumotlarni o`qish quyidagicha

// XML malumotini o`qishda element qo`shish.php

```
<?php
$xml=simplexml_load_file("malumot.xml"); // XML ni o`qish
foreach ($xml->children() as $second) {
  echo '<b> Maxsulot id='. $second['pid']. "</b><br>";
  foreach ($second->children() as $third) {
    echo "&nbsp;". $third->getName().'=';
    if ($third->getName()=='nomi') echo $second->nomi.'<br>';
    if ($third->getName()=='narxi') echo $second->narxi.'<br>';
    if ($third->getName()=='chegirma') echo $second->chegirma.'<br>';
  }
} ?>
```



3.3.1 – rasm. XML faylining ma`lumotini element bo`yicha birma bir o`qish

PHP orqali XML ma`lumotlarni o`qish

```

<?php
$xml=simplexml_load_file("malumot.xml"); // XML ni o`qish
foreach ($xml->maxsulot as $m) {
    echo '<p><b>Maxsulot id ='. $m['pid']. "</b><br>";
    echo ' nomi='. $m->nomi. '<br>';
    echo ' narxi='. $m->narxi. '<br>';
    echo ' chegirma='. $m->chegirma. '<br></p>';
} ?>

```

PHP orqali XML ma`lumotlariga XPath buyrug`I orqali filtrlarni qo`llashga misol

```

<?php

$xml=simplexml_load_file("malumot.xml"); // XML ni o`qish
//$xpath_results = $xml -> Xpath("//maxsulot");
//$xpath_results = $xml -> Xpath("//maxsulot[2]");
//$xpath_results = $xml -> Xpath("//maxsulot[last()]");
//$xpath_results = $xml -> Xpath("//maxsulot[position()<4]");
//$xpath_results = $xml -> Xpath("//maxsulot[@pid]");
//$xpath_results = $xml -> Xpath("//maxsulot[@pid=315]");
//$xpath_results = $xml -> Xpath("//maxsulot[narxi='$1.00']");
$xml_results = $xml ->
    Xpath("//maxsulot[contains(narxi,'$1.')]");

foreach ($xml_results as $m) {

```

```

echo '<p><b>Maxsulot id ='. $m['pid']. "</b><br>";
echo ' nomi='. $m->nomi. '<br>';
echo ' narxi='. $m->narxi. '<br>';
echo ' chegirma='. $m->chegirma. '<br></p>';
}

```

```

// $xpath_results = $xml -> XPath("//maxsulot[position()<4]/nomi");
// $xpath_results = $xml -> XPath("//maxsulot[position()<4]/nomi[text()]");
// print_r ($xpath_results);
/*
foreach ($xpath_results as $m)
echo $m. "<br>";
*/
?>

```

XML ma`lumotlarini PHP orqali XPath->query orqali olish

```

<?php
$xmlDocument = new DOMDocument();
if ($xmlDocument->load('malumot.xml')) {
$xmlPath = new DOMXPath($xmlDocument);

$xmlNodeList = $xmlPath->query(
'/maxsulotlar/maxsulot[last()]',
'//maxsulot[position()<4]/nomi',
'//maxsulot[2]',
$xmlDocument);
header('Content-Type: text/plain');
foreach ($xmlNodeList as $xmlNode) {
echo $xmlDocument->saveXML($xmlNode) . "\r\n";
}
} ?>

```

Foydalanilgan adabiyotlar

1. Karimov I.A. «Yuksak ma'naviyat yengilmas kuch».
2. Каримов И.А. Баркамол авлод орзуси,- Ташкент «Шарк», 1999.
3. Котеров Д.В., Костарев А.Ф. PHP 5. – СПб.: БХВ-Петербург, 2005. – 1120 с.: ил.
4. Стивен Хольцнер. PHP в примерах. Пер. с англ. – ООО «Бином-Пресс», 2007 г. – 352 с.: ил.
5. Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Andrei Zmievski, and Jouni Ahto. PHP Manual. 2003 year. 4081 page.
6. Matt Doyle. Beginning PHP 5.3. Wiley Publishing, Inc. 2010 year. 841 page. ISBN: 978-0-470-41396-8.
7. Rasmus Lerdorf and Kevin Tatroe. Programming PHP. O'Reilly & Associates, Inc. 2002 year. 508 page. ISBN: 1-56592-610-2
8. Алексей Старыгин. XML разработка Web – приложений. Мастер-программ 2002 г. 586 стр.
9. Гарольд Э., Минс С. XML. Справочник. – Пер. с англ. – СПб: Символ-Плюс, 2002. – 576 с., ил. ISBN 5-93286-025-1
10. Хабибулин И.Ш. Самоучитель XML. – СПб: БХВ-Петербург, 2003. - 336 с.: ил. ISBN 5-94157-339-1
11. Рэй Э. Изучаем XML. – Пер. с англ. – СПб: Символ-Плюс, 2001. – 408 с., ил. ISBN 5-93286-023-5
12. Грейвс, Марк. Проектирование баз данных на основе XML. : Пер. с англ. — М, : Издательский дом “Вильямс”, 2002. — 640с. : ил. — Парал. тит. англ. ISBN 5-8459-0270-3 (рус.).

Foydalanilgan internet manzillar:

13. www.ziyonet.uz
14. <https://www.w3schools.com>
15. <https://www.w3schools.com/php>
16. <https://www.w3schools.com/xml>
17. <http://www.bestjquery.com/example/jquery-php/>

