

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС
ТАЪЛИМ ВАЗИРЛИГИ
БУХОРО ОЗИҚ-ОВҚАТ ВА ЕНГИЛ САНОАТ ТЕХНОЛОГИЯСИ
ИНСТИТУТИ



«ИНФОРМАТИКА ВА АХБОРОТ ТЕХНОЛОГИЯЛАРИ»
КАФЕДРАСИ

Реферат

МАВЗУ: PHP хакида умумий тушунча

Бажарди:

**12-03 МИИТ гурухи
талабаси Хожиев Азиз.**

Текширди:

Йўлдошев Ш.С.

БУХОРО – 2006

Кириш

Охирги йиллар мобайнида Web sayt компьютердан фойдаланувчиларнинг ҳаёт тарзига айланди, у дастур ва уйинларни кизикарли ва янада таъсуротли қилади. Хозирда бундай дастурлардан фойдаланиш учун мультимедия жуда зарур. Мультимедия соҳасидаги ривожланиш 1989 йилда узига катта ва мураккаб мультимедия иловаларини сиғдира оладиган йиғувчи компакт дискларда пайдо бўлганидан бошланди.

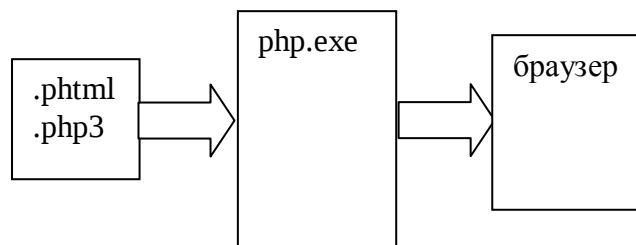
1994 йилда бошқа ривожланиш ҳам бошланди, бу—World,Wide,Web. World Wide Web - («Бутун жаҳон ургимчаги»). Web саҳифалардан тукилган бўлиб, узида Web-сайт тематикасига боглик ҳар хил маълумотларга эга. Интернетнинг фойдалилиги ҳисоблаш технологиясининг ривожланиши билан бирга тахминан 10 йилга орқада қолган ҳолда ушиб борарди. 80-йиллар охирида шахсий компьютерларнинг пайдо бўлиши информатикани билимдонлардан кенг оммага олиб ўтди. Интернет ривожланиши ҳамда, ҳамма ерларга тарқалиш давомида кенгайиб борарди. Олимлар ва ўқитувчилар Web ни неча йиллардан бўён ишлатиб келганлар. Лекин, жамият унинг жозибадорлигига эндигина тушуниб ета бошлади. Жаҳон бўйлаб компьютердан фойдаланувчилар бутун дунё компьютер турига кириш фикри билан банд эдилар, лекин, қупинча олинган маълумотлар уларнинг умидин чиппаққа чиқарарди, чунки Интернет фақатгина матнли тарзда эди. Кейинчалик Webнинг илк ривожланиш даврида саҳифанинг дизайни ва белгилари, ишораларига кам эътибор берилган.

Қупгина воситалар ва дастурлар маҳсулотлари билан бирга мулжал олиш кийин бўлди. Ҳар бир янги ишлаб чиқувчи компания қандайдир янги восита киритишга ҳаракат қилди.

Бу эса Web ҳужжатни гоҳида ўз компьютерида гавдалантира олмаган фойдаланувчига номоён буларди. Web учун яратилган янги дастурлаш тилида Web ҳужжатларни қуздан кечириш ва ишлаш учун янги дастурий таъминот урнатишга доим зарурат мавжуд эди. Шунингдек улар олдига қуйилган айрим воситаларни аралаштиришга тугри келарди. Яхши ечимлардан бири PHP технологияси бўлиб чиқди.

PHP хакида умумий тушунча

PHP – бу сервер скриптларининг тили бўлиб (server scripting language), у серверда интерпритацияланадиган ва бажариладиган HTML га ўрнатилади. PHP HTML нинг препроцессори ҳисобланади, яъни унинг иши қуйидаги схема бўйича қурилган:



1 расм.

Сервер файлини браузерга “беришидан” олдин уни препроцессор-интерпретатор кўриб чиқади. Бу иш амалга ошиши учун препроцессор қайта ишлайдиган файллар маълум кенгайтмага (одатда бу .phtml ёки .php3 бўлади, аммо уларни ўзгартириш ҳам мумкин) ва препроцессор учун кодга эга бўлишлари (бу талаб албатта бўлиши шарт эмас) керак. Саҳифани жўнатишдан аввал PHP-код серверда чиқади ва натижа браузерга яна HTML – саҳифа кўринишда тақдим қилинади, бироқ бу саҳифа серверда сақланадиган саҳифадан кескин фарқ қилиши мумкин. .html/.htm кенгайтмали оддий саҳифаларни эса Web-сервер браузерга ҳеч қандай ишловсиз юборади.

Perl ёки C каби бошқа тилларда ёзилган CGI-скриптлардан фарқи – бу CGI-дастурларда чиқариладиган HTML-кодни ўзингиз ёзасиз, PHP ни қўллаш орқали эса сиз ўз скрипт-дастурингизни тайёр HTML-саҳифага очиладиган ва ёпиладиган теглар (мисол учун `<?php и ?>`) ёрдамида киритишингиз мумкин.

PHP ни, *мижоз скриптлари* тили ҳисобланадиган JavaScript/Jscript/VBScript дан фарқли равишда *сервер скриптлари* тили дейилади. Бу шуни англатадики, PHP-скрипт серверда бажарилади, мижозга эса унинг иши натижаси берилади, JavaScript-код эса бутунлигича мижоз машинасига берилади ва фақат ўша ерда браузер томонидан бажарилади.

MS Internet Information Server хаваскорлари PHP ни Active Server Pages (ASP) га жуда ўхшаш деб топадилар, Java энтузиастлари эса PHP ни Java Server Pages (JSP) га ўхшатишади. PHP нинг яна баъзи аналоглари бўлиб ColdFusion (www.allaire.com) ва embPerl ҳисобланади. Ушбу тилларнинг ҳаммаси Web- серверда бажариладиган кодни HTML-саҳифа ичига жойлаштиришга имкон беради. Дастурлашга хоҳиш ва билимингиз етарли бўлса, ўзингиз ҳам Web-саҳифага кодни, дейлик Фокал ёки Алгол тилида жойлаштиришга имкон берувчи бундай препроцессорни ёзишингиз мумкин. Энг осони, PHP ишини мисол орқали кўрсатишдир. Php элементли web-саҳифа қуйидаги кўринишга эга:

```
<html>
<head>
<title>Намуна</title>
</head>
<body>
<?php echo "Салом, мен PHP-дастурман!"; ?>
</body>
</html>
```

Ушбу скрипти бажарганимиздан кейин куйидаги сўзлар ёзилган саҳифачага эга бўламиз:

Салом, мен PHP-дастурман!

Ушбу саҳифанинг охириги матнини очиб куйидагиларни кўришимиз мумкин:

```
<html>
<head>
<title>Example</title>
</head>
<body>
Салом, мен PHP-дастурман!
</body>
</html>
```

Кўриб турганингиздек, натижавий саҳифада PHP-коддан асар ҳам йўқ. Бу жуда оддий ва фойдасиз кўриниши мумкин, аммо PHP янада мураккаб ва фантастик ишларни қилишга имкон беради, уларнинг айримлари билан кейинроқ танишиб ўтамиз.

PHP имкониятлари

Оддий қилиб айтганда, PHP да CGI–дастур ёрдамида қилиш мумкин бўлган барча ишларни бажариш мумкин. Масалан: формалардаги маълумотларни қайта ишлаш, динамик саҳифаларни генерациялаш, куки (cookies) ларни узатиш ва олиш. Бундан ташқари, PHP да кўпгина маълумот омборлари (databases) қўлланган, бу эса Web-иловаларни МО ёрдамида ёзишни жуда осонлаштиради.

Куйида қўлланадиган МО ларнинг айримларнинг рўйхати:

Adabas D InterBase Solid

dBase mSQL Sybase

Empress MySQL Velocis

FilePro Oracle Unix dbm

Informix PostgreSQL ODBC

Буларнинг барчасига қўшимча тарзда, PHP IMAP, SNMP, NNTP, POP3 протоколларини ва ҳатто HTTP ни ҳам тушунади, шунингдек, сокетлар (sockets) билан ишлаш ва бошқа протоколлар билан муомала қилиш имконига эга.

PHP нинг қисқача тарихи

PHP нинг бошланиши деб 1994 йилнинг кузини айтса бўлади, бу вақтда Rasmus Lerdorf ўз уй саҳифасининг имкониятларини кенгайтириш ҳамда оддий масалаларни бажариш учун кичикроқ движок ёзишга қарор қилди. Бундай движок 1995 йил бошига тайёр бўлди ва Personal Home Page Tools (шу ердан PHP қисқартмаси келиб чиққан) деб аталди. У кўп нарса қила олмасди – энг оддий тилни ва бор-йўғи бир неча макросларни тушунарди

1995 йилнинг ўрталарига бориб иккинчи версия пайдо бўлди, у PHP/FI Version 2 деб номланарди. FI қўшимчаси Rasmus нинг формаларни қайта ишлай оладиган бошқа пакети (Form Interpreter) дан қўшилди. PHP/FI Apache ичида компилирланар эди ва стандарт API Apache ни қўллади. PHP скриптлар мос CGI – скриптлардан тезроқ бўлиб чиқди, чунки серверга янги жараёни яратишга зарурият йўқ эди. PHP тили имкониятларига кўра CGI-дастурлар ёзиш учун энг машҳур тил – Perl га яқинлашди. Кўпгина таниқли маълумот омборларининг қўллаши қўшилди (Масалан, MySQL и Oracle). GD – кутубхона интерфейси gif-картинкаларни жуда тез генерациялаш имконини берди. Шу вақтдан бошлаб PHP/FI кенг тарқала бошланди.

1997 йил охирларида Zeev Suraski ва Andi Gutmans интерпретатор хатоларини тузатиш ва скриптлар бажарилиши тезлигини ошириш мақсадида ички движокни қайта ёзишга қарор қилишди. Ярим йилдан сўнг, 6 июн 1998 йилда PHP 3 номли янги версия чиқарилди.

1999 йил ёзида PHP 3 бир неча тижорат маҳсулотларига қўшилди.

NetCraft нинг 1999 йил ноябрдаги ахборотига кўра PHP 1млн. дан ортиқ доменларда қўлланилди.

2000 йилда PHP 4 янги версия тайёрланди, унда ички движок яна қайта ёзилди (унинг номи Zend - www.zend.com). Янги версиянинг самарадорлиги мавжуд версиядан ўнлаб мартаба юқори бўлади. Ушбу маҳсулотнинг RC1 ва RC2 тест версиялари чиқарилди.

Нима учун PHP ни танлаш керак?

Web-иловаларни яратувчилар web-саҳифаларнинг фақат матн ва расмдан иборат эмаслигини гапириб ўтиришларининг ҳожати йўқ. Эътиборга лойиқ сайт фойдаланувчи билан маълум даражада интерактивликни қўллаши керак: маълумот кидириш, маҳсулотларни сотиш, конференциялар ва б. Яқин пайтларгача буларнинг барчаси Perl да ёзилган CGI-скриптлар орқали анъанавий амалга оширилар эди. Аммо CGI-

скриптларнинг жуда ёмон масштабланиши маълум бўлди. CGI-скриптнинг ҳар бир янги чакируви ядродан янги жараёни яратилишини талаб қилади, бу эса процессор вақтини олади ва оператив хотирани сарфлайди. PHP бошқа вариантни таклиф этади – у Web-сервернинг бир қисми сифатида ишлайди ва унинг бу жиҳати от Microsoft нинг ASP сига ёки Allaire нинг ColdFusion га ўхшайди.

PHP синтаксиси C ёки Perl синтаксисига жуда ўхшаш. Дастурлаш билан таниш кишилар PHP да дастур ёзишни жуда тез бошлашлари мумкин. Бу тилда маълумотларни типизациялашга қаттиқ талаблар йўқ ва хотирадан жой ажратиш/бўшатиш бўйича ҳаракатларга ҳам зарурият йўқ.

PHP да ёзилган дастурлар етарлича осон ўқилади. Perl-дастурлардан фарқли равишда PHP-кодни кўриб ўқиш ва тушуниш осон.

Дастурнинг бепуллигига қўшимча тарзда (лекин MySQL ни тижорат мақсадларида қўллашда лицензия олиниши талаб қилинади) PHP-MySQL боғланмаси кросс-платформали ҳисобланади. Бу шуни англатадики, сиз Windows да ишлаб туриб, Unix да ишлаш учун мўлжалланган иловаларни ишлаб чиқишингиз мумкин. Бундан ташқари, PHP ташқи CGI-процесс сифатида ёки скриптларнинг оддий интерпретатори сифатида, Apache ёки IIS web- серверига уланган модул сифатида ҳам ишлаши мумкин.

Ва нихоят, берилган маҳсулот биргаликдаги уринишлар билан яратилар экан, бирор бир савол туғилганда мурожаат қилиш мумкин бўлган жуда кўп ҳужжатлар ва жўнатувлар рўйхатлари мавжуд. Топилган хатолар етарли даражада тез тузатилади, сизнинг таклиф ва эътирозларингиз доим эшитилади, кўриб чиқилади ва агар улар қимматли бўлса – янги версияда реализация қилинади.

PHP нинг камчиликлари

1. PHP 3 нинг асосий камчилиги шундаки, ўз идеологиясига кўра PHP аввалдан кичикроқ скриптларни ёзишга мўлжалланган эди. Движок бир неча маротаба қайта ёзилганлигига қарамай, PHP 3 ни мураккаб лойиҳаларда қўллаб бўлмайди – катта скриптларни қайта ишлашда тизим самарадорлиги кескин тушиб кетади (PHP 3 тарафдорлари учун айтиш мумкинки, бундай камчиликнинг мавжудлигини дастур яратувчиси Zeev Suraski нинг ўзи ҳам тасдиқлаган). Аммо бу камчилик PHP 4 да йўқотилади, у, дастур яратувчисининг фикрича, катта лойиҳаларда ишлаш учун мўлжалланган.
2. PHP интерпритацияланадиган тил бўлиб ҳисобланади, бунинг натижасида у компилирланадиган C билан тезликда тенглаша олмайди. Лекин PHP даги лойиҳалар учун характерли бўлган кичик дастурларни ёзишда, яъни лойиҳа бутунлигича кўпгина

кодли майда саҳифалардан таркиб топганда хотирани ишга туширишга қилинадиган кўшимча харажатлар ва С да ёзилган CGI-дастурга чакирувлар кучга киради.

3. PHP да масалан, Perl даги CPAN каби тайёр модулар базаси унчалик катта эмас. Бунда ҳеч қандай чора кўриб бўлмайди – бу вақт масаласи. PHP 4 да ишлаб чиқувчилар CPAN га ўхшаш махсус архивни кўзда тутганлар ва фикримизча, уни тўлдириш учун тез орада етарли миқдорда модулар ёзилади.

4. Масалан, ASP даги каби сессиялар (session) қўлланмайди. PHP 4 да бу камчилик йўқотилади.

PHP ни қаердан олиш мумкин?

PHP тарқатиладиган бош сайт бўлиб, www.php3.net ҳисобланади. У ерда сиз PHP нинг UNIX-платформаси учун ҳам, Win32 платформаси учун версияларини топишингиз мумкин.

PHP нинг тўртинчи версиясини ишлаб чиқувчилар сайти Zend кодли ном билан www.zend.com манзилида жойлашган.

PHP ни қандай ўрнатиш мумкин?

MySQL ўзи нима?

MySQL – кичик, компакт кўп оқимли маълумотлар омбори сервери.

MySQL юқори тезлик, ўзгармаслиги ва ишлатишда қулайлиги билан характерланади.

MySQL ТсХ компанияси томонидан ишлаб чиқилган, у жуда катта маълумотлар омборини тез қайта ишлаш эҳтиёжидан келиб чиқиб яратилган. Компаниянинг таъкидлашича, 1996 йилдан бери серверда MySQL дан 40 дан ортиқ МО лар фойдаланади, уларда 10000 жадваллар бўлиб, уларнинг 500 дан ортиғи 7 миллиондан кўп сатрга эга.

MySQL кичик ва ўрта иловалар учун идеал ечим бўлиб ҳисобланади. Сервернинг яқуний матнлари кўпгина платформаларда компилирланади. Сервер имкониятлари Unix-серверларида янада тўлароқ намоён бўлади, чунки бу ерда кўп оқимлилиқ қўлланилади, бу эса самарадорликни ошишига олиб келади. Windows учун ишлаб чиқилган вариантда MySQL Windows NT сервиси сифатида ёки Windows 95/98 даги оддий жараён каби ишга тушиши мумкин. Жорий вақтда MySQL, 3.22 версиялари бутунлай ишга яроқли бўлсада, ҳали ҳам яратилиш босқичида турибди.

MySQL-сервер нотижорат фойдаланиш учун бепул ҳисобланади. Акс ҳолда қиймати 190 EUR бўлган лицензияни олиш зарур.

MySQL имкониятлари

MySQL ANSI 92 стандартида SQL сўровлар тилини қўллайди, бундан ташқари ушбу стандартда, бошқа бирорта МОБТ да мавжуд бўлмаган қўплаб имкониятлар мавжуд. MySQL имкониятларнинг қисқа рўйхати.

1. Маълумотлар омбори билан бир вақтда чекланмаган сонли фойдаланувчилар ишлаши мумкин.
2. Жадваллардаги сатрлар сони 50 млн. га етиши мумкин.
3. Буйруқларнинг тез бажарилиши. MySQL мавжуд серверлар ичида энг тезкори бўлиши мумкин.
4. Оддий ва самарали хавфсизлик тизими.

MySQL ҳақиқатдан ҳам жуда тезкор сервер, аммо бунга эришиш учун яратувчилар реляцион МОБТ ларига қўйиладиган бир неча талаблардан воз кечишларига тўғри келди.

MySQL да қуйидагилар мавжуд эмас:

1. `SELECT * FROM table1 WHERE id IN (SELECT id FROM table2)` каби киритилган сўровлар қўлланиши. 3.23 версияда бундай имконият яратилиши кўзда тутилмоқда.
2. Транзакциялар ёрдами реализация қилинмаган. Унинг ўрнига `LOCK/UNLOCK TABLE` дан фойдаланиш таклиф қилинади.
3. Ташқи (foreign) калитлар ёрдами мавжуд эмас.
4. Триггерлар ва сақланадиган процедуралар ёрдами мавжуд эмас.
5. Намойишлар (VIEW) ёрдами йўқ. 3.23 версияда намойишлар яратиш имконияти режалаштирилмоқда.

Дастур яратувчилари фикрига кўра, айнан 2-4 пунктлар тез ҳаракат қилиш имконини берди. Уларнинг иши сервер тезлигини сезиларли пасайтиради. Уларнинг бўлиши Web-иловалар яратишда критик ҳисобланмайди, шунинг учун юқори тезлик ва кичик йўқотиш эвазига сервер катта шуҳратга эришди.

PHP дан фойдаланиш

Сизнинг илк PHP-саҳифангиз

Шуни мамнуният билан айтиш мумкинки, энг қийини ортда қолди. Дастурни ўрнатиш – бу доим мураккаб жараён, сабаби ҳеч бир тизим бошқасига ўхшамайди, ҳар бир ҳолатда олдин ҳеч ким тўқнашмаган муаммолар юзага келиши мумкин. Ҳар ҳолда маълумотлар омборингиз ўрнатилган ва киритилган, PHP-движок компилирланган, ўрнатилган ва Web-сервер билан боғланган, бу сервер .php3 кенгайтмали хужжатлар билан нима қилиш кераклигини билади.

Илк PHP-саҳифамизни ёзишга киришамиз. test.php3 нмли матнли файл яратамиз ва унда қуйидагиларни ёзамиз:

```
<html>
<body>
<?php
$myvar = "Hello, World";
echo $myvar;
?>
</body>
</html>
```

Энди браузерни очамиз ва унда яратилган саҳифанинг URL ни терамиз, масалан: <http://stage/test.php3>. Браузер экранда куйидагиларни кўришимиз керак:

Hello, World

Агар бунинг ўрнига сиз хатолар тўғрисидаги хабарларни кўрсангиз, ушбу китобнинг биринчи бўлимларига, агар талаб қилинса, ҳужжатларга ҳам мурожаат қилинг ва ҳаммасини тўғри қилинганлигини текширинг.

Саҳифанинг охириги матнини очиб, унда қуйидагини кўрасиз:

```
<html>
<body>
Hello, World
</body>
</html>
```

7 расм.

Бу PHP-движок серверда саҳифани кўриб чиқиб, унда PHP-кодни топиши, уни қайта ишлаши ва Web-сервер томонидан сизнинг браузерингизга жўнатилган натижани чиқарганлиги учун содир бўлди.

Табриклаймиз, Сиз ҳаётингиздаги илк PHP-саҳифани ёздингиз! Бу энди белгиланган матнли статик HTML-саҳифа эмас. Бу ҳақиқий дастур, унга юборилган маълумотларга боғлиқ ҳолда турли натижалар олиш мумкин. Бу имконият эса Интернетдаги ҳужжатлар нашрининг фалсафасини бутунлай ўзгартиради, уларни фойдаланувчи ҳаракатларининг ўзгаришига қараб статикдан динамик ҳолатга ўтказди. Биринчи саҳифа-дастуримизда бу имкониятлар яққол кўринмайди, чунки у қиладиган ягона иш – матнни чиқаришидир, бу матнни ўз қўлимиз билан тезроқ ёзишимиз мумкин. Шундай бўлсада, бир неча саҳифадан сўнг, сиз қўлингизда нақадар қувватли асбоб борлигини тушунасиз. Ҳозир эса биринчи PHP-саҳифамизни батафсилроқ кўриб чиқамиз.

Юқорида келтирилган кодда эътибор бериш керак бўлган биринчи нарса бу – чегаралашлардир. **<?php** дан бошланадиган сатрни топинг. PHP-движок учун бу код

ишланиши ва бажарилиши керак бўлган буйруқлар блокининг бошланишини билдиради. Блок `<?php` чекловчиси билан тугалланади. Бошқача қилиб айтганда `<?php` и `>` символлари кавслар ролини бажаради. Улардан ташқаридаги барча нарсани PHP-движок ўтказиб юборади ва Web га қайта ишламай, фақат кавслар ичидагини бажарган ҳолда узатади. PHP нинг кучи шундаки, PHP-кодни HTML-саҳифанинг исталган – таъкидлайман – исталган жойига қўйиш мумкин. Энди батафсилроқ кўриб чиқамиз. `<?php ... ?>` кавслари ўрнига `<? ... ?>` қисқартирилган нотацияни қўйиш мумкин.

Шунингдек ASP билан ишловчи айрим дастурчиларга кавсларни `<%... %>` комбинация билан ёзиш қулайроқдир. Агар хоҳласангиз, PHP ни шундай кавсларни ишлатишга созлашингиз мумкин. Кавслар вариантларидан яна бири қуйида келтирилган:

```
<SCRIPT LANGUAGE="PHP">
```

йўриқномалар

```
<SCRIPT>
```

Сиз эътибор қаратган яна бир нарса – бу коднинг ҳар бир сатри охиридаги нуқтали вергул. Бу “ажратувчилар” деб аталади ва бир буйруқлар тўпламини бошқасидан ажратиб туриш учун хизмат қилади. Умуман олганда PHP-коднинг ҳаммасини бир сатрда, буйруқларни нуқтали вергул билан ажратиб ёзиш мумкин. Лекин бундай кодни ўқиш қийинчилик туғдиради, шунинг учун бизнинг мисолларда ҳар бир нуқтали вергулдан кейин сатрни кўчириш белгиси қўйилди, шунингдек буйруқлар гуруҳини аниқроқ ажратиш учун яна битта сатрни кўчириш белгиси қўйилди. Сатр охирида нуқтали вергул ҳақида унутманг, бу бошловчи дастурчиларнинг энг тез такрорлайдиган хатоларидир.

PHP тилининг ҳақиқий кучи унинг функцияларида билинади. Назарий жиҳатдан, функция – бу бирор бир операцияни бажарувчи буйруқлар блоки ҳисобланади. Агар PHP ни унга тегишли барча қўшимчалар билан мужассамлаштирилса, 700 дан кўпроқ функцияларига кира олишингиз мумкин. Шундай экан, ўзингизнинг функцияларингизни фақат камдан кам ҳолатда ёзишингиз мумкин. Ҳар бир функция ўз номи ва синтаксисига эга. Биринчи мисолимизда **echo** функция ишлатилди, у кавс ичидаги сатрни ёки орқасидаги ўзгарувчини чиқаради.

Яна бир марта якуний PHP-саҳифани диққат билан кўриб чиқамиз. Умуман олганда, у оддий HTML-саҳифадан фарқ қилмайди. Фақат .html (ёки .htm) кенгайтмаси ўрнига .php3 кенгайтмасини ўзгартирдик. Web-сервер учун бу кенгайтма сигнал вазифасини ўтади, яъни ушбу саҳифани жўнатишдан олдин PHP-движокдан ўтказиш кераклигини эслатади. `<html><body> .... </body></html>` сатрлари PHP-движок томонидан қабул қилинмади. У фақат `<?php ... ?>` кавслари ичида ёзилганларга эътибор қаратади.

Натижада биз 2 расмда кўрсатилган тасвирга эга бўламиз. HTML-саҳифани бутунлай PHP-буйруқлар ёрдамида генерациялаш мумкин, масалан:

```
<?php  
echo "<html>";  
echo "<body>";  
$myvar = "Hello World";  
echo $myvar;  
echo "</body>";  
echo "</html>";
```

Натижа бир хил бўлди. Аммо дастурлаш нуқтаи назаридан, бу код – сифатсиз. Чунки PHP-движокка олдинги икки сатр код ўрнига олти сатр кодни қайта ишлашга тўғри келади. PHP-движок ишини <html>, <body>, </body>, </html> тегларини чиқариш билан қийинлаштиришнинг кераги йўқ. Код ёзаётганингизда ҳеч қачон самарадорлик ҳақида унутманг. Ҳар доим кодни ўзгартириш ёки яхшилашга шундай ҳаракат қилинги, PHP-движок уни қайта ишлашга иложи борица кам вақт сарфласин.

PHP Супер глобаллари

\$GLOBALS

Берилган скрипт кўриниши глобал соҳасининг шу онда мумкин бўлган ҳар бир ўзгарувчисига мурожаат мавжуд. Ушбу массивнинг калитлари бўлиб глобал ўзгарувчиларнинг номлари ҳисобланади.

\$_SERVER

Жорий скриптнинг web-сервер томонидан ўрнатилган ёки бошқа бирор йўл билан бажариш муҳитига алоқадор ўзгарувчилар. \$HTTP_SERVER_VARS эски массивга ўхшаш (унга ҳали ҳам кириш мавжуд, лекин тавсия қилинмайди).

\$_GET

Скриптга HTTP GET орқали бериладиган ўзгарувчилар. \$HTTP_GET_VARS эски массивга ўхшаш (унга ҳали ҳам кириш мавжуд, лекин тавсия қилинмайди).

\$_POST

Скриптга HTTP POST орқали бериладиган ўзгарувчилар. \$HTTP_POST_VARS эски массивга ўхшаш (унга ҳали ҳам кириш мавжуд, лекин тавсия қилинмайди).

\$_COOKIE

Скриптга HTTP cookies орқали бериладиган ўзгарувчилар. \$HTTP_COOKIE_VARS эски массивига ўхшаш (унга ҳали ҳам кириш мавжуд, лекин тавсия қилинмайди).

\$_FILES

Скриптга HTTP post – файлларни юклаш орқали бериладиган ўзгарувчилар. \$HTTP_POST_FILES эски массивига ўхшаш (унга ҳали ҳам кириш мавжуд, лекин тавсия қилинмайди). Қўшимча қаранг: POST усулида юклаш.

\$_ENV

Скриптга атроф муҳит орқали бериладиган ўзгарувчилар. \$HTTP_ENV_VARS эски массивига ўхшаш (унга ҳали ҳам кириш мавжуд, лекин тавсия қилинмайди).

\$_REQUEST

Скриптга фойдаланувчиларнинг киритиш механизмлари орқали бериладиган ўзгарувчилар, уларга ишониб бўлмайди.

Изох: буйруқ сатрдан ишга тушганда, бу массив argv и argc киришларини ёқмайди; улар \$_SERVER массивида тақдим этилган. Ушбу массивга ўзгарувчиларни ёқиш тартиби ва мавжудлиги variables_order конфигурациясининг директивасига мос ҳолда аниқланади. Бу массив PHP нинг 4.1.0. гача бўлган версияларида тўғридан тўғри аналогларига эга эмас.

\$_SESSION

Скриптнинг шу ондаги сессиясида регистрация қилинган ўзгарувчилари. \$HTTP_SESSION_VARS эски массивига ўхшаш (унга ҳали ҳам кириш мавжуд, лекин тавсия қилинмайди).

Арифметик операциялар

Мактабдаги арифметика асосларини эслайсизми? Бу операциялар ҳам худди шундай ишлайди.

10-2 жадвал. Арифметик операциялар

Мисол	Номланиши	Натижа
$a + b$	Қўшиш	a ва b . Суммаси
$a - b$	Айириш	a ва b . Фарқи
$a * b$	Кўпайтириш	a ва b . ни кўпайтириш
a / b	Бўлиш	a ни b . га бўлишдан қисмли
$a \% b$	Modulus	a ни b . га бўлишдан қолган бутун сонли қолдик

Бўлиш операцияси ("/") сонни доим сузиб юрувчи вергул билан қайтаради, ҳатто агар операндалар бутун сонлар бўлсада (ёки бутун сонга конвертирланадиган сатрлар).

```
$a = 3;
$a += 5; // $a да 8 ўрнатади, агар биз шундай десак: $a = $a + 5;
$b = "Hello ";
$b .= "There!"; // $b да "Hello There!" ни ўрнатади, $b = $b га ўхшаш.
"There!";
```

Битли операциялар

Битли операциялар бутун сонли қийматлар спецификацияланган битларининг қийматини ўрнатиш имконини беради. Агар чап ва ўнг параметрлар сатр ҳисобланса, битли операция сатр символлари устидан бажарилади.

```
<?php
echo 12 ^ 9;    // '5' беради

echo "12" ^ "9"; // Backspace (ascii 8) символини беради
                // ('1' (ascii 49)) ^ ('9' (ascii 57)) = #8

echo "hallo" ^ "hello"; // ascii-қийматларни беради #0 #4 #0 #0 #0
                // 'a' ^ 'e' = #4

?>
```

10-3 жадвал. Битли операциялар

Мисол	Номланиши	Натижа
$\$a \& \b	And	Ҳам $\$a$ да, ҳам $\$b$. да ўрнатиш билан битлар ўрнатилади
$\$a \b	Or	$\$a$ да ёки $\$b$. да ўрнатиш билан битлар ўрнатилади
$\$a \wedge \b	Xor	$\$a$ да ёки $\$b$. да, лекин фақат биттасида ўрнатиш билан битлар ўрнатилади
$\sim \$a$	Not	$\$a$ да ўрнатиш билан битлар ўрнатилади ва тескариси.
$\$a \ll \b	Чапга силжитиш	$\$a$ ўзгарувчиси битларни $\$b$ қадам чапга силжитади (ҳар бир қадам/жойлашиш “2 га кўпайтириш” маъносини англатади).
$\$a \gg$	Ўнгга	$\$a$ ўзгарувчиси битларни $\$b$ қадам ўнгга силжитади (ҳар бир

\$b	силжитиш	кадам/жойлашиш “2 га бўлиш” маъносини англатади).
-----	----------	---

Таққослаш операциялари

Ўз номидан маълумки, таққослаш операциялари икки қийматни солиштиради.

10-4 жадвал. Таққослаш операциялари

Мисол	Номланиши	Натижа
$\$a == \b	тенг	TRUE , агар $\$a$ $\$b$ га тенг бўлса.
$\$a == \b	мос	TRUE , агар $\$a$ $\$b$ га тенг ва улар бир типда бўлса. (фақат PHP 4 да)
$\$a != \b	тенг эмас	TRUE , агар $\$a$ $\$b$ га тенг бўлмаса
$\$a < \b	тенг эмас	TRUE , агар $\$a$ $\$b$ га тенг бўлмаса
$\$a != \b	мос эмас	TRUE , агар $\$a$ $\$b$ га тенг бўлмасаёки улар турли типда бўлса. (фақат PHP 4 да)
$\$a < \b	кичикроқ	TRUE , агар $\$a$ $\$b$ дан жуда кичик бўлса.
$\$a > \b	каттароқ	TRUE , агар $\$a$ $\$b$ дан жуда катта бўлса.
$\$a <= \b	кичик ёки тенг	TRUE , агар $\$a$ $\$b$ дан кичик ёки тенг.
$\$a >= \b	катта ёки тенг	TRUE , агар $\$a$ $\$b$ дан катта ёки тенг.

Шартли операция бўлиб, "?:" (тернар) ҳисобланади, у C ва бошқа кўпгина тилларда бажарилади.

```
(expr1) ? (expr2) : (expr3);
```

Бу ифода $expr2$ да ҳисобланади, агар $expr1$ **TRUE** да ҳисобланса, ва $expr3$ да, агар $expr1$ **FALSE** да ҳисобланса.

Хатоларни бошқариш операциялари

PHP хатоларни бошқаришнинг битта операциясини қўллайди: (@) белгиси. Агар у PHP ифодасининг префикси сифатида қўйилган бўлса, у ҳолда ушбу ифода билан генерацияланадиган барча хатолар ёпилади.

Агар `track_errors` ёқилган бўлса, ушбу ифода билан генерацияланадиган хатолар тўғрисидаги ҳар бир хабар `$php_errormsg` глобал ўзгарувчисига сақланади. Бу

Ўзгарувчи ҳар бир янги хато пайдо бўлиши билан қайта ёзилади, шунинг учун агар зарур бўлса, уни дарҳол текшириб туринг.

```
<?php
/* Файлли хато бўлиши мумкин */
$my_file = @file ('non_existent_file') or
    die ("Failed opening file: error was '$php_errormsg'");
// фақат функцияларда эмас, барча ифодалар учун ишлайди:
$value = @$cache[$key];
// агар $key индекси мавжуд бўлмаса, билдириш бўлмайди.
?>
```

Изоҳ: @-операцияси фақат ифодаларда қўлланилади. Асосий қоида: агар сиз бирор нарсанинг қийматини олсангиз, префикс сифатида @ операциясини қўйишингиз мумкин. Масалан, сиз уни ўзгарувчилар, функциялар ва **include()** чакирувлар, константалар ва бошқаларга қўйишингиз мумкин. Сиз уни функциялар ёки синфлар ёки if ва foreach ва б. каби бошқарув тузилмаларини аниқлаш учун қўя олмайсиз.

Бажариш операциялари

PHP битта бажариш операциясини қўллайди: (``) тескари апострофлари. Бу бир томонли кавслар эмаслигига эътибор қаратинг! PHP тескари апострофлар ичидаги маълумотларни қобиқ буйруғи сингари бажаришга ҳаракат қилади: чиқариш қайтарилади (яъни у шунчаки чиқишда берилмайди, у ўзгарувчи томонидан ўзлаштирилиши мумкин).

```
$output = `ls -al`;
echo "<pre>$output</pre>";
```

Изоҳ: Агар safe mode хавфсизлик режими ёқилган ёки агар **shell_exec()** ўчирилган бўлса, (``) операцияси ишламайди.

Инкремент/декремент операциялари

PHP C стилидаги pre- ва post-инкремент ва декремент операцияларини қўллайди.

10-5 жадвал. Инкремент/декремент операциялари

Мисол	Номланиши	Эффект
++\$a	Pre-increment	\$a ни 1 га катталаштиради, сўнг \$a ни қайтаради.
\$a++	Post-increment	\$a ни қайтаради, сўнг \$a ни 1 га катталаштиради.
--\$a	Pre-decrement	\$a ни 1 га камайтиради, сўнг \$a ни қайтаради.

\$a--	Post-decrement	\$a ни қайтаради, сўнг \$a ни 1 га камайтиради.
-------	----------------	---

Мантиқий операциялар

10-6 жадвал. Мантиқий операциялар

Мисол	Номи	Натижа
\$a and \$b	And	TRUE, агар \$a, ва \$b TRUE бўлса.
\$a or \$b	Or	TRUE, агар \$a ёки \$b TRUE бўлса.
\$a xor \$b	Xor	TRUE, агар \$a ёки \$b TRUE, лекин иккаласи эмас.
! \$a	Not	TRUE, агар \$a TRUE бўлмаса.
\$a && \$b	And	TRUE, агар \$a ҳам, \$b ҳам TRUE бўлса.
\$a \$b	Or	TRUE, агар \$a ёки \$b TRUE бўлса.

Сатрли операциялар

Икки хил сатрли операциялар мавжуд. Биринчиси - конкатенация операцияси ('.'), у ўнг ва чап аргументлардан ташкил топган бирлашмани қайтаради. Иккинчиси – ўзлаштириш операцияси ('.'=), у ўнг аргументни чап аргументга бирлаштиради.

```
$a = "Hello ";  
$b = $a . "World!"; // энди $b таркибида "Hello World!" мавжуд.  
$a = "Hello ";  
$a .= "World!"; // энди $a таркибида "Hello World!" мавжуд
```

XIX. dBase функциялари

Ўрнатиш

Бу функциялардан фойдаланиш учун сиз PHP ни --enable-dbase опцияси билан компилирлашингиз керак.

Конфигурация

Бу кенгайтириш конфигурациянинг ҳеч қандай директиваларини аниқламайди.

Ресурслар типлари

Бу кенгайтириш ресурсларнинг ҳеч қандай типларини аниқламайди.

Аниқланган константалар

Бу кенгайтириш ҳеч қандай константаларни аниқламайди.

Хулоса.

Мен “Тизимли дастурлаш тиллари” фанидан PHP дастуридан фойдаланиб фойдаланувчиларни руйхатдан утказиш мавзусида курс ишимни ёздим.

Ушбу курс ишини ёзиш давомида кўпгина маълумотлар билан танишдим. Шулардан ўзлаштириш оператори хусусан унинг ҳолатлари, ўтиш операторлари, модуллар, функциялар ҳақида ҳамда ҳар бир операторни танлашда қандай кўрсаткичларга аҳамият бериш лозим; ҳар бир кўрсаткич қандай принципда ишлашини ўрганиб чиқдим.

Замонавий технологиялар жадаллик билан ривожланаётган кунда, дастурлаш тилларининг ҳам янгидан-янги турлари ишлаб чиқилмоқда. Шу билан бирга уларнинг ишлаш принциплари тобора мукамаллашиб, фойдаланувчи ишлаши учун қулай имкониятлар туғилмоқда. Янги технологияларни ўрганиш, уларни ишлатиш ва улар учун дастурлар тузиш ҳам замонамизнинг долзарб масалаларидан бирига айланди.

Ушбу курс ишимни бажара туриб Визуал дастурлаш тилида дастур ёзиш усулларини, уларнинг ёзилиш турларини, ривожланиш тарихини ва бир қанча шу дастур билан боғлиқ бўладиган ходисаларни ўрганиб чиқдим. Дастурларининг хилма-хиллиги мени хайратга солди. Ўзим учун янги бўлган дастурларининг турлари билан танишиб чиқдим.

Менинг фикримча маълумотларнинг компьютерга дастурларининг кейинги келажаги истиқболлидир. Яъни уларнинг ҳозиргига нисбатан ихчам ва фойдаланиш учун қулай турлари ишлаб чиқилади.

Барчасини хулоса қилиб айтадиган бўлсак, курс ишини бажараётим шундай тушунчага эга бўлдимки: дастурий воситалар билан биргаликда уларнинг тузилиши, уларнинг дастурий бошқариш усулларини ўрганиш, шу ўрганилган нарсаларни ҳаётга кенг жорий этишдадир деб ўйлайман.

Бу ўрганганларим келажакда ўзимга ҳаёт йўлимда ас қотади деган умиддаман.