

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS
TA'LIM VAZIRLIGI**

BUXORO DAVLAT UNIVERSITETI
Fizika –matematika fakulteti

“Axborot texnologiyalari” kafedrası

“Kompyuter grafikasi va web dizayn” fanidan

REFERAT

Mavzu: PHPda fayllar, session va cookie bilan ishlash.

Bajardi:

Zayniddinov E.

Qabul qildi:

Xayatov X.

Buxoro-2016

Mavzu: PHPda fayllar, session va cookie bilan ishlash.

1. PHP tarixi.
2. PHP tili asoslari php dasturlari .
3. Izohlar.
4. O'zgaruvchilar.
5. Tashqi o'zgaruvchilar.
6. Konstantalar.
7. Cookie texnologiyasini o'qish.
8. FTP texnologiyasi.
9. PHPda sessiyalar bilan ishlsh.

PHP tarixi.

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

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

PHP – web texnologiya tili. PHPni o'rganish uchun avval HTML va dasturlash tilidan habardor bo'lish talab qilinadi. HTML/CSS va JavaScript larni mukammal

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

PHP tili asoslari PHP dasturlari

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

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

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

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

Serverga brouzerning murojlat qilishida yordamida php-stsenariylari bajarilishini ko'rib chiqamiz. Avval brouzer .php kengaytmali sahifani so'raydi, so'ngra web-server dasturni PHP mashinadan o'tkazadi va natijani html-kod shaklida qaytaradi. Agar standart HTML sahifani olib, kengaytmasini .php ga o'zgartirilsa va PHP mashinadan o'tkazilsa, foydalanuvchiga o'zgartirmasdan qaytaradi. Bu faylga

PHP komandani qo'shish uchun, PHP komandalani maxsus teglar ichiga olish kerak. Bu teglarning 4 xil shakli mavjud bo'lib, ixtiyoriysidan foydalanish mumkin:

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

Biz XML yoki SGML uslubiga rioya qilamiz.

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

`<?`

```

if(5<3){

echo("<p>Hello, world!<p>");

?>

<p>Hello!</p>

// bu qator PHP kodi sifatida qaralmaydi

// va kod bloki bajarilayotgan bo'lsa chiqariladi

<?

    echo("<p>Hello, world!<p>");

}

?>

```

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

Izohlar

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

```

<php

echo("<p>Hello</p>"); // izox

echo("<p>Hello</p>"); # izox

/*

    bu ham izox

```

*/

?>

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

Masalan:

<php

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

?>

// bu izox brouzerda ko'rinadi.

<!-- HTML izoxi.

Bu izox HTML kodda ko'rinadi, brouzerda emas -->

Izoxlarni faqat operator oxiriga emas, quyidagicha joylash ham mumkin:

<?

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

```
echo strstr($a,"H");
```

```
// bu funktsiyani keyinchalik qarab chiqamiz
```

?>

O'zgaruvchilar

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

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

O'zgaruvchi unga qiymat berilganda initsializatsiya qilinadi va dastur bajarilguncha mavjud bo'ladi. Ya'ni web-sahifa xolida to so'rov tugamaguncha.

Tashqi o'zgaruvchilar

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

Tabiiyki **\$HTTP_ENV_VARS** massivi o'zgaruvchilari tizimga bog'liqdir (chunki ular aslida atrof muxit o'zgaruvchilaridir). Atrof muxit o'zgaruvchilari qiymatlarini sizni mashinangiz uchun env (Unix) yoki set (Windows) komandasi yordamida ko'rishingiz mumkin.

So'ngra PHP u GET-o'zgaruvchilarning guruxini yaratadi. Ular so'rov satrini taxlil qilishda yaratiladi. So'rov satri **\$QUERY_STRING** o'zgaruvchida saqlanadi va so'ralgan URL dagi "?" simvoldan keyingi informatsiyadan iborat. PHP so'rov satrini **&** simvollari bo'yicha aloxida elementlarga ajratadi, va har bir elementda "=" belgisini qidiradi. Agar "=" belgisi topilgan bo'lsa, tenglik chap tomonidagi simvollaridan iborat o'zgaruvchi yaratadi. Quyidagi formani ko'ramiz:

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

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

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

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

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

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

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

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

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

Agar sahifa so'rovi POST usuli yordamida bajarilsa, POST-o'zgaruvchilarning guruxi yaratilib, interpretatsiya qilinadi va **\$HTTP_POST_VARS** massivga joylashtiriladi.

Konstantalar

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

define(CONSTANT, value)

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

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



```

echo(CONSTANT1);

echo(CONSTANT2);

echo(CONSTANT3);

?>

```

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

```

<?

define(CONSTANT,"Hello");

if(defined("CONSTANT"))

{

    echo("<p>CONSTANT is defined</p>");

}

?>

```

Cookie texnologiyasini.

Cookie — klient kompyuterida saqlanuvchi va u har safar serverga murojaat qilayotganda web-serverga yuboriladigan matn satridir. Shu tariqa ma'lumotlar turli skriptlar aro formalar yoki URL adreslarsiz uzatilishi mumkin. Cookie qiymatini o‘rnatish uchun quyidagi sintaksisdan iborat bo‘lgan setcookie funksiyasidan foydalaniladi:

```

bool setcookie (string name [, string value [, int expire [, string
path [, string domain [, bool secure]]]])

```

Funksiya klient kompyuterida saqlanuvchi cookie larni tasnif etadi. Quyida ushbu funksiya parametrlarining tasnifi ketirilgan:

- name. Cookie ning nomi.
- value. Cookie ning qiymati.
- expire. Cookie ning amal qilish muddati. Agar uberilsa, u tugagach cookie o'chirib tashlanadi. Agar ko'rsatilmasa, cookie brauzer oynasi yopilgandan so'ng o'chirib tashlanadi

- path. Serverdagi cookie ga ruxsat etilgan adres.
- domain. cookie ga ruxsat etilgan domen.
- secure. HTTPS protokol orhali bog'lanishda cookie ning havfsizlik belgisi.

Standart holatda cookie HTTPS uchun ham HTTP dagi kabi ishlaydi.

Ko'rinib turibdiki, cookie HTTP-so'rovning qismi hisoblanadi va u brauzerga yuboriladi. Shu sabab HTML-kod formalashtirishdan oldin uning qiymati o'rnatilishi kerak. Bu shuni anglatadiki, setcookie Funksiyasining qiymati HTML-tegdan oldin va echo operatorigacha o'rnatiladi.

Agar bu qoida saqlanmasa setcookie funksiyasini chaqirish FALSE qiymat qaytaradi va bu cookie formirovka qilinishida xatolik bo'lganligini anglatadi. cookie ning to'g'ri formirovka qilinishi funksiyaning TRUE qiymat qaytarishiga olib keladi. Bu cookie klientga qabul qilindi degani emas albatta. Chunki brauzerni sozlashda cookie o'chirib tashlash yoki serverga jo'natmaslik parametrlari o'rnatilgan bo'lishi mumkin.

PHP imkoniyatlari

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

foydalaniladigan boshqa sohalar ham mavjud. Bu sohalarni biz uchta asosiy qismga bo'lamiz:

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

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

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

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

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

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

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

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

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

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

Dasturiy vositani sozlash va o'rnatish

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

Denver Distributivi

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

PHP dasturlash tilining asosiy tuzilishi

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

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

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

Yuqorida aytib o'tganimizdek, PHP dasturlash tili Ci va Perl dasturlash tiliga o'xshash. Biroq keltirilgan dastur Ci va Perl dasturlash tilidagi dasturdan ancha katta farq qiladi. Bu 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 Ci va Perl dasturlash tillaridan kamchiligi shuki, murakkab skriptlarni PHP dasturlash tili ancha sekin bajaradi.

PHP-skriptlar – bu serverda bajariladigan va qayta ishlanadigan dasturlardir. Bu skriptlarni JavaScript tipidagi skriptlar bilan taqqoslash mumkin emas, chunki JavaScript tilidagi skriptlarda yozilgan buyruqlar faqat 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 hildimi yoki skript natijasini ko'rdimi buni bila olmaydi. Agarda skript mijoz kompyuterida qayta ishlansa (masalan, JavaScript tilidagi dastur), u holda mijoz skript kodidan iborat HTML-sahifani ko'radi.

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

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

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

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

Bu keltirilgan usullardan birinchisi har doim ham bajarilavermaydi. Undan foydalanish uchun qisqa teglarni ishlatish kerak, yoki PHP3 uchun short_tags() funksiyani ishlatish kerak, yoki PHP tilining konfiguratsion fayliga short_open_tag buyruqni o'rnatish kerak, yoki PHP dasturlash tilida enable-short-tags parametr bilan kompilyatsiya qilish kerak. Agarda php.ini-dist buyruqqa Yuqoridagilar avtomatik qo'shilgan bo'lsa, u holda qisqa teglardan foydalanish tavsiya etilmaydi. Ikkinchi usul xuddi o'rniga qo'yishga o'xshaydi, masalan, JavaScript kodlari va uning uchun mos html teglar. Shuning uchun undan har doim foydalanish mumkin, lekin bu noqulayligi uchun kamdan-kam ishlatiladi. Uchinchi usuldan faqat ASP texnologiyasidagi teglar asp_tags konfiguratsiyasida ishlatilgandagina foydalaniladi.

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. Yana shuni aytish kerakki, PHP-fayl CGI-skriptga o'xshamaydi. PHP-fayl bajarilishi shart emas, yoki yana qandaydir belgilanadi.

PHP-faylni serverda qayta ishlash uchun jo'natishda server tomonidan brauzer satrida bu faylni yo'lini ko'rsatish shart. PHP skriptlar www orqali kirishga ruxsat

etilgan joyda joylashishi shart. Agarda PHP-fayl lokal kompyuterda mavjud bo'lsa, u holda uni buyruqlar satri interpretatori yordamida qayta ishlash mumkin.

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

Cookie texnologiyasini o'qish

Cookie qiymati o'rnatilganidan so'ng, sahifa qayta yuklanmagunicha darhol undagi skript uchun cookie ga ruxsat mavjud bo'lmaydi. Chunki cookie foydalanuvchi kompyuterida saqlanadi va brauzer orhali web-serverga jo'natiladi. Bundan tashqari, cookie qachonki uning domeni server domeni bilan mos tushgandagina jo'natiladi.

Cookie ga ruxsatni olish uchun maxsus `$_COOKIE` superglobal massividan foydalaniladi. Massivning qiymati sifatida oldin ishlatilgan cookie nomi olinadi. Massiv odatda skript yuklanayotgan vaqtda `$_GET`, `$_POST` va `$_REQUEST` massivlari bilan avtomatik tarzda to'ldiriladi

Cookie dan foydalanishdan oldin uning qiymati o'rnatilganligiga ishonch hosil qiling. Buning uchun `isset` funksiyasidan foydalanish juda qulay. Quyidagi misolda message nomli cookie ning qiymati tekshirilishi va ko'rsatilishi keltirilgan.

Misol:

<HTML>

<HEAD> <TITLE> Cookie ning qiymatini o'qish <\TTITLE> <\HEAD>

<BODY>

<CENTER>

<N1> Cookie ning qiymatini o'qish <\H1>

Cookie ning qiymati:

```
<?php
```

```
if (isset ($_COOKIE ['message']))
```

```
{
```

```
echo Cookie ning qiymati:' . $_COOKIE ['message' ] ;
```

```
}
```

```
else
```

```
{
```

```
echo 'Cookie o'rnatilmagan' ;
```

```
}
```

```
?>
```

```
<\CENTER;.
```

```
<BODY>
```

```
<\HTML>
```

Cookie massivlarda ham tashkil etilgan bo'lishi mumkin. Masalan quyida uchta cookie o'rnatilgan:

```
setcookie ("cookie[one]" , "Bugun");
```

```
setcookie ("cookie[two]" , "Hayot");
```

```
setcookie ("cookie[three]" , "go'zal!" ) ;
```

Natijada \$_COOKIE['cookie'] massiv qiymatlari quyidagi tarzda bosmaga chiqarilishi mumkin:

```
if (isset ($_COOKIE ['cookie'])) {
```

```
foreach ($_COOKIE['cookie'] as $data)
```

```
{
```

```
echo "$data <BR>";
```

```
...
```

FTP texnologiyasi

Quyida FTP ning ba'zi atamalarini ko'rib chiqamiz:

1. FTP

File Transfer Protocol (fayllarni almashish protokoli); tarmoq orqali (xususan Internet) fayllarni uzatish protokoli. Buni bir kompyuterdan boshkasiga fayl nusxasini ko'chirish kabi tushunish mumkin.

2. FTP Server

Bu faylni jo'natilishi so'rovini kutuvchi kompyuter yoki server.

3. FTP Client

Bu FTP serverga so'rov jo'natuvchi kompyuter. So'rov tekshiruvi yoki tasdiqlanishidan so'ng FTP klient kompyuteri ma'lumotlarni serverga yuklashi yoki serverdan yuklab olishi mumkin.

4. Anonim FTP

Ushbu serverga FTP klient kompyuteri orhali avtorizatsiyadan o'tmasdan bog'lanishi mumkin. Bunday imkoniyatni bir nechta web saytlarda ko'rishimiz mumkin, qaysiki ro'yxatdan o'tmasdan ma'lum fayllarni yuklab olish imkoniyatini beradi.

5. FTP Host

Ushbu xizmat FTP sayt sifatida, ya'ni saytda ro'yxatdan o'tib, saytda keltirilgan ma'lumotlar fayllarini yuklab olish imkoniyatini beruvchi kompyuter. Bu foydalanuvchilar soni cheklangan pullik xizmat hisoblanadi.

6. FTP sayt

FTP xost kompyuteri tasarrufidagi foydalanuvchi login paroli talab qilinuvchi web sahifa. Xost serverda bir nechta web-sahifa joylanishi mumkin, bunda har bir sayt uchun alohida foydalanuvchi avtorizatsiyasi mavjud.

7. FTP Proxy

FTP Proxy bu shunaqa server kompyuterki, bunda so'rov FTP serverga jo'natiladi, ushbu so'rov proksi orhali o'tib, keyin zarur manzilga yo'naltiriladi.

E-mail texnologiyasi

Bu xizmat internet tizimi orhali o'zaro pochta xabarlar almashinuvi usulidir. Bunday tizim oraliq saqlanish metodi asosida yo'lga qo'yilgan. Zamonaviy internet protokollar xabarlar jo'natish ikoniyati bo'yicha Yuqori darajaga ko'tarildi. Pochta jo'natishning eng keng tarqalgan protokoli SMTP (Simple Mail Transport Protocol) hisoblanadi.

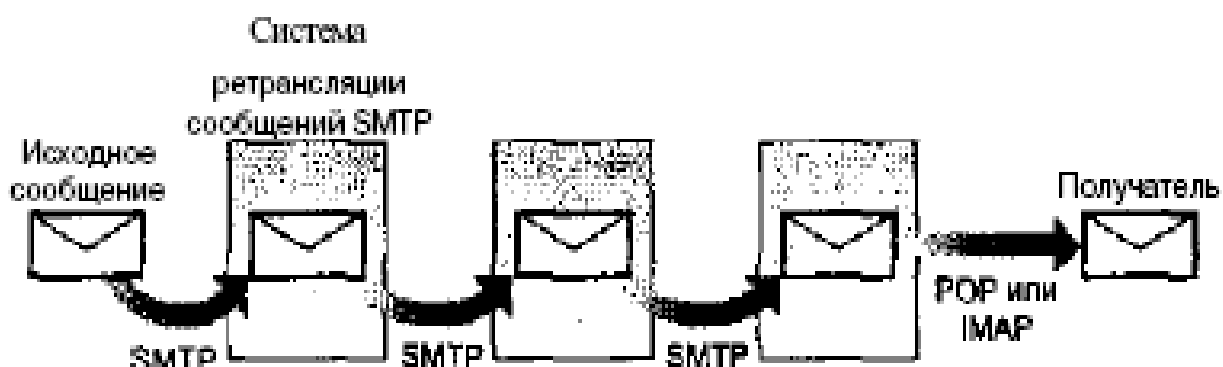


Рис. 22.2 Для пересылки и получения почты используются различные протоколы

Pochtani qabul qilish pochta protokoli POP (Post Office Protocol), yoki Internet xabarlarini qabul qiluvchi protokol IMAP (Internet Message Access Protocol) dan foydalaniladi.

PHPda sessiyalar bilan ishlash.

Sessiyalardan foydalanish uchun dastlab session_start() funksiyasini chaqirish lozim. Bu funksiya har qanday HTML tegdan oldin yozilishi shart. Aks holda xatolik beradi. Sessiyada cookie dan farqli ravishda fayllar serverda joylashadi.

Sessiyaga qiymat berish \$_session global o'zgaruvchisi orqali amalga oshiriladi.

Sessiyadan foydalanishga quyidagicha misol keltiramiz. 4ta fayl hosil qilamiz.

- 1) Session1.php faylida color va animal nomli o'zgaruvchi hosil qilamiz.
- 2) Session2.php faylida color va animal o'zgaruvchisi qiymatlarini chiqaramiz.
- 3) Session3.php faylida o'zgaruvchilar qiymatini o'zgartiramiz.
- 4) Session4.php faylida sessiyani o'chiramiz.
- 5) pages.php faylida sahifalarga havolalar o'rnatamiz.

Session1.php faylida color va animal nomli o'zgaruvchi hosil qilamiz.

```
<?php
//sessiyani boshlash
Session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
//sessiyaga qiymat berish
$_SESSION["color"] = "green";
$_SESSION["animal"] = "cat";
Echo "Sessiya ma'lumotlari o'rnatildi";
Include "pages.php"
?>
</body>
</html>
```

Session2.php faylida color va animal o'zgaruvchisi qiymatlarini chiqaramiz.

```

<?php
Session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
Echo "Favorite color is"._SESSION["color"].".<br>;
Echo "Favorite animal is"._SESSION["animal"].".<br>;
Print_r($_SESSION);
Include "pages.php"
?>
</body>
</html>

Session3.php faylida o'zgaruvchilar qiymatini o'zgartiramiz.
<?php
Session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// sessiya ma'lumotlarini o'zgartirish
$_SESSION["color"] = "yellow";
$_SESSION["animal"] = "dog";

Echo "Favorite color is"._SESSION["color"].".<br>;
Echo "Favorite animal is"._SESSION["animal"].".<br>;
Print_r($_SESSION);
Include "pages.php"

```

```
?>
</body>
</html>
Session4.php faylida sessiyani o'chiramiz.
<?php
Session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// sessiyaning barcha o'zgaruvchilarni o'chirish
Session_unset();
// sessiyani tugatish
Session_destroy();
Include "pages.php"
?>
</body>
</html>
```