

PHP Davrasida

Bu kitobimni men **sevgili singlim Nozima** uchun bahshida etaman. Har doim meni qo'llab quvatlagan, har doim meni tushinib yordam berganligi uchun tashakkur etaman.

Kitobimni yozishda menga ko'mak bergan <http://www.technoidea.boom.ru> sitesiga va <http://www.uzinfo.info/> sitesiga o'z minatdorchiligimni bildirib o'taman.

Bu kitob Linux sistemasida Open Office.org programmasida yozilgandir. GNU GPL litsensiyasi ostida tarqatiladi.

Savol va takliflarni uzinfo@inbox.ru / kitob@uzinfo.info da kutib qolaman!!!

2 – Bosma (Ba'zi bir qoshimchalar kiritilgan)
Asosan PHP4 bo'yichadir.

Bu kitobni tijorat maqsadida ishlatish qat'iyan man etiladi. O'zbekiston ichida tamoman bepul tarqatiladi. Mening yani yozuvchining ruhsatisiz chop etish, sotish qat'iyan man etiladi

PHP Davrasida kitobi Open Source

Assalomu aleykum Azizlar,

Mana bugun biz Web Sahifalar yaratish darslarimizning keyingi qismi yani Ma'lumotolar Ba'zasi bilan ishlovchi va boshqa HTML komandalari bilan qilib bo'lmaydigan ishlarni bizga osonlikcha qilib beruvchi programmalash tili yani PHP ni o'rganishni boshlaymiz.

(Bu kitobim oldingisiga qo'shimcha qilingan holdadir, yani eski PHP davrasida kitobini yangilanganidir)

Agarda siz shu so'zlarni o'qiyotgan bo'lsangiz unda siz Web sahifa yaratishni boshlagansiz va keyingi pog'onasiga o'tib, PHP o'rganishni boshlamoqchisizdir.

Keling avalam bor PHP nimaligini, uni kelib chiqishini yozib o'tsak.

PHP tarihi:

PHP 1995 yili, Rasmus Lerdorf Perl/CGI da internet site'ga kirganlarning qanchaligini hisoblovchi kichik bir programmacha yozdi. O'sha vaqtda bundan ishlarni bajaruvchi hech qanday programma bo'lmagan, shuning uchun unga judayam ko'p hatlar kelib, uni qanday qilganligini so'rab surishtirishni boshlashgan. Shunda Lerdorf o'zining bu kichik programmasini yani PHP(Personal Home Page yoki Hypertext Processor) ni tekinga tarqatishni boshlagan.

Bundan tez tarqalib, juda ko'p qulanishni boshlanganini ko'rgan Lerdorf PHP ni ishini yani hususiyatlarini, qo'shimcha komandalarini kengaytirishni boshlaydi. Vaqti kelib Perl uning uchun kamliq qilib u bu programmani C programmalash tilida yozishni boshladi, va PHP'ni ishlab chiqarishda butun dunyo programistlari qatnasha boshlashdi.

1997 yili PHP 3.0 chiqdi, va butun dunyo bo'yicha 50 000 ortiq insonlar shu programmani ishlatishni boshlagan edi.

1999 yilga kelib esa PHP ishlatuvchilar soni 1 000 000 dan ortib ketdi.

Bu PHP tarihini kichik bir qismi edi.

PHP yoki HTML?

PHP HTML kodining ichida yozilishi ham mumkindir. Yani huddi HTML komandlari va ora sira qandaydir boshqa ko'dlar uchrashi mumkindir, mana bu kabi:

```
<html>
<title> <? print "Hello World"; ?></title>
</html>
```

Agarda siz shu sahifani index.php deb saqlab Internet Browser bilan qarasangiz sizning Sahifa boshi "Hello World" bo'ladi.

Lekin PHP sahifalarning ko'rsatishi uchun sizda Apache Server PHP module bilan yoki IIS PHP module bilan birga bo'lishi kerak.

Apache eng yahshi va tez ishlaydigani Linux yoki Unix sistemalariga qo'yilganidir, lekin agarda sizga Windows XP yoki NT yoqsa unda ularga ham qo'yib tekshirishingiz mumkindir.

PHP Davrasida kitobi Open Source

Apache serverni siz <http://www.apache.org> sahifasidan olishingiz mumkindir.

Agarda siz Ma'lumotlar bazasi bilan ishlamoqchi bo'lsangiz unda sizga MySQL ham kerak bo'ladi, uni siz <http://www.mysql.com> dan topishingiz mumkindir.

PHP module'larni esa siz <http://www.php.net> site'sidan olib o'rnatishingiz mumkindir.

Yuqoridagi programmalarni o'rnatgandan so'ng siz PHP sahifalari bilan ishlashni boshlashingiz mumkindir. Bu programmalarni o'rnatishni o'rgatish bu kitobda ko'rsatilmaydi, ularni hammasi boshqa kitobimizda yoritib o'tiladi va bu kitob sizga faqatgina PHP sahifalari bilan ishlashni, yaratishni bosh qismini yani kirish qisminigina o'rgatib o'tishni o'z zimmasiga oladi.

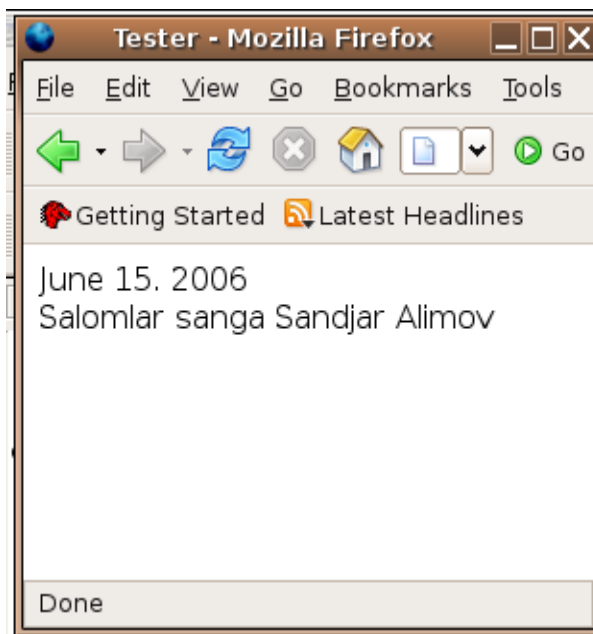
Agarda siz yuqoridagi programmalarni qo'yib bo'lgan bo'lsangiz unda keling kichkina php sahifasini yozamiz.

```
<?
$site_title="PHP kitobi";
$author="Sandjar Alimov";
?>
<html>
<head>
<title><? print $site_title; ?></title>
</head>

<body>
<?
print date("F d. Y");
print "<br> Salomlar sanga $author";
?>
</body>

</html>
```

va sizning oynangizga quyidagi chiqadi



O'zgaruvchilar va O'zgarmaslar

O'zgaruvchilar hamma programistlash tilining asosidir. Bularga O'zgarmas sonlar, belgilar va bshq. kiradi.

Misol qilib sonlarni olaylik. Ular butun sonlar, qoldikli sonlar bo'lsihi mumkindir va yana bularni orasiga ikkilik, sakizlik, o'n oltilik sanoq sistemasidagi sonlar ham kiradi.

Yoki belgilardan &%/\$# va yana harflar ham kiradi.

Agarda biz qandaydir o'zgaruvchanga nimadir belgilab berish niyatimiz bo'lsa unda biz quyidagicha yozishimiz shartdir:

```
$ovqat="manti";
```

```
$rang="qizil";
```

Bulardan tashqari yana ishchi begilarimiz bor yani:

<i>Nomi</i>	<i>Ta'rifi</i>
\n	Yangi qator
\r	kursorni qaytishi
\t	gorizontal tabulyatsiya
\\	Teskari qiyshiq chiziq
\\$	Dollar belgisi
\"	Qo'shtirnoq belgisi
\[0-7]{1,3}	Sakizlik sanoq sistemasini yozilish
\x[0-9A-Fa-f]{1,2}	O'n oltilik sanoq sistemasi yozilishi

Bu belgilarni kerakli bo'lgan joylarda qulaygina ishlatishingiz mumkindir.

Shu yerda o'rinli deb bildim "" belgisi bilan " belgilari orasida farq bordir yani:

Misol:

```
$double_list="item1\ntitem2\ntitem3";
```

```
$single_list='item1\ntitem2\ntitem3';
```

Ekranga chiqarganingizda single_list tamoman bir biri bilan birga va \n belgisi bilan chiqadi, double_list esa bu belgilersiz bo'sh joy bilan birga chiqadi.

Masivlar

Masiv: Bir qancha o'zgaruvchilarni birligidir.

yozilishi: \$ismi[raqami]

misol:

```
$gosht[0]="tovuq";
```

```
$gosht[1]="qo'y go'sht";
```

```
$gosht[2]="mol go'sht";
```

agarda siz print \$gosht[1] desangiz unda oynangizga “qo'y go'sht” chiqadi.

Yoki array() komandasini ishlatishingiz mumkin.

misol:

```
$ovqat=array(“manti”, “shurvo”, “kabob”);
```

Agarda sizga masivlarni so'z bilan ajratish qilay bo'lsa unda PHP buni ham qabul qiladi.

misol:

```
$maktab[“ismi”]=“Cholpon”;
```

```
$maktab[“raqami”]=“258”;
```

```
$maktab[“shahri”]=“Toshkent”;
```

Ichma ich massivlar ham yoza olasiz.

yozilishi: \$ismi[raqami1][raqami2]..[raqamiN];

misol:

```
$shahmat[0][0]=“Peshka”;
```

```
$shahmat[0][1]=“Tora”;
```

```
$shahmat[1][1]=“Ot”;
```

Obyektlar

Obyektlar PHPda ham ishlatiladi.

Obyekt: Bu o'zgaruvchan bo'lib, uni na'munasi oldindan aniqlangan shablon orqali yaratiladi, yani

class'lar orqali. Obyektlar OOP (Object Oriented Programming) ning asosiy qismini yani o'zagini tashkil etadi.

Misolda ko'raylik:

```
class appliance {  
var power;  
function set_power($on_off){  
$this->power=$on_off  
}  
}  
...  
$blender=new appliance;
```

Bu misolda biz class so'zi bilan appliance nomli obyektни belgilab oldik, so'ngra esa uning ichida power o'zgaruvchisi borligini ko'rsatib set_power funksiyasini ochdik uning ichida esa power o'zgaruvchisiga funksiya olgan qiymatni belgilab berdik.

So'ngra \$blender'ga biz shu obyektни hususiyatlari bilan birga belgiladik.

Mantiqiy funktsiyalar

TRUE, FALSE

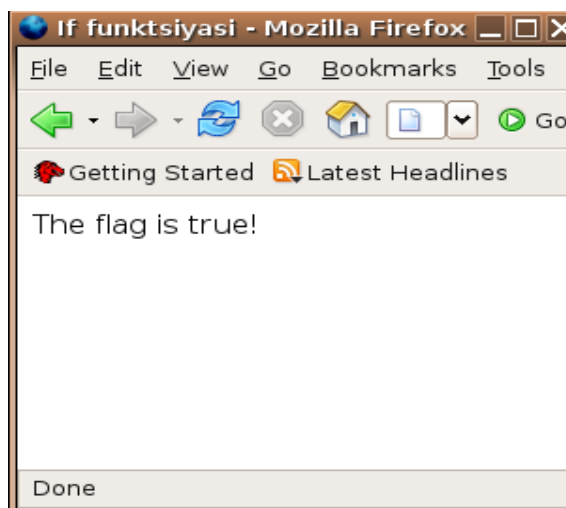
Solishtirish funktsiyalariga “if else” funktsiyasidir. Bunday solishtirishda sizda TRUE (Rost) yoki FALSE (yolg'on) chiqadi. keling misol bilan ko'rsak.

```
$flag=TRUE;  
if ($flag==TRUE):  
print “The flag is true!”;  
else:  
print “The flag is false!”;  
endif;
```

Agarda flag o'zgaruvchimiz rost bo'lsa unda ekranimizga “The flag is true!” yoziladi. Agarda yolg'on bo'lsa unda ekranimizga “The flag is false!” yoziladi.

Kelin tamoman yangi bir misol bilan ko'rib chiqsak. Bu faylni ifed.php nomi bilan saqlaymiz.

```
<?  
$flag=TRUE;  
?>  
<html>  
<head>  
<title>If funktsiyasi</title>  
</head>  
<body>  
<?  
if ($flag==TRUE):  
print “The flag is true!”;  
else:  
print “The flag is false!”;  
endif;  
?>  
</body>  
</html>
```



Mana rasmdan ko'rib turibsizki flag o'zgaruvchanga biz TRUE berganimizda bizning “The flag is true” yozuvimiz ekranga chiqadi. Huddi shunday agarda siz FALSE bersangiz unda ekraningizga “The flag is false” yozuvi chaqadi.

Shu yerda yozib o'tish o'rinli deb bildim. TRUE va FALSE yozuvlarini siz raqam bilan ham belgilab berishingiz mumkin. TRUE=1 FALSE=0.

Agarda yuqoridagi misolda \$flag=1; deganingizda ham oynangizga “The flag is true!” yozuvi chiqqan bo'lardi.

Nomlash

Obyektlarni, funktsiyalarni, o'zgaruvchanlarni nomlash uchun siz hamma belgilardan (lotincha)

ishlatishingiz mumkindir. Faqatgina boshlanishi son dan va yordamchi belgilardan bo'lishi kerak emasdir. Va ichida yordamchi belgilari bo'lmasligi kerak.

misol:

<i>Mumkin</i>	<i>Mumkin emas</i>
my_function	This&That
Size	!counter
somewords	4word
otherstring	!otherstring

O'zgaruvchanlarni tanitish

Yuqorida o'zgaruvchanlarga qandaydir so'zni, yoki gapni tanitish, yani berishni ko'rib chiqdik.

Lekin yana bir bor qaytarib o'tish yana ham ko'proq o'rhanishingizga, agarda tushunmagan bo'sangiz tushunib yetishingizga yordam beradi.

So'z yoki gaplarni quyidagicha qilib tanitamiz:

```
$sentence="Salom hayot!";
```

Sonlarni esa:

```
$price=43.22;
```

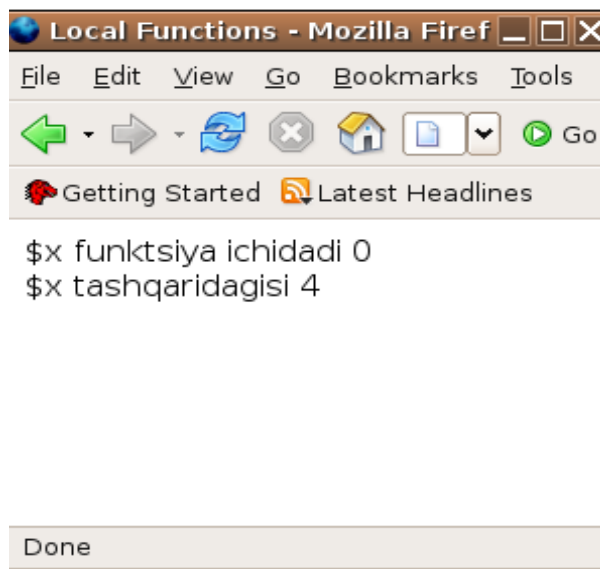
```
$ogirligi=185;
```

Ichki o'zgaruvchan

Ichki (local) o'zgaruvchan deb funktsiyaning ichida tanitilgan o'zgaruvchanga aytiladi.

misol:

```
<?
$x=4;
function assignx(){
    $x=0;
    print "$x funktsiya ichidagi $x";
}
assignx();
print "<BR> $x tashqaridagisi $x";
?>
```

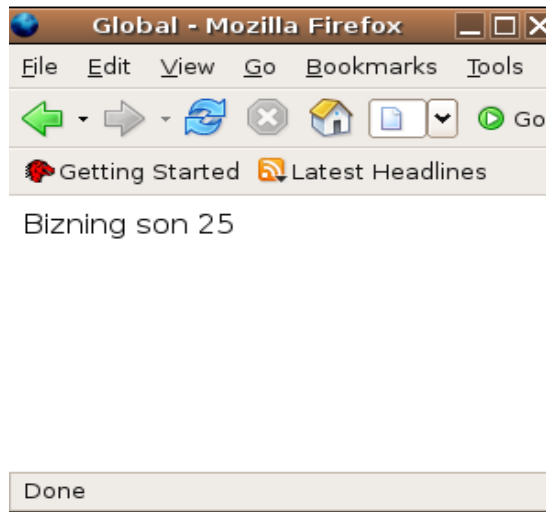


Shunda ekraningizda yuqoridagi rasmdagi kabi \$x ning ikki hil raqami chiqadi.4 bu umumiy tanitilgan raqami, 0 esa ichkisidir.

Global o'zgaruvchan

Global, yani umumiy o'zgaruvchanlar butun sahifamiz bo'yicha hamma joyda ishlatishingiz mumkindir. yuqoridagi misolda $x=4$ bu bizning global o'zgaruvchanimizdir. Agarda uni biz funktsiyamiz ichida ishlatmoqchi bo'lsak unda:

```
$sonlar==15;  
function arifmet(){  
GLOBAL $sonlar  
$sonlar=$sonlar+10;  
print "Bizning son $sonlar";  
}  
arifmet();
```



Agar shuni biz php sahifamizga qo'shib ishlatsak quyidagi ekranimizga chiqadi.

Static o'zgaruvchan

Static: Bu ichki o'zgaruvchan bo'lib, ammo funktsiyaning ichidan chiqqaningizdan so'nga ham o'z qiymatini saqlab qoladi, va shu funktsiyaning keyingi bor ishlatganingizda o'z qiymati bilan ishlaydi. Misolda ko'raylik:

```
function keep_track(){  
STATIC $count=0;  
$count=$count + 10;  
print $count;  
print "<BR>";  
}
```

```
keep_track();  
keep_track();  
keep_track();
```

Agarda siz shu kodni ishlatsangiz unda sizning ekraningizda 10 20 30 sonlari yoziladi. Ko'rib turganingizdek har gal yengittan ishlatganingizda count o'z qiymatini saqlab qoldi.

Bir turdan ikkinchi turga o'tish

Agarda siz qandaydir son hoh butun son hoh qoldiqli bir turdan ikkinchisiga o'tkazish niyating bo'lsa unda nima qilish kerak?

Keling men sizga avval qanday turlari borligini yozib so'ngra bir turdan ikkinchisiga o'tkazishni ko'rib chiqamiz.

<i>Turi</i>	<i>ta'rifi</i>
(int) yoki (integer)	butun son
(real), (double), (float)	qoldiqli sonlar
(string)	harf, belgilar
(array)	Massivlar
(object)	Obyektlar

Keling biridan ikkinchi truga o'tkazishni ko'raylik.

```
$son=12;
```

```
$son=(double)$son;
```

shunda 12 soningiz 12.0'ga aylanadi.

yoki yana bir boshqa misol

```
$son1=4.0;
```

```
$son2=5;
```

```
$son3=(int)$son1+$son2;
```

Shunda son3 9 soniga teng bo'ladi.

```
$son1=7.8;
```

```
$son2=6;
```

```
$son3=(int)$son1+$son2;
```

Sunda son3=13'ga

Agarda siz obyekt bermoqchi bo'lsangiz unda:

```
$model="Toyota";
```

```
$new=(object)$model;
```

siz buni ekranga chiqarish uchun scalar so'zidan foydalanishingiz kerak, yani:

```
print $new->scalar;
```

O'zgarmaslar

Oldindan aniqlab berilgan son yoki sozlar tarkibidir.

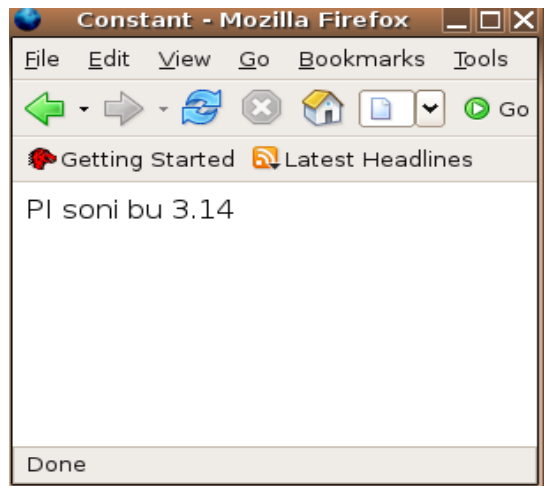
misol uchun agarda biz PI so'ziga 3.14 sonini bermoqchi bo'lsak va bu son hech ham o'zgarmasligini bilsak unda biz o'zgarmasni yozishimiz kerak (constant)

misol:

```
define("PI", "3.14");
```

```
print "PI soni bu";  
print PI;
```

buni biz sahifamizga qo'shsak unda oynamizda quyidagicha ma'lumot paydo bo'ladi.



Mana shunday qilib hamma O'zgaruvchanlar va o'zgarmaslarni va shu qatorda funktsiyalar, obyektlarni ko'rib chiqdik.

Shu yerda bazi bir funktsiyalarni ham ko'rib chiqaylik.

print() - Bu oynaga ma'lumot yuborishdir.

sprintf() - Bu qandaydir ma'lumotni formatlash(kerakli shaklga kiritish) uchun va uni qandaydir o'zgaruvchiga berish uchun ishlatiladi.

printf() - qandaydir ma'lumotni formatlab oynaga yuborish uchun ishlatiladi.

misol:

```
<?
```

```
$format=""The are %s monkeys in the $d";
```

```
printf($format, $num, $location);
```

```
?>
```

agarda biz \$num'ga 3 bersak va \$location'ga "tree" bersak unda quyidagi oynamizga chiqadi.

"The are 3 monkeys in the tree" bo'ladi.

yoki

```
<?
```

```
$money1=68.75;
```

```
$money2=54.35;
```

```
$money=$money1+$money2;
```

```
$formatted=sprintf("%01.2f", $money);
```

```
?>
```

shunda \$formatted=123.00 bo'ladi, agarda biz sprintf siz oynaga chiqarganimizda unda 123 chiqgan

bo'lardi.

Formatlashga kiritish turlari.

%-protsent belgisi.

b- son, ikkilik sanoq sistemasida (0 va 1)

c- son, songa mos keladigan ASCII belgilarini chiqaradi. (65=A belgisi)

d- son, belgili onlik sanoq sistemasi (+10 yoki -10)

u-son, belgilanmangan onlik sanoq sistemasi (10, 5)

f- qoldikli son, qoldikli son shaklida (1.25, 20.36)

o- son, sakizlik sanoq sistemasida (23, 65)

s- belgi, yani harflar yoki boshqa belgilar

x- son, o'n oltilik sanoq sistemasi (harflari kichkinada yoziladi) (12ab, d23c)

X- son, o'n oltilik sanoq sistemasi (harflari kattada yoziladi) (12AB, D23C)

Mana shular bilan siz foydalanib ma'lumotingizni formatlashingiz mumkindir.

Operandlar

<i>Operatorlar</i>	<i>Ta'rifi</i>
()	Qavs
new	Yangi, obyekt hosil qilish
! ~	Mantiqiy YO'Q
++ --	Bitta songa oshirish, kamaytirish
@	Hatolarni berkitish
/ * %	Bo'lish, ko'paytirish, qoldiq
+ - .	Qo'shish, ayirish, kesish
<< >>	Chapga, o'nga surish
< <= > >=	kichik, kichik va teng, katta, katta va teng
== != === <>	Teng, teng emas, o'hshash, teng emas
& ^	AND, XOR, OR
&&	AND, OR
?:	uchlik operatori
= += *= /= .=	Belgilab berish operatorlari
%= &= = ^=	
<<= >>=	
AND XOR OR	Mantiqiy va, XOR, yoki

Keling yuqoridagi operatorlar bilan aloqali ba'zi bir misollarni ko'rib chiqaylik.

```

$a=5;           // a teng bechga
$a=3*3*4*5;     // a teng shu sonlar ko'paytmasiga
$a=(5-(6-4));   //avval 6-4 songra esa 5-(6-4) bo'ladi
$a+=5;          //$a ning qiymatiga 5 qo'shilib yana $a ga yozish.
$a/=5;          //$a ning qiymati 5'ga bo'linib keyin esa qiymatni $a'ga yozadi.
$a++;           //$a ning qiymatiga 1 qo'shiladi
$a--;           //$a ning qiymatidan 1 ayirib tashlanadi

```

lekin operandlarni joylashishini ham o'zgartirib yozishingiz mumkin, ammo uni bajaruvchi ishi ozgina o'zgaradi.

misolda ko'raylik

```

$qiymat=$a++    // avval $qiymat'ga $a ning qiymatini yozadi, so'ngra $a ning qiymatiga 1
qo'shadi.

```

PHP Davrasida kitobi Open Source

`$qiymat=++$a` *//avval \$a ning qiymaga 1 qo'shiladi, so'ngra \$qiymat ga \$a ning qiymatini beradi.*

agarda siz tenglikni tekshirmoqchi bo'lsangiz unda:

`$a==$b` *//\$a ni \$b bilan teng/tengmasligini tekshiradi*

`$a<$b` *//\$a ni \$b dan kichikligini tekshiradi*

Uchlik tekshirish.

`($a-12)?5:-1` *//agarda \$a=12 bo'lsa unda 5 soni qaytadi, agarda teng bo'lmasa 1 soni qaytadi.*

round()

Bu funktsiyamiz qandaydir qoldikli sonni yahlitlash uchun ishlatiladi.

`float round(float son [,nuqtadan_keyin])`

bu yerda son bu biz yahlitlamoqchi bo'lgan sonidir,

`nuqtadan_keyin` bu nuqtadan so'ng qancha songacha yahlitlashni bildiradi.

misol:

`<?`

`$son=12.3256;`

`echo $round($son, 3);`

`echo $round ($son, 2);`

`echo $round ($son, 1);`

`?>`

shunda bizning oynaga quydagilar chiqadi:

12.326

12.33

12.3

Bu yerda ko'rganingizdek 5 va undan yuqori bo'lsa unda oldindigi songa 1 qo'shadi agarda 5 dan kam bo'lsa unda faqatgina olib tashlaydi holos.

ceil()/floor()

`ceil()` - bu keyingi songa yahlitlaydi (12.3 bolsa unda 13 bo'ladi)

`floor()` - bu bitta past songa yahlitlaydi (12.6 bo'lsa unda 12 bo'ladi)

Yani bu yerda qoldiqga qaralmaydi faqatgina butun songa yani bitta yuqorisini olish uchun `ceil` ishlatiladi va bitta pastgi sonni olish uchun esa `floor` funktsiyasi ishlatiladi.

abs()

Bu funktsiya sonni oldidagi belgisiga olib tashlaydi, yani agarda sizda manfiy son bo'lsa unda uni musbatga o'giradi.

misol:

`<?`

`echo abs(-4.2);` *// oynaga 4.2 chiqadi*

`echo abs(4.2);` *// oynaga 4.2 chiqadi*

`echo abs(5);` *// oynaga 5 chiqadi*

`?>`

rand()/srand()

Qandaydir tasodifiy son olish uchun shu ikki funktsiya ishlatiladi.

PHP maslahat bergan ishlatish quyidagidir.

```
srand((double)microtime()*1000000)  
$randomnumber=rand()
```

Mana shu ko'dni ishlatsangiz sizda haqiqatdan ham tasodifiy son bo'ladi. Agarda siz ikki son orasida son omoqchi bo'lsangiz unda *rand(0, 10)* kabi ishlatishingiz mumkindir, shunda sizda 0 va 10 orasida qandaydir son chiqadi.

IF

If operatori, bu bizning tekshirish operatirimizdir.

Yozilishi:

```
if (tekshirish) {  
  yozuvimiz  
}
```

yoki

```
if (tekshirish) {  
  yozuvimiz  
}  
else {  
  yozuvimiz  
}
```

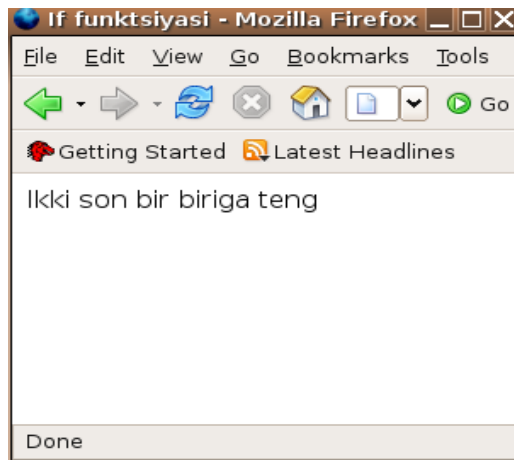
Misol:

```
if($a==$b) {  
  print “\$a teng ekan \$b ga”;  
}  
else {  
  print “teng emas ekan”;  
}
```

Bu tekshirishimiz, bizga \$a va \$b ni bir biri bilan solishtirib agarda ular teng bo'lsa unda ekranimizga “\$a teng ekan \$b ga” yozuvini chiqaradi, agarda teng bo'lmasa unda “teng emas ekan” yozuvini chiqaradi.

keling butun bir misolni ko'rib chiqamiz.

```
<?
$a=15;
$b=15;
?>
<html>
<head>
<title>Iflar </title>
</head>
<body>
<?
if ($a==$b){
print "Ikki son bir biriga teng";
}
else{
print "Ikki son bir biriga teng emas ekan";
}
?>
</body>
</html>
```



Mana ko'rib turganingizdek bir biriga teng degan yozuv ham ekranimizga chiqdi.

elseif

Bu if operatorimizga yana bitta tekshirishni qo'shib beradi.

```
if (tekshirish) {
yozuv
}
elseif (tekshirish) {
yozuv
}
```

Misolda ko'raylik.

```
if ($a<200) {
print "Sonimiz 200 sonidan kichik ekan";
}
elseif ($a<500) {
print "Sonimiz 200 dan katta, lekin 500 dan kichik ekan";
}
else {
print "Sonimiz 500 dan ham katta ekan";
}
```

Mana shu misolda sonimiz tekshirilib, uni qiymatiga qarab ekranimizga yozuv chiqai.

if operatorini yozishni ikkinchi turi ham bordir. Bu ikki tur bir birdan qilish fazifasi bo'yicha hech qanday

farq qilmaydi.

if (tekshirish):

yozuv

else:

yozuv

endif;

Yuqoridagi yozish turi undan yuqoridagidan hech ham farq qilmaydi.

while

Bu operatorimiz qandaydir ishni bir necha marotaba qayta qayta qilish uchun juda ham qulaydir.

while (tekshirish):

yozuv

endwhile;

keling bir misol bilan ko'rib chiqaylik.

```
$n=5;
```

```
$ncopy=$n;
```

```
$factorial=1;
```

```
while ($n>0):
```

```
$factorial=$n*$factorial;
```

```
$n--;
```

```
endwhile;
```

```
print "The factorial of $ncopy is $factorial";
```

shunda oynangizga quyidagi chiqadi.

Ko'rganingizdek juda ham qulay va kerakli operatoridir.

do..while

Operator do..while huddi while kabi ishlaydi, lekin kichkina farqi tekshirish boshida emas ohirida bo'ladi, shuning uchun do..while eng kamida bir marotaba ishlaydi.

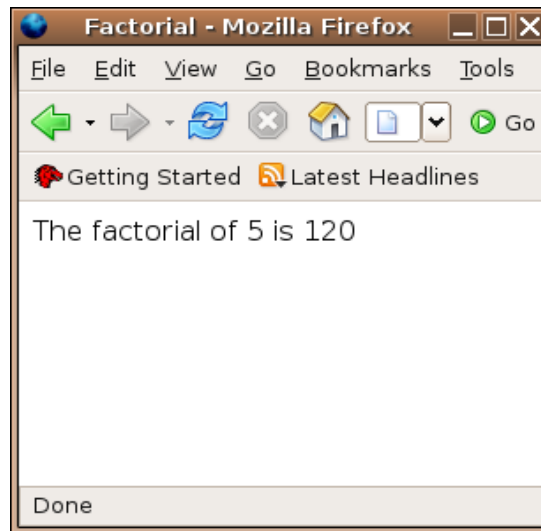
do:

yozuv

while (tekshiruv);

keling misolda ham ko'raylik.

```
$n=5;
```




```
$ncopy=$n;
$factorial=1;
do:
$factorial=$n*$factorial;
$n--;
while ($n>0);
print "The factorial of $ncopy is $factorial";
```

bu ham yuqoridagi misol kabi oynamizga “The factorial of 5 is 120” kabi yozuv chiqaradi.

for

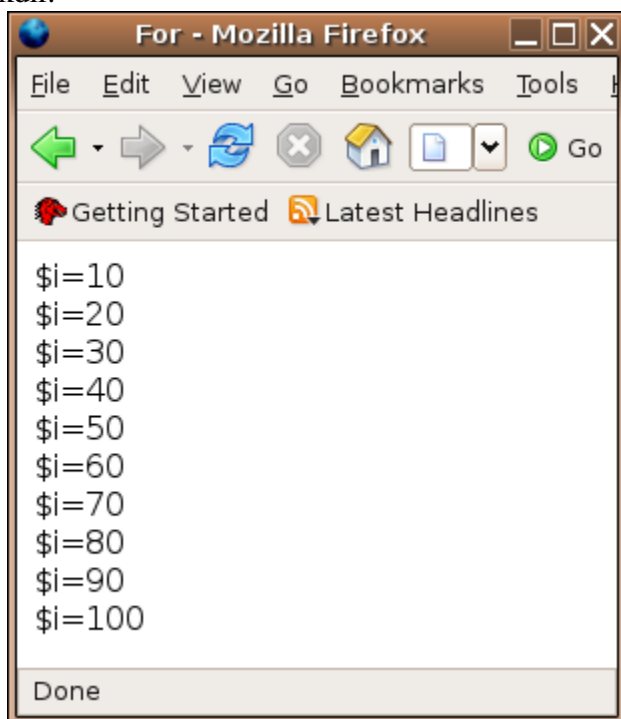
Bu operatorimiz biz uchun qandaydir programma ko'dimizni qanchadir marotaba qayta qayta ishlatib beradi.

```
for(tanitish:tekshiruv; o'zgarishi){
yozuv
}
```

Keling misol bilan ko'rib chiqamiz.

```
for ($i=10;$i<=100;i+=10){
print "\$i=$i <br>";
}
```

shuni qandaydir html ko'dingizga qo'yib tekshirib chiqing, shunda oynangizga shu kabi yozuv chiqishi kerakdir.



Bu misolda bizning operatorimiz \$i ga boshida 10 sonini beradi va uni yuzdan kichik yoki teng bo'lgunicha ishlatadi, va har bir oshishi 10 sonidir, har gal \$i'ga 10 soni qo'shilib boradi.

break

Bu qo'shimcha operatirimiz qandaydir ishni bo'lib qaytadan chiqib ketish uchundir.

Yani misolga qarang:

```
$x=5;  
for (;$x+=2) {  
  print "$x ";  
  if ($x==15):  
    break;  
  endif;  
}
```

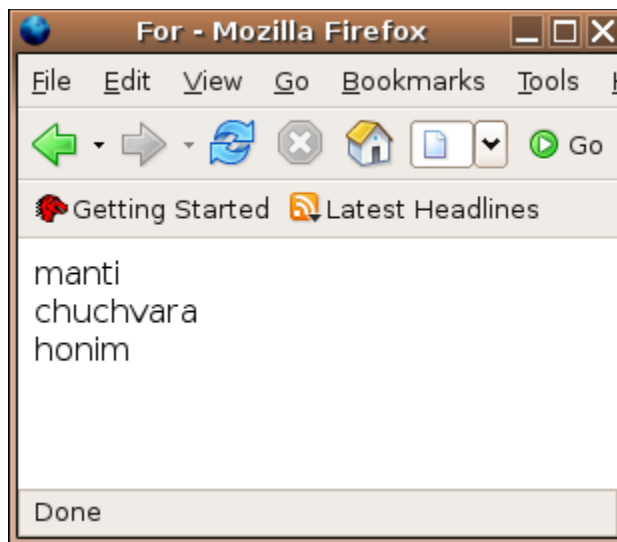
Shunda for operatormiz \$x ning 15 soni bo'lganigacha ishlaydi, for operatorini ishini esa break operatiri bo'ladi.

foreach

Massivlar bilan ishlash uchun for'ning qulaylashtirilganidir.

```
foreach (massiv as $element) {  
  yozuv  
}  
yoki  
foreach (massiv as $kalit=>$element){  
  yozuv  
}  
misol uchun:  
$menu=array("manti", "chuchvara", "honim");  
foreach ($menu as $item){  
  print "$item <br>";  
}
```

shunda oynangizga quyidagicha chiqadi.



switch

Huddi **if** operatiri kabidir, lekin undan farqi, birdaniga bir qancha tekshirishlar tekshiriladi.

Keling eng yahshisi misol bilan ko'raylik.

```
switch(tanitish){  
  case(shart):  
    yozuv  
  case(shart):  
    yozuv  
  ...  
  default:
```

```

yozuv
}

```

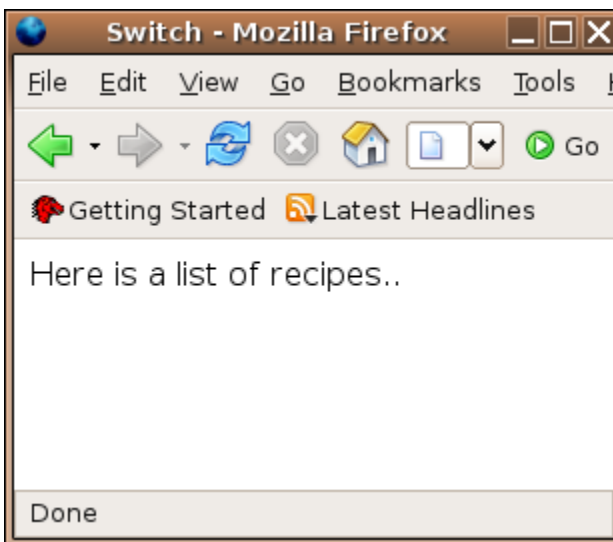
bunda tanitilgan o'zgaruvchanni shartlariga qarab turib nimani bajarishini belgilab berasiz. Agarda hech qanday shart bajarilmasa unda **default**'ning ichidagi ish bajariladi.

Misol:

```

$user_input="recipes";
switch($user_input){
case("search"):
print "Lets perform search";
break;
case("dictionary"):
print "What word would you like to look up?";
break;
case ("recipes"):
print "Here is a list of recipes...";
break;
default:
print "Here is the menu...";
break;
}

```



Mana ko'rganingizdek \$user_input bir qancha so'zlarga tekshiriladi (search, recipes, dictionary) va recipes to'g'ri kelganligi uchun undagi ish nima bo'lsa shuni bajariladi. Nima uchun break ishlatdik, sababi ishni qilib bo'lgandan keyin biz switch operatorini ishini bitirishini hohlaymiz, qolganlarini tekshirishi bizga kerak emas, shunda ortiqcha vaqt ketmaydi.

continue

Agarda for, yoki boshqa davomli operatorlaringizni ichida yozgan ko'dingizni qandaydir qismidan so'ng qolganlarini bajarilmasdan yana boshidan boshlanishini hohlasangiz unda **continue** operatorini ishlatasiz. Keling misolda ko'raylik.

```

$boundary=100;
for ($i=0;$i<=boundary; $i++){
if (! is_prime($i)):
continue;
endif;
$prime_counter++;
}

```

Bu misolda toq bo'lgan son bo'lsa (is_prime) unda for ishini davom etiradi, yani \$prime_counter++ operatori ishmasdan boshiga for operatoriga ish beriladi. Boshiga ish berilishi degani for yana boshidan

ishini boshlamaydi, vaholanki for o'z ishini davom etiradi, lekin for ning ifdan keyingi ishlari bajarilmasdan boshiga qaytadi.

Funtsiya nima?

Funktsiya bu programmadagi bir programma parchasi bo'lib o'z ismi bo'ladi. Funktsiya qandaydir aniq bor ishni bajaradi. Funktsiyani ustunligi shundaki funktsiya bir marotaba yozilib uni hohlagan joyingizda qayta qayta ishlatishingiz mumkindir. Va agarda sizga kerak bo'lsa unda uni osongina o'zgartirishingiz mumkindir.

Yuqorida biz funktsiyani azgina bo'lsa ham ko'rib chiqdik, bu yerda esa sizga batafsil yani to'la shakilda tushuntirib o'taman.

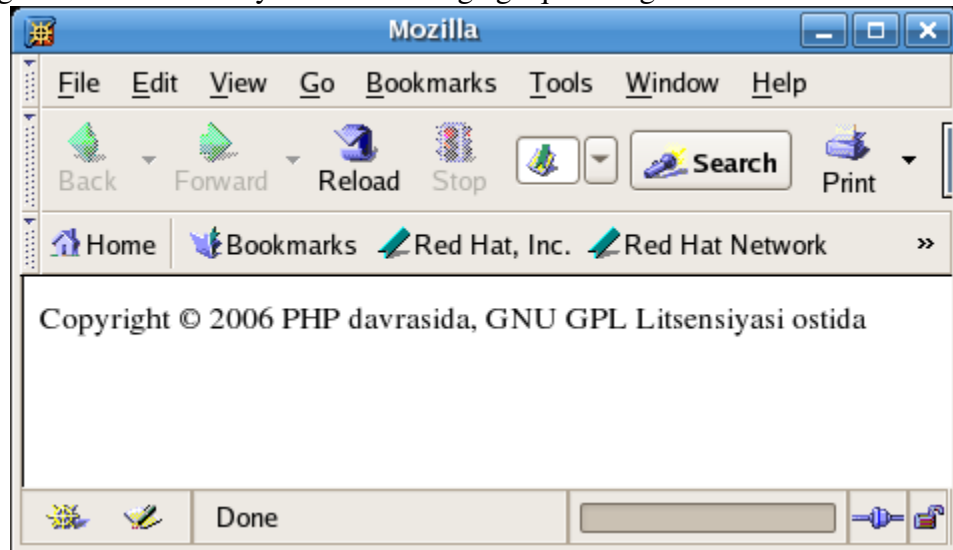
Yozilishi

```
function ($o'zgaruvchi1, $o'zgaruvchi2, $o'zgaruvchi3...){  
yozuvlar  
}
```

misolda:

```
function copyright(){  
print "Copyright &copy; 2006 PHP davrasida, GNU GPL Litsensiyasi ostida"  
}
```

agarda siz shu funktsiyani html ko'dingizga qo'shsangiz unda



shu kabi yozuv chiqishi kerak.

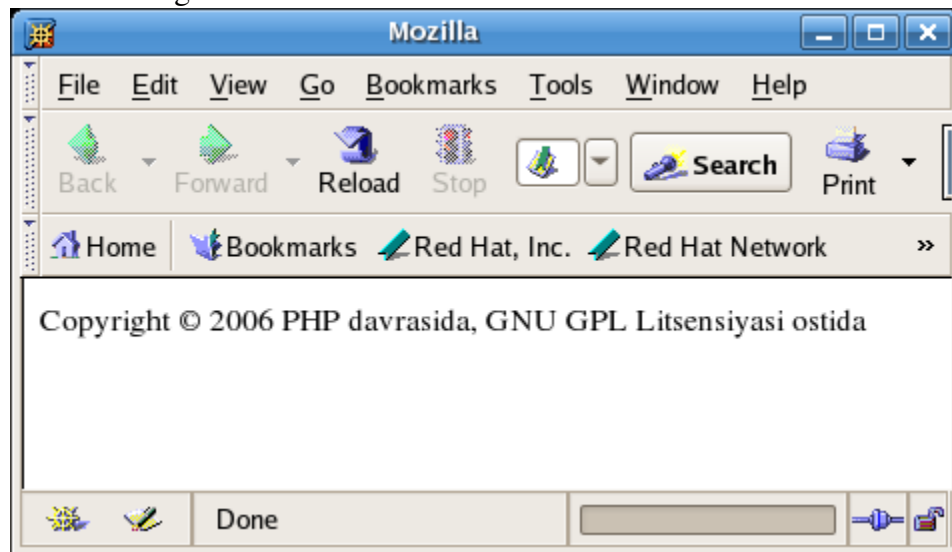
Funktsiyani siz bir birini ichiga qo'shib ham yozishingiz mumkindir. Va yana bir misol, agarda siz universal copyright funktsiyani yozmoqchi bo'lsangiz unda sahifa yoki nimanidir nomini tashqaridan olishingiz mumkin. Misolga qarang

```
function copyright($kitob){
```

PHP Davrasida kitobi Open Source

```
print "Copyright &copy; 2006 $kitob , GNU GPL Litsensiyasi ostida"  
}  
copyright("PHP davrasida");
```

shunda ekraningizda



yuqoridagi kabi chiqadi.

Funktsiyangiz qandaydir yozuv yoki sonni ham sizga qaytarishi mumkin. Yanikim, agarda siz hisob kitob ishini qilmoqchi bo'lsangiz va sizga uni natijasi kerak bo'lsa unda siz funktsiyangizdan qandaydir natijani qaytarishingiz mumkindir.

Misolga qarang:

```
$narh=15;
```

```
$son=3;
```

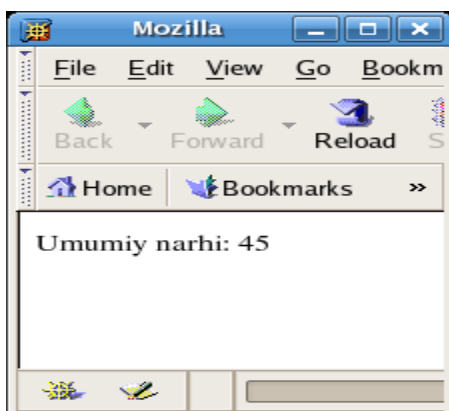
```
function hisob($narh, $son){  
return $narh*$son;  
}
```

```
$total=hisob($narh, $son);
```

```
print "Umumiy narhi: $total";
```

shunda sizga narhni songa qanchaligiga ko'paytirib natijasini chiqarib beradi.

Va o'ynamizda:



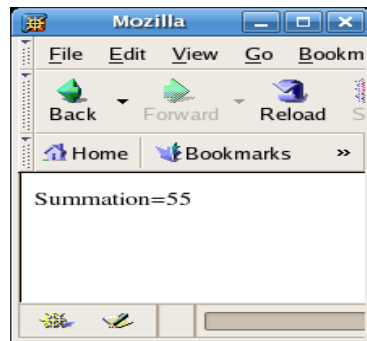
© Toshkent 2006 GNU GPL

kabi yozuv chiqadi. Mana ko'rib turganingizdek funktsiyalar juda ham qulay va kerakli programmalash asbobidir.

Yana funktsiyalarni ich ichiga, yani o'zidan o'zini chaqirishingiz mumkindir, bu recursive function deb ataladi. Misol bilan ko'raylik:

```
function summation($count){  
if ($count!=0):  
return $count+summation($count-1);  
endif;  
}  
$sum=summation(10);  
print "Summation=$sum";
```

bu yerda qilinadigan ishi shuki, avval 10 ga 9 ni qo'shadi yahi, o'zini yana chaqiradi faqat 9 soni bilan 9 sonidagi funktsiya esa 8 bilan yana o'zini chaqiradi bu ohirida 0 bo'ganda tugab sizni ekranga quyidagini chiqaradi:



Bu juda ham qulaydir.

Agar siz juda ham ko'p funktsiyalar ishlatish niyatida bo'lsangiz unda funktsiyalar uchun alohida bir fayl yaratib shu faylga hammasini yozishingiz mumkin, keyin esa shu faylni asosiy oynangizda chaqirib ichidagi funktsiyalarni ishlatishingiz mumkin.

Buning uchun qandaydir qoshimcha.inc yaratib yani fayl yaratib, uni ichiga bir qancha funktsiyalar yozing.

Misol: qoshimcha.inc

<?

```
function copyright(){  
....  
}
```

```
function hisob(){  
...  
}  
function summation (){  
...  
}  
?>
```

Keyin esa asosiy ko'dingizda
include (faylga yo'l);
operatori bilan qo'shing. Yani:
<?
include (qoshimcha.inc);
hisob();
\$sum=summation(10);
...
?>

o'zingiz qandaydir funktsiyalar yozing va tekshirib yahshilab o'rganib chiqing.
Shu bilan funktsiyalar degan mavzuyimizni yopiq deb bilaman va keyingi mavzuga otaman.
Agarda savollaringiz bo'lsa unda hatlaringizni uzinfo@inbox.ru / kitob@uzinfo.info kutib qolaman!!!

O'zgarmaslar

Quyida o'zgarmas va o'zgaruvchanlarni bitta nom bilan ataymiz, yani O'zgarmaslar deb atab ketamiz. Bu mavzuyimizni biz yana ham yoritib, yana ham ochib beraylik. Oldingi darslarda biz o'zgarmaslarni ko'rdik, lekin chuqur ko'rib chiqmagan edik, shuning uchun biz bu mavzuyimizda o'zgarmaslarni to'la ko'rib chiqamiz.

array()

Bu o'zgarmaslar turini biz yuqoridagi darslarimizda ko'rdik

list()

Array kabi, ammo farqi shundaki list operatori bilan siz bir qancha o'zgarmasga ma'lumot berishingiz mumkin. Mana quyidagi misolga qarang.

```
list($name, $surname, $company)=split(" ", "Sandjar | Alimov | BAT");  
print "$name <br>";  
print "$surname <br>";  
print "$company <br>";
```

shunda ekraningizda har bir o'zgarmasning ma'lumotlari yani:

Sandjar

Alimov

BAT

degan yozuv chiqadi.

range()

Bu operatorimiz butunsonli o'zgarmaslarni yaratishda juda qulay va tezdir.

```
$lottery=range(0, 9);
```

bu yerda biz lottery'ga 0 dan 9 gacha bo'lgan sonlarni tanitib beramiz.

in_array()

Bu funktsiya qandaydir o'zgaruvchanlar massivining ichida biz qidirayotgan ma'lumot bor yo'qligini tekshirib beradi.

misol:

```
$language=array("English", "Gealic", "Spanish");  
$exist=in_array("Spanish", $language); //bu yerda exist'ga TRUE ma'lumoti yoziladi.
```

array_keys()

Agarda sizda bir biriga ulangan qandaydir o'zgarmaslar to'plami bo'lsa, va uning asosiy nomlari yani kalitlarini ismini olish uchun array_keys() funktsiyasi ishlatiladi.

misol:


```
$great_wines=array("Australia"=> "Clarendon Hills 96", "France"=> "Comte George de Vogue 97", "Austria"=> "Feiler Artinger 97");  
$great_labels=array_keys($great_wines);
```

shunda

```
$great_labels[0]=Australia
```

```
$great_labels[1]=France
```

```
$great_labels[2]=Austria
```

bo'ladi.

array_values()

array_keys kabi faqatgina uni ishini teskarisini bajaradi, yanikim kalitlarni emas, shu kalitlarga tegishli bo'lgan ma'lumotlarni chiqarib beradi.

array_push()

Bu operatorimiz, bizning o'zgarmlarning ohiriga bitta yoki bir qancha belgi yoki so'z qoshadi.

```
Int array_push(array o'zgaruvchan, mixed so'zlar [, ...])
```

Misol:

<?

```
$languages=array("Uzbek", "Russian", "English");
```

```
array_push($languages, "French", "German");
```

```
foreach ($languages as $item){
```

```
print "$item <BR>";
```

```
}
```

?>

Shunda ekraningizda Uzbek, Russian, English, French, German chiqadi.

array_pop()

array_push'ning tamoman teskarisidir, yani masivning eng ohirgi elementini o'chiradi va uni qaytaradi, yani:

```
$languages=array("Uzbek", "Russian", "English");
```

```
$a_language=array_pop($languages); //a_language="English, languages="Uzbek", "Russian"
```

array_shift()

Aynan array_pop kabi ammo farqi shundaki masivni boshidan bitta elementni o'chirib keyin qolganlarini bitta oldinga suradi.

Huddi array_pop kabi u ham o'chirilgan elementni qaytaradi.

array_unshift()

array_shift ni tamoman teskarisin yani masivni boshiga bir element (so'z) qo'shadi.

Keling misolga qarang:

```
$languages=array( "French", "Russian", "German");  
$a_lang=array_shift($languages); //a_lang="French"  
array_unshift($languages, "Uzbek"); //languages="Uzbek", "Russian", "German"
```

array_pad()

Bu operator, masivni biz hohlagancha hajmini oshirib beradi, yani biz masiv (o'zgarmas) yaratganimizda uni hajmi karoq bo'lgan bo'lishi mumkin, biz uni kattalashtirish uchun shu operatoridan foydalanamiz.

Misolga qarang.

```
$weight=array(1, 2, 3, 15, 35);  
$weight=array_pad($weight, 8, 100);  
//bu yerda biz uni hajmini 8 gacha oshirib, yangi yaratilgan hujralarga 100 sonini yozib chiqishini  
belgiladik.  
//$weight=1, 2, 3, 15, 35, 100, 100, 100
```

agarda biz sonlarni boshiga qo'shmoqchi bo'lsakchi unda nima qilamiz?

Misolga qarang

```
$weight=array(1, 2, 3, 15, 35);  
$weight=array_pad($weight, -8, 100);  
//shu yerda biz $weight nin boshiga 100 raqamidan 3 donasini qo'shamiz, yanikim, weight'ning hajmini 8  
oshiramiz, va faqat bosh joylarni weightning boshiga qoshib qo'yamiz. :)
```

reset()

Bu operatorimiz, qandaydir masiv (o'zgarmaslarni) ichidagi ko'rsatgichni (kursor, cursor) masiv elementlarining eng birinchisiga olib borib qo'yadi, va birinchi element qiymatini qaytaradi.

Misol:

```
$fruits=array( "apple", "orange", "banana");  
$a_fruits=reset($fruits); //a_fruits=apple
```

each()

Each() operatori har gal ishlatilgan ikki ish qiladi, yani kursor turgan o'zgavchanni qaytaradi, va kursorni keyin masiv elementiga o'tkazadi. Va belgilangan o'zgarivchanga ikkita son yuboradi, yani

Misolga qarang:

```
$spices=array( "parsley", "sage", "rosemary");  
$a_spice=each($spices);  
print "$a_spice[0] $a_spice[1] <BR>";  
$a_spice=each($spices);  
print "$a_spice[0] $a_spice[1]";
```

shunda oynangizga "0 parsley 1 sage" yozuvi chiqadi, bu degani, avalambor birinchi elementni o'qib unga tegishli bo'lgan kalit va ma'lumotni a_spice'ga yozadi yani a_spice[0]'da kaliti, a_spice[1]'da esa uni qiymati yoziladi. Keyin esa ikkinchi element o'qilib (sage) u ham a_spice'ga yoziladi.

end()

Kursorni masivning eng ohiriga olib borib qo'yadi.

```
end(array masiv);
```

next()

Hudi each() operatori kabidir faqatgina farqi kursorni bitta oldinga surgandan keyin, yangi turgan yeridagi elementni qaytaradi.

prev()

Next()'ning tamoman teskarisi.

array_walk()

Juda ham qulay operator, yanikim qiladigan ishi, qandaydir funktsiyaga, qandaydir o'zgarmaslarni yuborishi mumkindir. Keling bir masiv'da qaytarilib kelgan o'zgarmaslar bo'lsa ularni ochirib tashlashini qiluvchi funktsiya, yoki bir ko'd yozaylik.

```
Function delete_dupes($element){
static $last="";
if ($element==$last) {
unset($element);
}
$last=$element
}
$emails=array( "bom@bom.ru", "test@test.ru", "bom@bom.ru");
sort($emails);
reset($emails);
array_walk($emails, "delete_dupes");
```

shunda sizda emails="[bom@bom.ru](#)", "[test@test.ru](#)" qoladi, ko'rganingizdek ikta bir hil o'zgarmasni olib tashladi.

array_reverse()

Array_reverse operatori butun masivni teskarisiga qilib joylashtiradi, yani masivni ichidagi elementlarni teskarisiga qo'yib chiqadi.

Misolga qarang:

```
$machines=array( "DONG", "Mitsubishi", "BMW");
$rev_machine=array_reverse($machine);
//rev_machine="BMW", "Mitsubishi", "DONG" kabi qiymatlarni oladi.
```

array_flip()

Agarda kalit/qiymat kabi masiv, o'zgaruvchi yaratgan bo'lsangiz, va ularni o'rni bilan almashtirish niyatingiz bo'lsa unda shu operatoridan foydalaning.

Misol:

```
$til=array("France"=>"French", "Uzbekistan"=>"Uzbek", "England"=>"English");  
$tes_til=array_flip($til);  
//shunda tes_til'ning qiymati quyidagicha bo'ladi tes_til="French"=>"France", "Uzbek"=>"Uzbekistan",  
"English"=>"England"
```

Mana ko'rganingizdek bularni hammasi sizga kerakli bo'lgan operatorlar va funktsiyalardir. Ammo bular bilan bir qatorda yana boshqa operatorlar ham bordir, ular masivlanri hajmini ko'rsatuvchi funktsiya va operatorlardir.

sizeof()

Bu operator, masivning ichinda qancha elementlari bo'lsha shularni ko'rsatadi, qaytaradi.

```
$langs=array("English", "Russian", "Greec", "German");  
$lang_size=sizeof($langs);
```

```
//lang_size=4
```

count()

sizeof() kabidir ammo bitta farqi qo'shimcha ma'lumot berishi mumkindir

1. Agarda masiv bo'lsa va unda qiymat bo'lsa, qancha elementi bo'lsa shuni sanab beradi
2. Agarda masiv emas bir dona o'zgarivchi bo'lsa unda 1 soni qaytadi
3. Agarda o'zgarvchi ham bo'lmasa unda 0 raqamini qaytaradi.

array_count_values()

Qandaydir masivimizning ichida agarda bir qancha o'zgarmaslar qaytarilib kelgan bo'lsa shularni nechtaligini ko'rsatadi.

```
$params=array("PA", "MA", "CA", "PA", "MO", "CA", "PA", "MA", "CO");  
$params_c=array_count_values($params);
```

```
//params_c="PA"=>3, "MA"=>2, "CA"=2, "MO"=>1, "CO"=>1 kabi qiyamtlarni oladi.
```

Mana endi keling masivlarimizni ichidagi o'zgarmaslarni saralashni ko'rib chiqaylik.

sort()

Oddiy saralash, yani ketma ket qilib qo'yib chiqishdir.

```
$alfa=array("A", "C", "E", "D", "B");  
sort($alfa);  
//alfa="A", "B", "C", "D", "E"
```

rsort()

sort() kabidir faqatgina teskarisiga saralab beradi.

asort()

Ayana yuqoridagilar kabi, ammo saralagandan so'ng hamma tegishli indexlar (kalitlar) o'rni saqlanib qoladi.

Misol:

```
$cities=array("Tashkent", "Samarkand", "Andijan", "Qarshi");  
asort($cities);
```

```
shunda  
cities[2]=Andijan  
cities[1]=Samarkand  
cities[0]=Tashkent  
cities[4]=Qarshi
```

kabi bo'ladi, yani hamma kalitlar saqlanib qolgan holda saralab berdi.

arsort()

asort() kabi faqatgina tesklari tomonga saralaydi.

ksort()

Masivdagi kaltilarni saralaydi, va shu bilan bir qatorda kalitlarga tegishli bo'lgan ma'lumotni saqlab qoladi.

Misol:

```
$langs=array("France"=>"French", "England"=>"English", "Italy"=>"Spanish");  
ksort($langs);
```

```
shunda langs:  
England=>English  
France=>French  
Itali=>Spanish
```

kabi qiymatlarni o'ziga qoladi.

krsort()

Aynan ksort() kabi, faqatgina shu ishni teskarsiga saralab chiqadi.

usort(), uksort, uasrot() bulan siz hohlagan sarlash funtsiyasi bilan ishlatishingiz mumkin. Siz hohlagan funtsiya, bu siz yozgan funksiya ham bo'lishi mumkindir. Bu operatorlarni o'zingiz mustaqil ko'rib chiqing, chunki bu uncha ham qiyin operatorlar emasdir.

array_merge()

Bir qancha masivlarni birlashtirish.

Misol:

```
$a1=array("a", "b", "c");
```

```
$a2=array("d", "e", "f");
```

```
$a3=array("g", "h", "i");
```

```
$a4=array_merge($a1, $a2, $a3);
```

//a4="a", "b", "c", ...i" mana shu holga keladi, yani a1, a2, a3 larni qoshib qo'yadi.

array_slice()

Masivning, o'zgaruvchaning ichidan qandaydir qismini kesib olish, yani boshlangich va qancha kesish kerakligini ko'rsatsangiz yetarli, yani:

```
array array_slice(array masiv, int boshlanish, [int uzunligi]);
```

array_splice()

Qandaydir masivni ichini qandaydir qismi o'zgartirish yoki o'chirish uchun ishlatiladi.

```
array_splice(masiv, int boshlanish, [int uzunlik], [array o'zgartirish masivi]);
```

shuffle()

Masiv ichini tarqatib tashlash, yani ichini tasodifiy ravishda o'rnilarini belgilab beradi.

Shular bilan o'zgaruvchanlar bilan ishlovhci operatorlarni tugatib, boshqa mavzuga o'tamiz.

Object Oriented Programming (OOP)

Bu programmalash, yangi programmalash turidir, yani, agarda siz C++, Visual Basic, Delphi kabi programmalash tillarini o'rgangan bo'lsangiz bu siz uchun yangilik bo'lmaydi, agarda bilmasangiz unda shu mavzuni ko'rib chiqqaningizdan so'ng sizda bu programmalash turi haqida ma'lumotingiz bo'ladi. OOP-Bu programmangizda qandaydir bir object yaratib undan keyingi o'zingizga kerak bo'lgan yerlarda ishlatishdir. Yani bo'lmagan bir object hosil qilishdir.

Class

Class: bu OOPning asosidir. Yani siz klas yaratasisiz va undan keyinchalik foydalanishingiz mumkindir.

Keling misolda ko'raylik

```
class Class_name{                //Class_name nomli klas yaratdik
var $attribute_1;                //ichida hohlaganimizcha o'zgaruvchi tanitamiz
...
var attribute_2;
function function1(){            //qancha istasak shuncha funktsiya yaratamiz
...
}
...
function functionN(){
...
}
...
}                                //bu yerda esa Class_Name yopiladi.
```

Mana ko'rganingizdek biz bu yerda class bilan boshladik va ichida kerakli o'zgarmas, va kerakli funktsiya yaratib berdik.

Agarda siz obyektningizni ichida shu obyektningizni qandaydir funtsiyasiga yoki o'zgaruvchisiga murojat qilmoqchi bo'lsangiz unda **\$this** funtsiyasidan foydalanasiz.

Misol:

```
<?
class Webpage(){
var $bgcolor;
function setBgColor($color) {
$this->bgcolor=$color;
}
function getBgColor(){
return $this->bgcolor;
}
}
?>
```

PHP Davrasida kitobi Open Source

Mana biz o'zimizga klas yaratdik, endi uni qanday qilib ishlatishimiz mumkindir? Endi obyekt yaratamiz.

```
$home_page=new Webpage;           //home_page nomli obyekt yaratdik, yani u aynan Webpage kabidir
```

```
$home_page->setBgColor("black");    //Bu yerda esa biz home_page'ga black rangini yubordik.
```

Siz bu yaratgan klaslardan yana bir qancha ichki klas yarata olasiz, yanikim, Webpage klasidan foydalanib unga boshqa qo'shimchalar ham qo'shishingiz mumkindir.

Misol:

```
<?
class Webpage(){
var $bgcolor;
function setBgColor($color) {
$this->bgcolor=$color;
}
function getBgColor(){
return $this->bgcolor;
}
}
class UzWebpage extends Webpage{
var admin;
function setSiteAdmin($adm){
$this->admin=$adm;
}
function getSiteAdmin(){
return $this->admin
}
}
?>
```

Mana bu yerda biz Webpage'ni yana ham kengaytiridik, yani UzWebpage qildik va unga qo'shimcha funktsiya berdik, yani sitening administratorini berib chiqdik.

Array `get_class_methods(string class_ismi)`

Bu funktsiyamiz klasning ichida tanitilgan funktsiyalarni ko'rsatib beradi.

Misol uchunyuqoridagini tekshirib ko'raylik

yani eng ohiriga

```
$test=get_class_methods(Webpage);
print $test[0];
print $test[1];
```

yozib tekshirib ko'ring, sizda setbgcolor, va getbgcolor chiqishi kerak.

Array get_class_var(string class_ismi)

Classning ichidagi o'zgarmaslarni qaytaradi. Yani yuqoridagi get_class_methods kabi ammo, farqi o'zgarmaslarni qaytarishidadir.

Array get_object_vars(object_ismi)

Yaratilgan obyektning ichida agarda qandaydir o'zgarma nima tanitilgan bo'lsa shularni hammasini ko'rsatib beradi.

Kelin misolga qarang:

```
<?
class Vehicle{
var $wheels;
}
class Land extends Vehicle{
var $engine;
}
class car extends Land{
var $doors;
function car($doors, $eng, $wheels){
$this->doors=$doors;
$this->engine=$eng;
$this->wheels=$wheels;
}
function get_wheels(){
return $this->wheels;
}
}
$toyota=new car(2, 400, 4);
$vars=get_object_vars($toyota);
while (list($keys, $value)=each($vars)):
print $keys==> $value <br>";
endwhile;
?>
```

shunda oynangizda

doors==>2

engine==>400

wheels==>2

chiqadi.

method_exists()

Obyektimizda qandaydir funktsiya bor yo'qligini tekshirib chiqadi. Agarda bor bo'lsa unda TRUE agarda yo'q bo'lsa unda FALSE qaytadi.

get_class()

Qandaydir obyektimiz qaysi klasni ichidaligini qaytaradi.

Misolga qarang.

```
Class Vehicle {  
...  
}  
class Land extends Vehicle {  
...  
}
```

```
$car=new Land;  
$class_a=get_class($car);
```

shunda class_a=Land bo'ladi.

get_parent_class()

Ayana yuqoridagi kabi ammo, asosiy ona classni qaytaradi, yani

```
$class_a=get_parent_class($car);
```

class_a=Vehicle bo'ladi.

is_subclass_of()

Bu funktsiya qandaydir obyekt qaysidir klasdan yaratilib yaratilmaganini tekshiradi.

Bool is_subclass_of(object obyekt_ismi, string class_ismi)

get_declared_classes()

Bu funktsiya esa, qanday klaslar yaratilgan yani tanitilgan bo'lsa hamamsini ekranga chiqaradi.

Array get_declared_classes()

Mana shu bilan OOP haqida mavzuyimiz bitdi.

Fayllar bilan ishlash.

Azizlar, bu mavzuyimiz, PHP'da fayllar bilan ishlashni batafsil tushuntirib, bir qancha misollar bilan

ko'rsatib o'tadi.

file_exists()

Bu funktsiyamiz qanadidir fayl bor/yo'qligini tekshirish uchun kerakdir.

Bool file_exists(string fayl)

misol:

```
if(! file_exists($filename)):
print "File $filename does not exist!";
endif;
```

is_file()

Funktsiya file_exists funktsiyasi kabidir, ammo farqi shundaki, u fayl bor yo'qliginigina emas balki faylga yozish yoki fayldan o'qish mumkin mumkin emasligini ham tekshiradi.

Bool is_file(string fayl)

misol:

```
$file = "test.txt";
if (is_file($file)):
print "The file $file is valid and exists!";
else:
print "The file $file does not exist or it is not a valid file!";
endif;
```

Misolimizda test.txt nomli fayl borligi tekshiriladi va agarda bor bo'lsa va u bilan ishlash mumkin bo'lsa unda oynamizga "The file test.txt is valid and exists!" degan yozuv chiqaradi.

filesize()

Filesize() funktsiyasi faylni hajmini o'lchab beradi, agarda fayl yo'q bo'lsa unda FALSE qiymati qaytadi.

int filesize(string fayl)

misol:

```
$fs=filesize("pastry.txt");
print "Pastry.txt is $fs bytes.";
```

fopen()

Bu funktsiyamiz faylni ochib beradi, keyin esa shu faylning file handle yani shu faylning ochilish raqamini qaytaradi, shu raqam orqali keyin siz shu fayl bilan ishlashingiz mumkindir.

Int fopen(string fayl, string turi [, int joylashishini_yoqish])

Fayl ismi bir nechta turda yozilishi mumkindir:

1. oddiy fayl yo'lini ko'rsatish
2. php://stdin, php://stdout, php://stderr kabi agarda siz shunday bergan bo'lsangiz unda

kirish/chiqish (standart input/output) bilan ishlash mumkindir.

3. Http:// yoki ftp:// bo'lsa unda http serveridagi fayl bilan yoki ftp serveridagi fayl bilan ishlaydi

Ochish turlari:

| Ochish turi | Nimaligi |
|-------------|---|
| r | Faqat o'qish uchun |
| r+ | O'qish va yozish |
| w | Faqat yozish |
| w+ | Yozish va o'qish |
| a | Faqat yozish, ammo yozish faylning ohiridan boshlanadi |
| a+ | O'qish yozish, ammo yozish va o'qish faylning ohiridan boshlanadi |

Kichkina misolda ko'raylik

```
$file="userdata.txt";
```

```
$fh=fopen($file, "a+") or die ("File ($file) does not exist!");
```

bu misolda userdata.txt fayli yozish va o'qishga, ochiladi va kursor faylning ohiriga qo'yiladi, agarda yo'q bo'lsa unda sizga "File userdata.txt does not exist!" deb ishini tugatadi.

fclose()

Ochilgan faylni yopish uchun ishlatiladi, yopishda oldindan saqlab qo'yilgan file handle, yani ochish raqami ishlatiladi.

Int fclose(int raqam)

misol:

```
$file="userdata.txt";
```

```
if (file_exists($file)):
```

```
$fh=fopen($file);
```

```
fclose($fh);
```

```
else:
```

```
print "File $file does not exist!";
```

```
endif;
```

Bu misol aval fayl bor/yo'qligini tekshiradi, so'ngra esa bor bo'lsa ochadi, va keyin esa yopadi.

Agarda bunday fayl mavjud bo'lmasa unda ekranga bu fayl yo'q degan yozuv chiqib ishini tugatadi.

is_writeable()

Funktsiya fayl borligini tekshiradi, va unga yozish mumkin mumkin emasligini tekshiradi.

Bool is_writeable(string fayl)

fwrite()

Funktsiyasi, faylga ma'lumot yozish uchun ishlatiladi.

Int fwrite(int fayl_raqami, string o'zgaruvchan [, int uzunligi])

Keling misolda ko'raylik.

```
<?
$data="08:13:00 | 12:37:12 | 208.247.106.87 | win98";
$filename="test.txt";
if (is_writable($filename)):
$fh=fopen($filename, "a+");
$success=fwrite($fh, $data);
fclose($fh);
else:
print "Could not open $filename for writing";
endif;
?>
```

Shunda file'ning ichiga \$data'ning qiymati yoziladi. Ammo bu php kodini ishlatishdan oldin test.txt nomli fayl yaratib qo'yish esdan chiqmasin. Agarda yaratilmagan bo'lsa unda oynangizda "Could not open \$filename for writing" degan yozuv chiqadi.

fputs()

Bu funktsiyamiz aynan fwrite kabidir, uni fwrite'dan farqi hechham yo'qdir.

is_readable()

is_writeable kabi funktsiyadir, ammo farqi faylni o'qish mumkinligini tekshiradi.

Bool is_readable(string fayl)

fread()

fputs() kabidir, ammo farqi bu yozmaydi, qaytanga fayldan ma'lumot o'qiydi.

Int fread(int fayl_raqami, int uzunlik)

Kelin misolga qarang.

```
$fh=fopen("pastry.txt", "r") or die ("Can't open file");
$file=fread($fh, filesize($fh));
print $file;
fclose($fh);
```

Biz bu yerda butun fayldagi bor ma'lumotni \$file ga yozdik, va print orqali oynamizga chiqardik.

fgetc()

Bittadan belgilarni fayldan o'qish uchundir.

String fgetc(int fayl_raqami)

Misolga qarang:

```
$fh=fopen("pastry.txt", "r");
while(!feof($fh)):
```

```
$char=fgetc($fh);  
print $char;  
endwhile;  
fclose($fh);
```

Bu yerda biz butun faylni birma bir fayldan o'qib keyin oynaga chiqarib beradi (print funktsiyasi orqali) feof-funktsiyasi bu yerda faylning ohiriga yetib borgan bormaganligini bildiradi, biz while'da faylni ohiri bo'lmasa unda o'qi va oynaga chiqar komandasini berdik.

fgets()

Qatorma qator o'qish uchundir.

Misol:

```
$fh=fopen("pastry.txt", "r");  
while (!feof($fh)):  
$line=fgets($fh, 4096);  
print $line;  
endwhile;  
fclose($fh);
```

Mana bu yerda biz qatorma qator butun faylni o'qidik.

fgetss()

Aynan fgets kabidir, ammo farqi agarda o'qilayotgan ma'lumot ichida HTML yoki PHP ko'dlari bo'lsa ularni olib tashlashga harakat qiladi.

String fgetss(int fayl_raqam, int uzunlik, [, string olib_tashlanmaydigan_ko'dlar])

file()

Bu funktsiya, faylning ichidagi hamma qatorlarni bitta o'zgarmasga (masivga) yozadi.

Array file(string fayl [, int yol_yoqish])

misol:

```
<?file_array=file("pastry.txt");  
while (list($line_num, $line)=each($file_array)):  
print "$line_num: ", htmlspecialchars($line);  
endwhile;  
?>
```

Bu yerda siz bilmaydigan yana bitta funktsiya ishlatildi:

htmlspecialchars- bu funktsiya agarda ma'lumotlar ichida HTML ko'di bo'lsa unda uni qandoq bo'lsa shunday qoldiradi, yani uni huddi oddiy tekst deb biladi va ularni bajarmaydi. Agarda olib tashlasak bu funktsiyani tekstimizda HTML ko'di uchrasa ularni hammasini bajarib o'tadi.

Shu kichik ko'dimizda biz pastry.txt'ni file_arrayga yozdik va undan o'qib oynaga chiqardik.

readfile()

Funktsiyaning ishi shundan iboratki qandaydir faylni o'qib undagi ma'lumotni chiqish, yani Ineternet Browser'larga yuboradi.

Int readfile(string fayl [, int yol_yoqish])

Agarda sizda qandaydir faylingiz bo'lsa va uni ochib chiqishga yuborish uchun

```
<?
$file="test.txt";
readfile($file);
?>
```

shu ko'dning o'zi yetarlidir.

popen()

int popen(string buyruq, string turi)

Fayllarni oddiy ochishdan tashqari yana uni sistemada ochish mumkindir, yani siz uni ochish bilan bir qatorda yana u bilan har qanday sistemadagi programmalar bilan birga ish qildirtirishingiz mumkindir. Keling misolda ko'raylik.

```
<?
$fh=fopen("test.txt", "w"); //test nomli fayl ochamiz
fputs($fh, "Salom Toshkent\n");
fputs($fh, "Bugun havo yahshi\n");
fputs($fh, "Hayot gozal\n");          //faylning ichiga bir qancha ma'lumot yozamiz
fclose($fh);                          //faylni yopamiz
$fh=popen("grep Toshkent < test.txt", "r"); //Biz bu yerda Unix sistemasidagi grep buyrugini
                                           //ishlatib test.txt faylidan Toshkent so'zini qidiradi va uni topgan qatori ekranga chiqaradi

fpassthru($fh);                       //ekranga chiqaramiz
?>
```

pclose()

popen bilan ochilgandan keyin pclose bilan yopasiz.

fsockopen()

Bu funktsiyamiz Internetdagi qandaydir Server bilan TCP yoki UDP protokollari orqali bog'lanadi.

Int fsockopen(string url, int port [, int hat_raqami [, string hat_teksti [, int time_out]])

misol:

```
<?
$fp=fsockopen("www.example.com", 80, $errno, $errstr, 30);
if(!$fp){
```

```
echo "$errstr ($errno)<br>\n";  
}  
...  
?>
```

Mana shu kabi ishlatishingiz mumkindir.

socket_set_blocking()

Bu funktsiya fsockopen bilan ochilgan sessiyani vaqtinchaga, ishimizni tugatgungacha yopish yoki ochish uchun ishlatiladi

socket_set_blocking(int ochilgan_raqam, boolean tur)

pfsockopen()

fsockopen() kabidir, faqat farqi ulanishingiz avotmat shakilda uzilmaydi.

exec()

Tashqi programmani ishlatish, yani aga qda unix sistemasida ishlasangiz unda sizga gedit kabi programma avotmat shakilda ochilishi kerak bo'lishi mumkindir. Unda siz shu funktsiyadan foydalanasiz.

String exec (string buyruq [, string o'zgaruvchan [, int qaytish]])

misol:

```
<?  
exec ("ping -c 5 www.technoidea.boom.ru", $ping);  
for ($i=0; $i<count($ping); $i++):  
print "<br>$ping[$i]";  
endfor;  
?>
```

shunda sizga quyudagicha chiqadi

```
PING boom.ru (194.67.45.28) 56(84) bytes of data.  
64 bytes from 194.67.45.28: icmp_seq=1 ttl=247 time=95.1 ms  
64 bytes from 194.67.45.28: icmp_seq=2 ttl=247 time=93.8 ms  
64 bytes from 194.67.45.28: icmp_seq=3 ttl=247 time=93.6 ms  
64 bytes from 194.67.45.28: icmp_seq=4 ttl=247 time=93.7 ms
```

Teskari apostrov (`)

Bu exec kabidir, yani shu belgini orasidagi hamma komanda exec kabi ishlab keyin chiqqan ma'lumot o'zgarmasga yoziladi

misol:

```
$output=`ls`;  
print "<pre>$output</pre>";
```


bu kichik ko'dimiz, o'zi turgan katalog (direktorini) ichidagi hamma bor fayl va kataloglarni oynaga chiqaradi.

passthru()

Aynan exec kabidir, faqatgina bir farqi butun chiqqan ma'lumotni (qaytgan) oynaga, yani Internet Browserga yuboradi.

escapshellcmd()

string escapshellcmd(string buyruq)

Bu funktsiyamiz hamma hafli belgilarni o'chirib keyin buyruqni ishlatadi.

Yani agarda siz Unix sistemasida `rm -rf *` desangiz ko'dingiz turgan yerdagi hamma katalog va fayllar o'chib ketadi, shuning uchun agar siz `escapshellcmd` bilan ishlatsangiz `*` belgisi ishlatilmaydi, yani hech narsa o'chirilmaydi.

Misol:

```
$user_in=`rm -rf *`;
exec($user_in);          //hafli hamma fayllar o'chib ketishi mumkindir.
```

```
$user_in=`rm -rf *`;
exec(escapshellcmd($user_in));    //hafli emas, sababi * belgisi olib tashlanadi
```

basename()

Bu funktsiyamiz faylga yo'ldan yani misol uchun "[c:\program files\mozilla\mozilla.exe](#)" dan mozilla.exe, yani faylni ismini ajratib olib beradi.

Misolga qarang:

```
$path="/usr/local/sbin/skip.test";
$file=basename($path);
```

shunda `$file="skip.test"` bo'ladi.

getlastmod()

Bu funktsiya, faylni eng ohirgi marotaba o'zgartirilgan sana va vaqtni qaytaradi.

Int getlastmod(void)

misol:

```
echo "Last modified:", date("H:i:s a", getlastmod());
```

stat()

Bu fayl faqida jami ma'lumotlarni qaytaradi.

Array stat(string fayl)

Qaytaradigan ma'lumotlar soni 12'tadir ular

1. Device (qanday uskuna)
2. inode (indeks yo'li)
3. inode himoyasi
4. Yollanmalar soni
5. Fayl egasining raqami
6. Fayl egasining gruppasining raqami
7. inode uskunasi turi
8. Hajmi (byte)
9. Faylga ohirgi marotaba bo'lgan murojat vaqti
10. Faylni ohirgi marotaba o'zgartirilgan vaqti
11. Faylni ohirgi marotaba o'zgartirilgan vaqti
12. Faylni sistemaga yozishdagi bloklari
13. Ajratilgan bloklar soni

Keling misolga qaraylik

```
$file="datafile.txt";  
list($dev, $inode, $inodep, $nlink$, $uid, $gid, $inodev, $size, $atime, $mtime, $ctime,  
$bsize)=stat($file);  
print "$file is $size bytes. <br>";  
print "Last access time: $atime<br>";  
print "Last modification time: $mtime <br>";
```

shunda:

index.php is 285 bytes.

Last access time: July 15 2006 12:00:00

Last modification time: July 15 2006 11:00:00

Mana misoldan ko'rib turganingizdek biz o'zimizga kerakli bo'lgan ma'lumotlarni o'ynamizga chiqardik.

chgrp()

Faylning tegishli bo'lgan gruppasini o'zgartiradi.

Int chgrp(string fayl, mixed gruppa)

filegroup()

Faylning guppasining raqamini qaytaradi, yani egasi bo'lgan gruppasining raqamini.

Int filegroup(string fayl)

chmod()

Faylga bo'lgan ruhsatlarni, yani gruppa, yoki boshqalarni qanday huquqlari borligini aniqlash, yanikim yozish, o'qish yoki ishlatish huquqlarini o'zgartirishdir.

PHP Davrasida kitobi Open Source

Int chmod(string fayl, int ruhsat)

Huquqlar sakizlik sanoq sistemasida beriladi. Yani aynan Unix/Linux sistemalaridagi kabidir.

Misol:

```
chmod ("datafile.txt", 0755)      //egasiga hamma huqulqar yani 7
                                   //gruppaga o'qish va ishlatish yani 5
                                   //boshqalarga ham o'qish va ishlatish 5
```

fileperms()

chmod() funktsiyasi bilan berilgan ruhsatlarni ko'rsatadi. Yani faylga bo'lgan ruhsatlarni qaytaradi.

Int fileperms(string fayl)

chown()

Faylning egasini o'zgartirish uchundir.

Int chown(string fayl, mixed egasi)

fileowner()

chown() funktsiyasi bilan belgilab berilgan fayl egasini qaytaradi.

int fileowner(string fayl)

copy()

Faylni nushasini olish.

Int copy(string qayerdan, string qayerga)

misol:

```
$data_file="data1.txt";
copy($data_file, $data_file'.bak.') or die ("Could not copy $data_file");
```

rename()

Fayl ismini o'zgartirish.

Bool rename(string eski_ismi, string yangi_ismi)

misol

```
$data_file="data1.txt";
rename($data_file, $data_file'.old.') or die ("Could not rename $data_file");
```

unlink()

Faylni o'chirish.

Int unlink(string fayl)

dirname()

Qandaydir faylga yo'ldan, faqatgina katalogini olib ajratib beradi, yani basename() funktsiyasi kabi ammo bu kataloglar uchundir.

string dirname (string yol)

misol:

```
$path="/usr/local/index.php";  
$dirs=dirname($path);
```

shunda dirs="/usr/local" bo'ladi.

is_dir()

Katalogligini tekshiradi, agarda katalog bo'sa unda TRUE qaytaradi, agarda fayl bo'lsa unda FALSE qaytaradi.

bool is_dir(string ism)

```
$file="index.html";  
$dirs="/usr/local";  
$isdir=is_dir($file);           //isdir=FALSE  
$isdir=is_dir($dirs);          //isdir=TRUE
```

mkdir()

Yangi katalog yaratadi.

Int mkdir(string yol, int turi)

Bu yerda turi=katalogga qanday huquqlar berilishini belgilaydi.

opendir()

fopen() kabidir, ammo farqi bu kataloglar uchundir.

Int opendir(string yol)

closedir()

opendir() bilan ochilgan katalogni yopish uchun ishlatiladi.

Void closedir(int ochiq_katalog_raqami)

readdir()

Katalogni ichidagi ichki katalog, yoki fayllarni birma bir o'qib ismini qaytaradi.

String readdir(int ochiq_katalog_raqami)

misolga qarang:

```
$dh=opendir('.');  
while ($file=readdir($dh)):  
print "$file <br>"; endwhile;  
closedir($dh);
```

PHP Davrasida kitobi Open Source

Bu yerda biz turgan katalogimizni ichidagi hamma fayllarni va kataloglarni o'qib oynamizga chiqaramiz.

chdir()

Bu turgan katalogimizni o'zgartirish uchun ishlatiladi, yani DOS'dagi CD kabidir.

Int chdir (string katalog)

rewinddir()

opendir() bilan ochilgan katalogni boshiga olib borib qo'yadi (agarda keyin o'zgartirilgan bo'lsa).

Keling endi biz sahifamiz uchun bir kichik hisoblagich yozlaylik, yani har bir kirganni hisoblab boradi. Bu siz uchun kichik bir o'rganish va oldindan o'tilgan funktsiyalar bilan yana ham yaqinroq tanishishga yordam beradi.

Men sizga hisoblagichni ko'dini beraman va siz uni o'rganib, o'zingizga keraklicha qilib olishingiz mumkindir.

```
<?
```

```
$access="hits.txt";
```

```
$visits=@file($access);
```

```
$current_visitors=$visits[0];
```

```
$all_me=(int)$current_visitors+1;
```

```
$fh=fopen($access, "w");
```

```
$success=fwrite($fh, $all_me);
```

```
fclose($fh);
```

```
print "Visited: $all_me";
```

```
?>
```

Mana shu kichik ko'ddir.

Fayllar bilan ishlashga qo'shimcha

Ko'rganingizdek bizni faylni server'ga yuklashda kichkina muammoyimiz bor edi, yanikim yuklangan fayl o'z ismini emas vaqtincha berilgan ismini qabul qila olardi.

endi keling shuni yana boshqa yo'lini ko'raylik.

Quyida keltirilgan misolni ko'rib chiqing

test.html fayli:

```
<html>
```

```
<head>
```

```
<title> File Upload </title>
```

```
</head>
```

```
<body>
<h2> <b> Faylni tanlab yuklani bos </b></h2>
<form action="upload.php" method="post" enctype="multipart/form-data">
<input type="file" name="filename"><br>
<input type="submit" value="yukla"><br>
</form>
</body>
</html>
```

upload.php fayli:

```
<html>
<head>
<title> Yuklandimi? </title>
</head>
<body>
<?php
if(copy($_FILES["filename"]["tmp_name"],
"temp/".$_FILES["filename"]["name"]))
{
echo("Fayl yuklandi");
}
else
{
echo("Fayl yuklana olmadi");
}
?>
</body>
</html>
```

Mana misollardan ko'rgan bo'sangiz kerak biz bu yerda GLOBAL \$_FILES ni ishlatdik va faylni o'zini ismi bilan yukladik.

Yana qo'shimcha ma'lumot olish kerak bo'lsa unda quyidagi misolga qarang.

upload.php fayli

```
<html>
<head>
<title> Fayl yuklandimi? </title>
</head>
<body>
<?php
if(copy($_FILES["filename"]["tmp_name"],
"temp/".$_FILES["filename"]["name"]))
{
```

PHP Davrasida kitobi Open Source

```
echo("yuklandi <br>");
echo("Fayl bilan aloqali malumotlar: <br>");
echo("Ismi: " ) ;
echo($_FILES["filename"]["name"]);
echo("<br>Hajmi: ") ;
echo($_FILES["filename"]["size"]);
echo("<br>Yuklangan yeri: " ) ;
echo($_FILES["filename"]["tmp_name"]);
echo("<br>Fayl turi: " ) ;
echo($_FILES["filename"]["type"]);
}
else {
echo("Yuklanmadi");
}
?>
</body>
</html>
```

Yana bir qulay va kerakli funktsiya bu
scandir()

Bu siz bergan katalogni ichidagi hamma fayl va kataloglarni ko'rsatadi.
array scandir (string katalog [, int saralash])

Misol:

```
<?php
$dir = 'temp' ;
$files = scandir($dir);
$sort_files = scandir($dir, 1);
print_r($files);
print_r($sort_files);
?>
```

Shunda oynamizga quyidagi kabi bir ma'lumot chiqadi

Array ([0] => . [1] => .. [2] => xpcor_core.dll)

Array ([0] => xpcor_core.dll [1] => .. [2] => .)

O'zingiz qilish uchun:

Sahifangizni ichidagi hamma fayllarni ko'rsatuvchi yani html yoki php, yani sahifa haritasini php'da yozing va o'zingizni tekshirib ko'ring.

[]

Bu belgimiz, uning ishi shundan iboratki “ichidagi belgilardan istalgan biri”, yani agarda [php] bo'lsa unda belgilar qatorida agarda p, yoki h bo'lsa to'g'ri keladi.

[0-9] – 0 dan 9 gacha bo'lgan sonlardan istalgan biriga to'g'ri keladi.

[a-z] – a harfidan z harfigacha bo'lgan harflardan istalgan biri

[A-Z] – A harfidan Z harfigacha bo'lgan harflardan biri

[a-Z] – a harfidan boshlanib Z harfigacha yani hamma kichik harflar va hamma katta harflardir.

Bular nimaga kerak, agarda biz qandaydir qatordan nimanidir, yani sonnimi yoki harf yoki boshqa belgilardan birini qidirish kerak bo'lsa unda shu qo'shimcha belgilardan foydalanishimiz mumkindir. Yana pastda qo'shimcha boshqa belgilarni ham ko'rib chiqamiz, undan so'ng misollarda ko'ramiz.

Belgilar

p+ bitta yoki yonma yon turgan bir qancha p harfi

p* nol yoki yonma yon turgan bir qancha p harfi

p? nol yoki bitta p harfi

p{2} ikta p harfini bildiradi

p{2,3} iktadan uchtagacha bo'lgan p harfini bildiradi

p{2,} ikta va undan ortiq bo'lgan p harflaridir.

\$ va ^ belgilari bular qatordagi joylashgan yerini bildiradi. Yani agarda p\$ bo'lsa unda p harfi bilan tugaydigan qatorni bildiradi. Agarda ^p bo'lsa unda boshlanishi p harfi bilan boshlanganligini bildiradi.

. bu belgi yani nuqta har qanday belgini bildiradi, yani istalgan belgi. Agarda biz p.p deb belgilasak, unda bu php, yoki pop, yoki istalgan boshqa harf yoki soni yoki belgini bildiradi.

Endi shularni birlashtirib yana ham mukammalroq qidirish belgisini yozishimiz mumkin.

^{2}\$ - istalgan va faqatgina ikkita belgidir

(.*) - va teglarini orasida bo'lgan har qanday qator yoki qatorlar, belgilar bo'lishi mumkindir.

p(hp)* - p harfi bilan boshlanib, undan keyin esa nol yoki hp so'zi qaytarilib kelishi mumkin. Yani phphp yoki p yoki phphp ham bo'lishi mumkindir.

Yuqorida ko'rib chiqqanlarimizni orasiga yani tayor, shaklga solingan:

[:alpha:] - alifbodagi hamma harflar (aA-zZ)

[:digit:] - raqamlar (0-9)

[:alnum:] - haf va raqamlar

[:space:] - bo'sh joy, space bar

Qatorlar, so'zlar bilan ishlash funktsiyalar.

ereg()

Berilgan qator, so'z ichidan oldindan belgilangan shablonni qidiradi, agarda topsa unda TRUE topa olmasa unda FALSE qaytaradi.

Int erereg(string shablon, string qator, [array topilganlar])

misol1:

```
$email="uzinfo@inbox.ru";
```

```
$if_com=ereg("(\\)(com$)", $email);
```

Bu yerda if_com=FALSE sababi tugashi .com bilan tugagani yo'qdir. Agarda [email=uzinfo@inbox.com](mailto:uzinfo@inbox.com) bo'lganda edi unda TRUE bo'lardi.

Misol2

```
$url="http://www.yahoo.com";
```

```
$www_url=ereg("^http://www\\([[:alnum:]]+\\)([[:alnum:]]+)", $url, $regs);
```

```
if ($www_url):
```

```
echo $regs[0];
```

```
print "<br>";
```

```
echo $regs[1];
```

```
print "<br>";
```

```
echo $regs[2];
```

```
print "<br>";
```

```
echo $regs[3];
```

```
endif;
```

Shunda oynangizga

<http://www.yahoo.com>

<http://www.yahoo.com>

yahoo

com

Shu kabi yozuv chiqadi. Keling birma bir ko'rib chiqaylik nimalar qilganimizni:

1. [^\(http://www\)](http://www). Bu yerda biz boshlanishi <http://www>. bilan boshlanuvchi (\\ -- bu biz nuqtani belgilab berdik)
2. [\(\[\[:alnum:\]\]+\)](#) Bu yerda esa nuqtadan keyin son va harflardan iborat bo'lgan so'z.
3. Va yana \\ bilan tugatib
4. [\(\[\[:alnum:\]\]+\)](#) Bu yerda esa nuqtadan keyin son va harflardan iborat bo'lgan so'z.

Bu yerdan ko'rib turibsizki agarda biz <http://www.uzinfo.php0h.com> qiladigan bo'lsak unda FALSE bo'ladi, sababi bu yerda bizda uchta nuqtamiz bor a yuqoridagi misolda esa biz faqatgina 2 ta nuqta bilan ko'rsatib ketdik.

Mana yuqoridagi misoldan siz ko'rganingizdek biz har hil belgilar ^, + larni ishlatishni ham ko'rib o'tdik.

ereg_replace()

Bu funktsiyamiz qandaydir shablonni (yani so'z, ibora, qator)ni qidiradi va siz bergan boshqa shablon bilan o'zgartiradi.

String `ereg_replace(string shablon, string o'zgartirish shabloni, string qator)`

Mana misolga qarang:

```
$copy_date="Copyright 1999";  
$copy_date=ereg_replace("([0-9]+)","2000", $copy_date);  
print $copy_date;
```

Shunda oynangizda

Copyright 2000

deb chiqadi sababi siz hamma ketma ket kelgan sonlarni 2000 bilan o'zgartirishni berdingiz.

eregi()

Bu ayanan `ereg()` funktsiyasi kabidir, ammo farqi shundaki harflarni katta kichikligiga qaramaydi. Yani A va a `eregi()` funktsiyasi uchun birdir.

Int `eregi(string shablon, string qator, [array topilganlar])`

misol:

```
$password="abc";  
if(!eregi("[[:alnum:]]{8,10}", $password):  
print "Invalid password! Passwords must be from 8 through 10 charactes in length.";  
endif;
```

Bu misolimizda `$password` o'zgaruvchisi ichidan har qanday harf, yoki son yoki boshqa belgilar qidirladi va ularni uzunligi 8-10 orasida bo'lmasa unda "Invalid password! Passwords must be from 8 through 10 charactes in length." yozuvini chiqaradi.

eregi_replace()

`ereg_replace` kabidir, faqat `eregi` kabi katta kichik harflar bu yerda bir hil deb biladi.

split()

Bu funktsiyamiz qatorni bir qancha bo'laklarga bo'lib beradi, oldindan biz aniqlab bergan shart bo'yicha.

Array `split(string shablon, string qator [, int to'htash])`

Misol:

```
$ip="168.192.0.0";  
$ipadr=split("\", $ip)  
print "$ipadr[0] <br>";
```

```
print "$ipadr[1] <br>";  
print "$ipadr[2] <br>";  
print "$ipadr[3] <br>";
```

Buni ishlatib ko'rganingizdan keyin oynangizga quyidagi chiqadi.

```
168  
192  
0  
0
```

Yani ko'rganingizdek, biz bergan shart asosida yani nuqtalardan bo'lib berdi.

spliti()

Aynan split() kabidir faqat farqi shundaki, harflarni katta,kichikligini farqi yoqdir.
Array spliti (string shablon, string qator [, int tugash])

sql_regcase()

Bu qo'shimcha funktsiya siz yozgan qatordagi harflar va belgilarga bittadan hudi shunday belgi qo'shadi, agarda harflar bo'lsa va ularni katta kichigi bo'lsa unda kattasini va kichigini qo'shib oynaga chiqaradi.
String sql_regcase(string qator)

misol:

```
$version="php 4.0";  
print sql_regcase($version);
```

oynamizga [Pp] [Hh] [Pp] [] [44] [..] [00] kabi yozuv chiqadi.

Yana bazi bir komandalarni ham yozib o'taylik.

str_split()

Bu funktsiyamiz yozuvimizni o'zgaruvchanlarga bo'lib tashlaydi.

array str_split(string yozuv [, int yozuv_uzunligi])

misol:

```
<?  
$str="parcha qil";  
$arr1=str_split($str);  
$arr2=str_split($str, 3);  
print_r($arr1);  
print_r($arr2);  
?>
```

Shunda bizning oynamizga quyidagilar chiqadi:

```
Array(
[0]=>p
[1]=>a
[2]=>r
[3]=>c
[4]=>h
[5]=>a
[6]=>
[7]=>q
[8]=>i
[9]=>l
)
Array(
[0]=>par
[1]=>cha
[2]=>qi
[3]=>l
)
```

Agarda sizga qandaydir yozuvni, ma'lumotni qandaydir belgilanga uzunlikda chiqarish kerak bo'lsa unda nima qilishimiz mumkin.

Buning uchun,

`wordwrap()`

Shu funktsiya yordam beradi bizga.

`string wordwrap(string yozuv [, int uzunlik [, string bolish [, boolean cut]]])`

bolish: Bo'lingan joylarga `\n` belgisini qo'yadi yani keyingi qator belgisini, yoki siz belgilab bergan belgini uzunlik: bir qatorda qancha belgi bo'lishini bildiradi

cut: agarda bu yerga 1 raqamini qo'ysak unda qancha uzunlik bergan bo'lsangiz va so'zimiz uzun bo'lib sigmasa ham so'zni bo'ladi.

Misol:

```
<?
$str="Bu yerda yana bir tekst";
$mod_str=wordwrap($str, 10,"<br>");
echo ($mod_str);
?>
```

Shunda oynaga

Bu yerda

yana bir
tekst

chiqadi.
agarda

```
<?
$str="Bu yerdaaaaaaa yana bir tekst";
$mod_str=wordwrap($str, 10,"<br>", 1);
echo ($mod_str);
?>
```

Unda oynamizga
Bu
yerdaaaaaaa
a yana bir
tekst

chiqadi, ko'rib turganingizdek shunday farqlari bor.

Qanday qilib biz berilgan yozuvdan kerakli raqamdagi harfni oynaga chiqarishimiz mumkin?
Siz aytishingiz mumkin yuqorida substr, yoki boshqa komandalar bilan qilishni.
Ammo bundan ham yanayam qulay yo'li bordir.

```
<?
$str="Tester is here";
echo $str[0];      //"T"
echo $str[4];      //"t"
?>
```

Shu orada yana bir kichkina funktsiyani yozib o'tish kerak.
Siz bilasiz kirilcha, yani rus yozuvlari bir qancha kodirovkada yozilishi mukindir, ularni o'zini orasida bir
biridan o'tkazish uchun sizga

convert_cyr_string()
Funktsiyasi yordam beradi.
string (convert_cyr_string(string str, string from, string to)
Bu yerda from bu qaysi kodirovkadan, va to qaysi kodirovkaga o'girish.
bu yerda siz from va to ga quyidagilarni yozishingiz mumkin
k - KOI8-R
w - Windows1251
i - ISO8859-5
a - X-CP866
m - X-Mac-Cyrillic

Misol:

```
<?
$str=" , , , , , ";
echo convert_cyr_string ($str, "w", "k")
?>
```

Mana bu yerda siz Windows1251 formatidan KOI8-R formatiga o'girdingiz.

Perl kabi yozish!!!

Agarda bilsangiz PERL tekstlar bilan ishlashda eng qulay va engi kuchli tildir, eng qiyin hollarda ham u tekstlarga ishlov bera oladi. Shuning uchun PHPda ham PERLda yozilgan yani shu kabi ko'nlarni ishlatga bo'ladi, yani buning qilinganligini sababi shundaki, yangittan yangi narsalarni o'ylab topmasdan, tayor bor bo'lganini ishlatishdir.

Keling misolga qaraylik:

/food/ bu yerda food so'zi ikkita qiyshiq chiziqqa olingandir
/fo+/ bu yerda esa fo bilan boshlanib qolganlari esa ahamiyati yo'qdir yani fo dan so'ng har qanday belgi bo'lishi mumkindir. Misol: fool, fo4, food va hkz.

Perl'da metasybollar ishlatiladi yani belgiga belgi qo'shilgan holi. Yani qandaydir belgidan oldin \ belgisini qo'ysak unda uni ma'nosi o'zgaradi. Misol uchun faqatgina "d" harfi bu "d" harfini bildiradi, agarda u "d" bo'lsa unda bu raqamlarni bildiradi.

/([d]+)0000/

\w belgisi esa harflarni bildiradi.

/<([w]+)>/ bu degani <> beglilari orasida harflar bor, yani HTML tagi kabidir, <body> yoki boshqalaridan , <center> ham bo'lishi mumkindir.

\b bu belgi begilar chegarasini bildiradi. Yani tugashidir.

/sa\b/ Bu degani salsa, lisa bo'lishi mumkin lekin sand so'zi bo'la olmaydi.

\B yuqoridagi yani \b teskarisidir. Yani boshlanishidir.

/sa\B/ so'zlar sand, sally, lekin salsa emas chunki tugashi ham sa bilan tugaydi.

Perl kabi ishlaydigan komandalar PHP'da

preg_match()

Qatordan shablona to'gri keluvchisi bor yo'qligini qidiradi, agarda bor bo'lsa TRUE agarda yo'q bo'lsa FALSE qaytaradi.

Int preg_match(string shablon, string qator [, array topilganlar])

misol:

```
$line="Vi is the greatest word processor ever created!";  
if(preg_match("\bVi\b\i/", $line, $match)):  
print "Match found!";  
print $match[0];  
print $match[1];  
endif;
```

Shunda oynangizga Vi va i harflari chiqadi yani Vi butun qidirish edi va i bilan tugashi kerak edi. Va topilganligi uchun if funktisiamiz ishladi va oynaga istagan yozuvlarimiz chiqdi.

preg_match_all()

Berilgan qatorda hamma topilganlarni qanaydir o'zgarmasga yozadi.

Int preg_match_all(string shablon, string qator, array topilganlar [, int tartiblash])

tartiblash:

PREG_PATTERN_ORDER-agarda tartiblashda boshqasini belgilamasngiz unda shu bo'ladi. Tartiblash ketma ket bo'ladi, yani avval 0 va uni ichidagi indexlarda topilganlarni butunasiga yoziladi. Va boshlanishi 1 va uni indexlarida esa faqatgina aynan topilganlarni o'zigina, yani hammasi emasdir. Pastda yana misol bilan ko'rsatib o'tiladi.

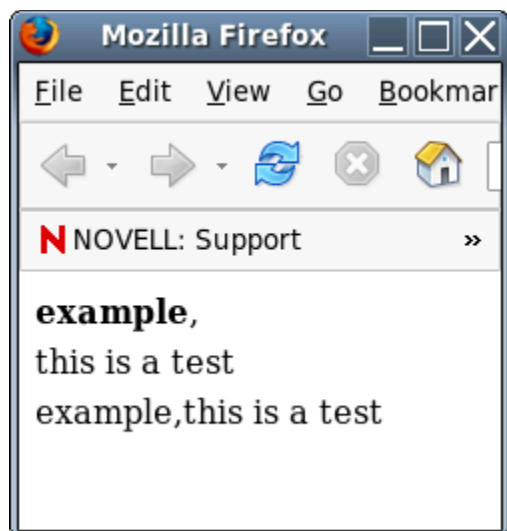
PREG_SET_ORDER- Bu yerda esa 0 ning indexlarida topilganlarni hammasi (yani butun), va undan so'ng esa topilganni o'zigina, 1 va uni indexlarida esa ikkinchi topilganlar va uni o'zigina.

Keling misolda ko'raylik, sababi buni so'zlar bilan aniq tushuntirib berish ancha qiyin, misolda hammasini yaqqol ko'rasiz.

PREG_PATTERN_ORDER

<?

```
preg_match_all("\<[^\>]+\>(.*)\<[^\>]+\>\|U", "<b>example</b><div align=left>this is a  
test</div>", $out, PREG_PATTERN_ORDER);  
print $out[0][0].", ".$out[0][1]."\n";  
print $out[1][0].", ".$out[1][1]."\n";  
?>
```

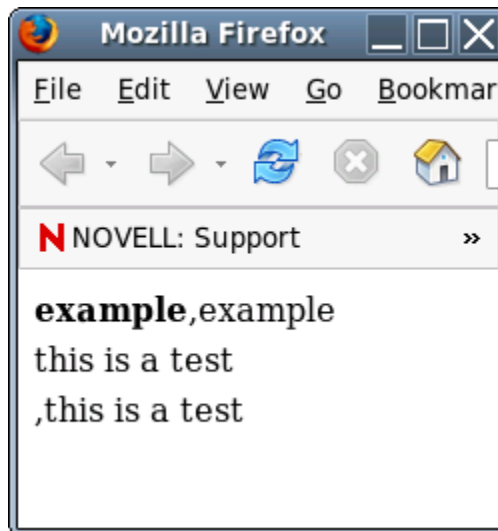


Mana ko'rib turganingizdek bu yerda bizda ikki hil oynaga chiqish, yani 1 chisida html ko'dlari ishladi, ikkinchisda esa yo'q yani html ko'dining ichidagi ma'lumotgina chiqdi.

Endi ikkinchisini ko'raylik.

<?

```
preg_match_all("|<[^>]+>(.*?)<[^>]+>|U", "<b>example</b><div align=left>this is a  
test</div>", $out, PREG_SET_ORDER);  
print $out[0][0].", ".$out[0][1]."\n";  
print $out[1][0].", ".$out[1][1]."\n";  
?>
```



Mana ko'rganingizdek bu yerda avval ikkalasi ham chiqadi, so'ng keyingi topilganlari chiqa boshlaydi.

`preg_replace()`

Aynan `ereg_replace()` kabi ishlaydi, faqat PERL qidirish/almashtirishdir.

Mixed `preg_replace(mixed shablon, mixed almashtirish, mixed qator [, int qanchasi_ozgaradi])`

`preg_split()`

`split()` funktsiyasi kabidir. Shablonda esa perl yozuvi bo'lishi ham mumkindir.

Array `preg_split (string shablon, string qator, [, int qanchasi [, int flag]])`

flag:

Flaglar bir biri bilan birga ham ishlatilinishi mumkindir.

`PREG_SPLIT_NO_EMPTY` – Agar bu flag bo'lsa unda faqatgina bo'sh bo'lmagan bo'limlarga qaytadi.

`PREG_SPLIT_DELIM_CAPTURE` – Bu flag agarda bo'lsa unda qavs ichidagi ma'lumot qaytadi.

`PREG_SPLIT_OFFSET_CAPTURE` – har bir topilgan ham qaytadi.

Misol:


```
$user_info="+wj+++Gilmore+++++wjgi]more@hotmail.com++++++Columbus+++OH";
$fields=preg_split("/^+{1,}/", $user_info);
while ($x<sizeof($fields)):
print $fields[$x]."<br>";
$x++;
endwhile;
```

Shunda oynamizga

```
wj
Gilmore
wjgi]more@hotmail.com
Columbus
OH
```

shu kabi ma'lumot chiqadi. Chunki biz + orasidagi har bir so'zni bo'lib oldik. Va oynaga birin ketin chiqardik.

preg_grep()

Buttun qatorni, masivni qarab chiqadi, va shablonga mos kelganlarni esa qaytaradi.

Array preg_grep(string shablon, array masiv)

misol:

```
$foods=array("pasta", "steak", "fish", "potatoes");
$p_foods=preg_grep("/p(w+)/", $foods);
$x=0;
while ($x<sizeof($p_foods)):
print $p_foods[$x]. "<br>";
$x++;
endwhile;
```

shunda oynamizga

```
pasta
potatoes
```

chiqadi, ko'rib tuganingizdek, p bilan boshlangan hamma qatorlarni qaytardi.

chop()

Bu finktsiyamiz qatorimizning so'ngi qismidan tugash va yangi qator belgilarini olib tashlaydi.

String chop(string qator)

misol:

```
$header="Table of Contents \n\n";
$header=chop($header);
```

shunda header="Table of Contents" bo'ladi, yani ohiridagi \n\n yangi qator belgilar olib tashlanadi.

str_pad()

str_pad() funksiyasi qandaydir qatorni uzunligini uzaytirib uzaytirish uzunligini bo'sh joy bilan yoki oldindan belgilangan belgi bilan to'ldiradi.

String str_pad(string qator, int uzunlik [, string to'ldirish_belgisi [, int to'ldirish_turi]])

to'ldirish_turi:

STR_PAD_RIGHT – to'ldirish o'ng tarafdin bo'ladi

STR_PAD_LEFT – to'ldirish chap tarafdin bo'ladi.

STR_PAD_BOTH – to'ldirish ikki tarafdin bo'ladi.

Keling misolda ko'raylik.

```
$food = "salad";
```

```
print str_pad($food, 5); //bu yerda salad so'zi chiqadi. Yani bo'sh joy bilan birga
```

keyingi misolga qarang.

```
$header="Table of Contents";
```

```
print str_pad($header, 27, "====", STR_PAD_BOTH); //oynamizga "==== Table of Contents  
//====" chiqadi
```

trim()

Bu funktsiya qatorni bo'sh joylardan va (\n, \r, \t, \v , \0) belgilaridan tozalaydi.

String trim(string qator)

strlen()

Qatorni uzunligini shu funktsiyamiz orqali bilish mumkindir.

Int strlen(string qator)

misol:

```
$string="hello";
```

```
$length= strlen($string); //length=5 bo'ladi.
```

strcmp()

Ikta qatorni bir biri bilan solishtiradi, bu yerda harflarni katta kichikligi ahamiyatga egadir.

Int strcmp(string qator1, string qator2)

Solishtirganda qaytgan sonlardan quydagini aniqlasa bo'ladi.

- 0 soni qaytadi agarda ikkalasi bir biriga teng bo'lsa
- <0 bo'ladi agarda qator1<qator2 dan bo'lsa
- >0 bo'ladi agarda qator1>qator2 dan bo'lsa

Misol:

```
$str1="butter";
```

```
$str2="butter";  
if((strcmp($str1, $str2))==0):  
print "Strings are equal";  
endif;
```

Bu misolda ikkita qator bir biri bilan solishtiriladi va agarda bir biriga teng bo'lsa unda "Strings are equal" degan yozuv chiqadi.

strcasecmp()

Aynan strcmp() kabidir faqatgina farqi shundaki bu yerda harflarni katta kichikligi hisobga olinmaydi.

Int strcasecmp(string qator1, string qator2)

strspn()

Bu funktsiyamiz qator1 dagi belgilarni qator2 da qanchasi borligini beradi.

Int strspn(string qator1, string qator2)

misol:

```
$password="12345";  
if(strspn($password, "1234567890") == strlen($password)):  
print "Passowrd cannot consist solely of numbers!";  
endif;
```

Bu yerda avval password "1234567890" bilan solishtiriladi, va shuning ichida qancha belgi borligini aniqlaydi yani bizda 5 ta belgi to'g'ri keladi. Keyin esa password'ning uzunligi o'lchanadi u ham 5 dir, yani ikkalasi ham bir biriga teng, demakki password faqatgina sonlardan iborat ekan, shuning uchun bizda oynamizga "Password cannot consist solely of numbers" degan yozuv chiqadi.

strcspn()

Bu funktsiyamiz, qator1 dagi belgilarni qanchaligini qaytaradi, qaysilari agarda qator2 da bo'lmasa.

Int strcspn(string qator1, qator2)

Misol:

```
$password="12345";  
if(strcspn($password, "1234567890")==0):  
print "Passowrd cannot consist solely of numbers!";  
endif;
```

Bu misolimiz yuqoridagi misol bilan bir hil ish qiladi, yani tekshirishda hamma belgilar qator2 da bo'lganligi uchun 0 qaytaradi va biz ham 0=0 ligini tekshirib to'g'ri ekanligidan shu yozuvni oynamizda ko'ramiz.

strtok()

Qatorni bo'laklarga bo'lib tashlaydi, yani bo'lish belgisi ko'rsatilishi kerak.

PHP Davrasida kitobi Open Source

String strtok(string qator, string bo'lish_belgisi)

Bu funktsiyamizning bir qo'shimchasi bor, yani funktsiyamiz butu qatorni bo'lishi uchun uni biz qayta qayta chaqirishimiz kerakdir, har gal chaqirganimzda bitta bitta bo'lib keyingiziga o'tadi.

Misolda ko'raylik:

```
$info="Linux is the best:linuxforum.ru | Linux, Copyrighted";  
//Bo'lish belgilarimiz :, |, , belgilaridir  
$tokens=":|,";  
$tokenized=strtok($info, $tokens);  
while($tokenized):  
echo "Element = $tokenized <br>";  
tokenized=strtok($tokens);           //Bu yerda biz qaytadan qatorni berishimiz shart emas  
endwhile;
```

Shunda oynamizga

Element = Linux is the best

Element = linuxforum.ru

Element = Linux

Element = Copyrighted

kabi ma'lumot chiqadi.

parse_str()

Bu funktsiyamiz qatordan o'zgaruvchi va uning qiymatini ajratib oladi. Buni misolda ko'ryalik.

Void parse_str(string qator)

misol:

```
$url="fname=wj&lname=gilmore&zip=43210";  
parse_str($url)  
//shunda bizda $fname=wj, $lname=gilmore, $zip=43210 bo'ladi  
//ekranga chiqarib ko'raylik.  
print $fname;  
print "<br>";  
print $lname;  
print "<br>";  
print $zip;
```

oynaga:

wj

gilmore

43210

yozuvi chiqadi.

explode()

Qatorni bir qancha bo'limlarga bo'lib tashlaydi.

Array explode(string belgi, string qator [, int tugash_yeri])

Bu yerda tugash yerini belgilasangiz qatordan shuncha bo'linishdan keyin tugaydi, agarda bermasangiz unda qatorni hammasini bo'lib qaytaradi.

Misol:

```
$info="wilson | baseball | indians";  
$user=explode("|", $info);  
print "$user[0] <br>";  
print "$user[1] <br>";  
print $user[2];
```

shunda oynamizda

wilson

baseball

indians

yozuvi chiqadi.

implode()

Bu funktsiya explode() ning teskarisidir, yani birlashtiradi.

String implode (string belgi, array qatorlar)

misol:

```
$uzb=array ("Tashkent", "Samarkand", "Bukhara");  
$all_in_one=implode("|", $uzb);  
//shunda all_in_one="Tashkent | Samarkand | Bukhara" bo'ladi.
```

strpos()

strpos() funktsiyasi qatordan qandaydir belgini qaysi o'rinda turganligini topish uchun ishlatiladi.

Int strpos(string qator, string belgi, [int boshlash_joyi])

Boshlash_joyi bu qatorni qaysi joyidan qidirishni boshlash kerak ekanligini ko'rsatadi.

Misol:

```
$log="  
206.169.23.11:/www/:2000-08-10  
206.169.23.11:/www/logs/:2000-02-04  
206.169.23.11:/www/img/:1999-01-31";  
$pos=strpos($log, "1999");  
print $pos;
```

shunda oynamizga birinchi uchragan 1999 ni qaysi joydaligini qaytaradi, yani bizning misolda 92 belgidir.

strrpos()

Funktsiyasi qatordan faqatgina bitta belgini qidiradi, va ohirgi belgini topadi.

Int strrpos(string qator, char belgi)

str_replace()

Bu funktsiyamiz, qatorni ichidan bizga kerak bo'lgan qismini qidiradi va uni biz belgilab bergan boshqasi bilan almashtiradi.

String str_replace(string qidirish_qator, string almashtirish_qator, string qator)

misol:

```
$favorite_food="My favorite foods are ice cream and chicken wings";  
$favorite_food=str_replace("chicken wings", "pizza", $favorite_food);  
//shunda favorite_food="My favorite foods are ice cream and pizza" bo'ladi.
```

strstr()

Bu funktsiyamiz qatordan qidirilgan birinchi belgini topadi va undan so'ng kelgan hamma belgilarni qaytaradi.

String strstr(string qator, string qisirish_qatori)

misol:

```
$url="http://www.uzinfo.php0h.com";  
$domain=strstr($url, ".");  
//shunda domain=.uzinfo.php0h.com bo'ladi.
```

substr()

Bu bishlangich joyini belgilab berilagan va uzunligini belgilangan joygacha bo'lgan hamma belgilarni qatordan o'qib qaytaradi.

String substr(string qator, int boshlanish_joyi [int uzunligi])

misol:

```
$car="1994 Ford";  
$model=substr($car, 6);  
//bu yerda biz 6 belgidan boshlab to ohirigacha bo'lgan hamma belgilarni oldik. $model=Ford  
  
$model=substr($car, 0, 4);  
//bu yerda esa biz yilin kesib oldik, yani birinchi belgimiz bizda 0 belgidir, va undan boshlab 4 ta belgi.
```

Agarda uzunligini (-) belgisi bilan beradigan bo'lsak unda qatorni uzunligi – boshlanish_joyi – uzunligi yani misolga qarang

```
$model=substr($car, 2, -5);  
//model="44"  
$model=substr($car, 2, -3);  
//model="44 F"  
$model=substr($car, 2, -1);  
//model="44 For"
```

Mana ko'rib turgan misollardan ko'rinshingiz mumkindir qanday hisob kitob qilinishi. Yani uzunligi 9 boshlanishi 2 yani $9-2=7$ va undan uzunliklarni ayirib tashlaymiz.

substr_count()

Qatorda qanchasi bergan qidirish_qatoriga mos kelishini beradi.

Int substr_count(string qator, string qidirish_qatori)

misol:

```
$hat="When the rain begins to fall.";  
$count=substr_count($hat, "in");  
//$count=2 bo'ladi.
```

substr_replace()

Bu boshlanish joyi belgilangan va uzunligi beglilangancha qator qismini almashtiradi.

String substr_replace(string qator, string almashtirish, int boshlanish_joyi [, int uzunligi])

Ishlash mantig'i aynan substr kabidir, yani uzunlik berilmasa qator ohirigacha almashtiradi va hkz.

Misol:

```
$hayot=" ni sevaman";  
$ism="Onam";  
$hayot=substr_replace($hayot, $name, 0, 0);  
print $hayot;  
//shunda oynamizga "Onamni sevaman" yozuvi chiqadi
```

Tekstni HTML formatida ekranga chiqarish

nl2br()

\n belgisini yani yangi qator belgisini
 belgisi bilan almashtiradi.

String nl2br(string qator)

misol:

```
$test_recipe="  
Party Sauce recipe:
```

```
1 can stewed tomatoes
3 tablespoons fresh lemon juice
Stir together, server cold.”;
$html_recipe=nl2br($test_recipe);
print $test_recipe;
print $html_recipe;
//shunda ko'rasizki birinchi print'da hamma yozuv bir qatorga chiqdi
//ikkinchi printdan bo'lsa esa hammasi qanday bo'lsa shunday chiqadi. Pastga qarang.
```

Party Sauce recipe: 1 can stewed tomatoes 3 tablespoons fresh lemon juice Stir together, server cold.

Party Sauce recipe:

1 can stewed tomatoes

3 tablespoons fresh lemon juice

Stir together, server cold.

Mana shu kabi.

htmlspecialchars()

Bu funktsiyamiz belgilarni html yozish formatiga o'giradi, yani < belgisi “<” yoki > “>” kabidir.

String htmlspecialchars(string qator)

htmlspecialchars()

yuqoridagi funktsiya kabidir, faqat bu funktsiyada bir qancha qo'shimcha belgilarni taniydi.

String htmlspecialchars(string qator)

taniydigan belgilari:

| | | |
|---|---|--------------------------|
| & | - | & belgisiga aylanadi |
| “ | - | " ga aylanadi |
| ' | - | ' ga aylanadi |
| < | - | < ga aylanadi |
| > | - | > ga aylanadi |

misol:

```
$user_in="I just can't get <you> of!";
```

```
$conv_in=htmlspecialchars($user_in);
```

```
//shunda conv_in="I just can &#039; get &lt; you &gt; of!" bo'ladi.
```

strip_tags()

Funktsiyamiz qatorda bor bo'lgan hamma PHP va HTML taglarini olib tashlaydi, va faqatgina tekstni qoldiradi.

String strip_tags(string qator [, string mumkin_bolgan_teglar])

misol:

```
$user_input="I just <B> love</b> PHP !!!";
```

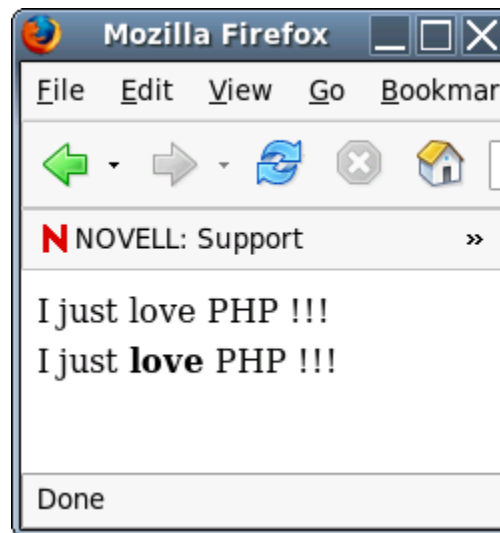
```
$strip=strip_tags($user_input);
```

```
print $strip;
```

```
print "<br> $user_input";
```

```
//bu yerda biz birinchi bo'lib strip_tags dan chiqqan tekstni oynaga chiqaramiz, va ikkinchisida esa  
//qanday bo'lsa shunday chiqardi.
```

Oynangizda quyidagicha chiqishi kerak.



Mana ko'rib tuganingizdek chiqadi.

get_meta_tags()

Bu funktsiyamiz HTML sahifasida yozilgan META taglarni o'qib olib beradi.

```
Array get_meta_tags(string fayl_ismi/URL [, int yolni_yoqis]);
```

Agarda sizda sahifangizda meta taglar bo'lsa yoki boshqa sahifada bo'lsa ularni o'qib olish niyatingiz bo'lsa unda shu funktsiyadan foydalanasiz.

Misol:

```
...  
<meta name="keywords" content="PHP, HTML, ...">  
...
```

va shuni o'qish uchun

```
$metas=get_meta_tags("test.html");
```

```
//shunda $meta["keywords"]="PHP, HTML, ..." kabi bo'ladi yani metas ning ichida keywords ochiladi
```

```
//va unga contentdagi ma'lumot yoziladi
```

strtolower()

Hamma harflarni kattadan kichikga o'giradi.

```
String strtolower(string qator)
```

misol:

```
$sharf="HAYOTDA hamma narsa YAHSHI!";  
$sharf=strtolower($sharf);  
print $sharf;  
//shunda o'ynamizda "hayotda hamma narsa yahshi!" yozuvi chiqadi.
```

strtoupper()

strtolower() funksiyasini teskarisidir. Yani kichik harfdan katta harfga o'giradi.
String strtoupper(string qator)

ucfirst()

Qatorni bosh harfini katta harfga o'giradi, yani birinchi harfni.
String ucfirst(string qator)

misol:

```
$sharf="hayot qanchalik Go'zal";  
$sharf=ucfirst($sharf);
```

```
//shunda $sharf="Hayot qanchalik Go'zal" ga teng bo'ladi.
```

ucwords()

Bu har bir so'zni bosh so'zini katta harfga o'giradi.
String ucwords(string qator)

misol:

```
$sharf="hayot qanchalik Go'zal";  
$sharf=ucwords($sharf);
```

```
//shunda $sharf="Hayot Qanchalik Go'zal" bo'ladi.
```

Mana shu bilan bu katta mavzuni ham yopamiz, va keyingi bundan yana ham qiziqarli mavzularga o'tamiz.

PHP va Web Sahifalarni Dinamik shakilda yaratish.

Sahifa shablonini yaratish.

Yani bu yerda biz oldindan qandaydir sahifani shablonini yaratamiz, va keyinchalik shuni ishlatgan holda har hil informatsiyalarni shu shablon orqali chiqaramiz.

include()

Faylimiz ichiga boshqa faylni qo'shish, biz bu funktsiyani yuqoridagi mavzularda ham ko'rib o'tganmiz.

include_once()

Aynan include kabidir, ammo farqi shundaki agarda programmamiznig qaysidir joyida biz oldindan qoshgan bo'lsak shu faylni unda qayta qo'shmaydi.

require()

Aynan include kabidir, faqat farqi shundaki agarda programmizda qandaydir tekshirish (if) bo'lsa va agarda tekshirishning ichida to'g'ri bo'lganda qo'shgin desak, va tekshirish noto'g'ri bo'lsa ham faylni qo'shadi.

Misol:

```
if (some_condition):  
    require('test.php');  
else:  
    print "No result";  
endif;
```

bu yerda some_condition FALSE bo'lsa ham test.php faylimizga qo'shiladi.

Boshqa bir misol:

user.tpl fayli

```
<?  
$site_name="PHP tutor";  
$email="uzinfo@inbox.ru";  
$name="Sinch";  
?>
```

index.php fayli

```
<? require ('user.tpl') ?>  
<HTML>  
<HEAD>  
<title><?print $site_name?></title>  
</HEAD>
```

PHP Davrasida kitobi Open Source

```
<body>
<?
print "Welcome to $site_name. For some questions sent to <a href=\"mailto:$email\">$name</a>";
?>
</body>
</HTML>
```

shunda user.tpl'dan ma'lumot o'qiladi va sahifa ichida ulardan foydalaniladi.

require_once()

Aynan include_once kabidir farqi require va include orasidagi farqidir.

Keling bir kichik misolda shularni hammasini ko'rib chiqaylik.

Fayl site.tpl

```
<?
$site_name="PHP Darslari";
$site_email="uzinfo@inbox.ru";
$site_path="http://uzinfo.php0h.com";
function show_header($site_name){
?>
<html>
<head>
<title><? print $site_name ?> </title>
</head>
<body>
Bu sahifa boshidir.
<hr>
<?
}
function show_footer(){
?>
<hr>
Bu tugashi
</body>
</html>
<?
}
?>
```

fayl test.php

```
<?
include ("site.tpl");
show_header($site_name);
show_footer();
?>
```

Va agarda shuni ishlatsangiz oynangizda quyidagicha sahifa ochilishi kerakdir.

Bu sahifa boshidir

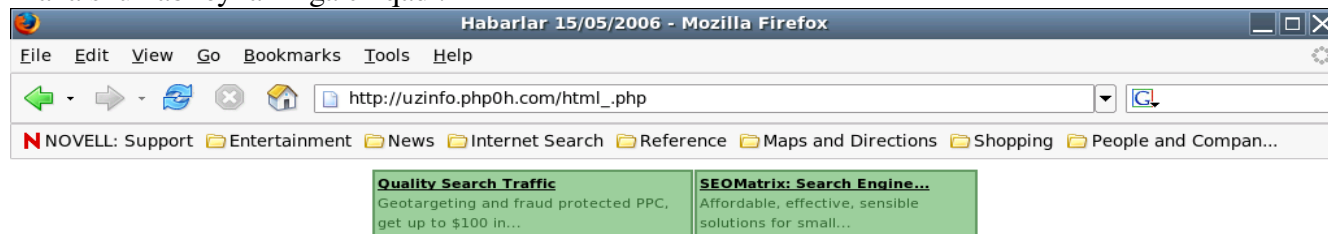
Bu tugashi

Mana shu kabi siz oldindan shablonlarni tayorlab olib keyinchalik o'zgartirish orqali oynaga sahifalarni chiqarishingiz mumkindir.

Agarda har kuni sahifa o'zgarishi kerak bo'lsa misol uchun yangiliklarni shunda siz faqatgina shablon yasaysiz va qolgan ma'lumotni esa siz fayldan o'qib olishingiz mumkindir.

Keling mavzuni yopishdan oldin kichik bir proekt qilaylik, yani yangiliklar sahifasini yarataylik. Va bizning keyinchalik qiladigan ishimiz faqatgina bitta faylni ichiga ma'lumotni to'ldirish bo'lsin qolgan ishni esa php programmasi qilsin.

Mana shu kabi oynamizga chiqadi.



Bugun OSB uchrashuvi bo'lib o'tdi

Bugun halqaro uchrashuv bo'lib o'tdi.

u Yerda Kofe va Choy Pele bo'ldi. Bu hayotga havfli insonni tutib kelgan inson Marodingo ekan.

Ho'sh bo'lib o'tgan habarlarni kechki habarlardan koring.

hayir sog' bo'ling.

Ma'lumot news.dreams.com olingan

Avtor Baqlon Opa

Done

PHP Davrasida kitobi Open Source

Buning ko'di esa

fayl site.tpl

```
<?
```

```
$yangilik_boshi="Bugun OSB uchrashuvi bo'lib o'tdi";
```

```
$kun="15/05/2006";
```

```
$butun_yozuv="Bugun halqaro uchrashuv bo'lib o'tdi.
```

```
u Yerda Ko'fe va Choy Pele bo'ldi. Bu hayotga havfli insonni tutib kelgan inson Marodingo ekan.
```

```
Ho'sh bo'lib o'tgan habarlarni kechki habarlardan koring.
```

```
hayir sog' bo'ling.";
```

```
$qayerdan="news.dreams.com";
```

```
$savor="Baqlon Opa";
```

```
?>
```

fayl news.php

```
<?include ("site.tpl") ?>
```

```
<html>
```

```
<head>
```

```
<title><? print "Habarlar $kun"; ?> </title>
```

```
</head>
```

```
<body>
```

```
<H1><? print $yangilik_boshi; ?> </H1>
```

```
<CENTER>
```

```
<?
```

```
print nl2br($butun_yozuv);
```

```
?>
```

```
<BR>
```

```
<HR>
```

```
<h6> <? print "Ma'lumot $qayerdan olingan"; ?> </h6>
```

```
<h6> <? print "Avtor $savor"; ?> </h6>
```

```
</CENTER>
```

```
</body>
```

```
</html>
```

Siz hohlasangiz yana ham takomillashtirishingiz mumkindir.

Bu juda ham oddir bir yangiliklar sahifasidir. Bizda endi ishimiz osonlashdi, yani siz faqatgina bitta sahifaga va aniq qaysi yerga nima yozsangiz qayrda nima chiqishini bilasiz. Ishimiz ancha qulaylashdi.

Formlar va ular bilan ishlash.

Formlarni siz HTML o'rgangan vaqtingizda tamoman ko'rib chiqgansiz, ammo u vaqtda siz ularni tamoman ishlata olmagansiz, chunki ishlatish uchun sizga PHP yoki PERL yoki Java kabi programmalash tillari kerak edi. Mana endi ularni ishlatishni ko'ramiz.

Keling avval bir misolni ko'raylik.

Oddiy bir html sahifa yozing bu sahida form elementlarini ishlatamiz.

test.html

```
<html>
<head>
<title>Test</title>
</head>
<body>
<form action="getme.php" method="post">
Name:
<input type="text" name="name" size="20" maxlength="20" value=""><br>
Email:
<input type="text" name="email" size="20" maxlength="20" value=""><br>
<input type="submit" value="GO"><br>
</form>
</body>
</html>
```

getme.php

```
<html>
<head><title>Getme.PHP</title> </head>
<body>
<?
print "Sizning ismingiz $name , emailingiz $email";
?>
</body>
</html>
```

shunda sizning oynangizda GO tugmasini bosganingizdan so'ng "Sizning ismingiz Sinch, emailingiz uzinfo@php0h.com" kabi bir yozuv chiqadi.

Ko'rib turganingizdek formalar bilan ishlash juda oson. Faqatgina action'ga qaysi fayl ma'lumotni qayta ishlashi kerak bo'lsa shuni yozasiz, va yoki post yoki get bilan yuborasiz.

Avtomatik ravishda email yuborish

Agarda siz sahifadan email yuborish niyatingiz bo'lsa yani administratorga unda siz mailto: funksiyasini ishlatishingiz mumkindir, ammo bu funktsiyani bir kamchiligi bordir. Kamchiligi shundaki hamma kompyuterlarda ham email klient o'rnatilgan bo'lmasligi ham mumkin. Unda hatingizni siz Web Browser orqali qo'lingizda yozib yuborasiz. PHP sizga bu ishni yana ham osonlashtiradi, yani sahifangizdan to'g'ridan to'g'ri yuboradigan qilishingiz mumkin.

Keling shuni ko'rib chiqaylik.

mail()

Mana shu funktsiya bizga yordam beradi.

Boolean mail(string yuborilish_adresi, string mavzu, string hat, [, string qo'shimcha_mavzu])

keling misolga:

```
$email= "uzinfo@inbox.ru";  
$subject= "Test";  
$message= "Bu bizning sahifadan hat yuborishimiz";  
$headers= "From: uzinfo.php0h.com";  
mail($email, $subject, $message, $headers);
```

endi keling formalarni ishlatgan holda ko'rib chiqaylik:

```
<html>  
<head>  
<title>Email yubor</title>  
</head>  
<body>  
<?  
$form= " <form action=\"email.php\" method=\"post\">  
<input type=\"hidden\" name=\"seenform\" value=\"y\">  
<b> Send us comments </b><br>  
Your Name:<br>  
<input type=\"text\" name=\"name\" size=\"20\" maxlength=\"20\" value=\"\"> <br>  
Your email:<br>  
<input type=\"text\" name=\"email\" size=\"20\" maxlength=\"40\" value=\"\"> <br>  
Your comments: <br>  
<textarea name=\"comments\" rows=\"3\" cols=\"30\"></textarea> <br>  
<input type=\"submit\" value=\"submit!\"> <br>  
</form>";  
if ($seenform!="y"):  
print "$form"; else:  
$recipient= "uzinfo@inbox.ru";  
$subject= "From my site";  
mail($recipient, $subject, $comments, $headers) or die ("Could not sent mail!");  
print "Thank you $name for taking a moment to send us your comments!";  
endif;
```



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

Mana ko'rganingizdek juda ham oson va qulaydir.

Ha yana shuni aytib o'tish kerrakki agarda sizning hosteringiz mail funksiyasini ishlatishga ruhsat bermagan bo'lsa unda siz bu funktsiyani ishlata olmaysiz.

Siz shu funktsiyalar va oldin ko'rib chiqqan funktsiyalar bilan yahshi bir "Guest Book" yoki shu kabi PHP da yaratish mumkin bo'lgan hamma programmalarini yozishingiz mumkindir.

Keling Guest Book yozaylik.

Mana men kichik guest bookni kodlarini beraman. Keyin o'zingizga qulay shakilda o'zgartisihingiz mumkindir.

guest.inc fayli

```
<?
$page_title="Sinchs Guest";
$bg_color="white";
$font_face="Arial, Verdana, Times New Roman";
$font_color="black";
$guest_file="comments.txt";
function view_guest($guest_file){
GLOBAL $font_color, $font_face;
print "Return to <a href=\"guest.php\">Guest</a>, <br><br>";
if (filesize($guest_file)>0):
$fh=fopen($guest_file, "r") or die ("Could not open $guest_file");
print "<table border=1 cellpadding=2 cellspacing=0 width=\"600\">";
while (!feof($fh)):
$line =fgets($fh,4096);
list($date, $name, $email, $comments)=explode("|", $line);
if ($name!=""):
print "<tr>";
print "<td><font color=\""$font_color\"" face=\""$font_face\""> Date:</font></td>";
print "<td><font color=\""$font_color\"" face=\""$font_face\""> $date</font></td>";
print "</tr>";

print "<tr>";
print "<td><font color=\""$font_color\"" face=\""$font_face\""> Name:</font></td>";
print "<td><font color=\""$font_color\"" face=\""$font_face\""> $name</font></td>";
print "</tr>";
```

PHP Davrasida kitobi Open Source

```
print "<tr>";
print "<td><font color=\"\$font_color\" face=\"\$font_face\"> Email:</font></td>";
print "<td><font color=\"\$font_color\" face=\"\$font_face\"> $email</font></td>";
print "</tr>";

print "<tr>";
print "<td valign=\"top\"><font color=\"\$font_color\" face=\"\$font_face\"> Message:</font></td>";
print "<td><font color=\"\$font_color\" face=\"\$font_face\"> $comments</font></td>";
print "</tr>";
print "<tr><td colspan=\"2\">&nbsp;</td></tr>";
endif;
endwhile;
print "</table>";
fclose($fh);
else:
print "<h3> Hech qanday yozuv yo'q</h3>";
endif;
}

function add_guest($name, $email, $comments){
GLOBAL $post_date, $guest_file;
$post_date=date("F j, Y, g:i a");
$content=" $post_date | $name | $email | $comments \n";
$fh=fopen($guest_file, "a") or die("Could not open $guest_file!");
$wr=fwrite($fh, $content) or die("Could not write $guest_file!");
fclose($fh);
}
?>
```

add_guest.php fayli

```
<html>
<? include("guest.inc");
?>
<head>
<title><?=$page_title; ?></title>
</head>
<body>
<?
if (!$seenform):
?>
```

PHP Davrasida kitobi Open Source

```
<form action="add_guest.php" method="post">
<input type="hidden" name="seenform" value="y">
Ism: <br>
<input type="text" name="name" size="15" maxlength="30" value=""> <br>
Email: <br>
<input type="text" name="email" size="15" maxlength="35" value=""> <br>
Fikrlar: <br>
<textarea name="comment" rows="3" cols="40"></textarea> <br>

<input type="submit" value="Yubor">
</form>

<?
else:
add_guest($name, $email, $comment);
print "<h3> Siznig fikringiz qo'shib qo'yildi <a href=\"guest.php\">Orqaga qaytish</a> </h3>";
endif;
?>

</body>
</html>
```

view_guest.php fayli

```
<html>
<? include("guest.inc");
?>
<head>
<title><?=$page_title; ?></title>
</head>
<body bgcolor="<?=$bg_color;?>">
<?
view_guest($guest_file);
?>
</body>
</html>
```

mana shu ko'dni o'rganib chiqing.

Database (Ma'lumotlar bazasi bilan ishlash)

Bu mavzuyimizda biz database'lar bilan ishlashni ko'rib chiqamiz, yani MySQL, MS Access, MS SQL, va shu kabi boshqa bazalar bilan.

Bu mavzuda sizni SQL tilini bilasiz dib olamiz, va bu yerda sizga SQL tilini o'rgatmaymiz, chunki bu kitob faqatgina PHP haqidadir.

Biz bu yerda MySQL ma'lumotlar bazasi bilan ishlashni ko'ramiz.

mysql_connect()

Funktsiya MySQL serveriga bog'lanish uchun ishlatiladi. Ulangandan keyin esa kerakli bazani tanlashingiz mumkindir.

Int mysql_connect (string host [:port] [, string login] [, string passowrd])

misol:

```
@mysql_connect("localhost", "user", "user") or die ("Could not connect to MySQL server!");
```

```
<?
```

```
$link=@mysql_connect("localhost", "user", "user") or die ("Could not connect to server!");
```

```
?>
```

mysql_select_db()

MySQL bazasini tanlash uchun ishlatiladi.

Int mysql_select_db(string bazani_nomi [, int ulanish_korsatgichi])

misol;

```
<?
```

```
@mysql_connect("localhost", "user", "user") or die ("Could not connect to MySQL server!");
```

```
@mysql_select_db("guestbook");
```

```
?>
```

mysql_close()

Bu oldindan ochgan ulanishni yopish uchundir.

Int mysql_close([int ulanish_korsatgichi])

```
<?
```

```
@mysql_connect("localhost", "user", "user") or die ("Could not connect to MySQL server!");
```

```
@mysql_select_db("guestbook");
```

```
print "Ulandik";
```

```
@mysql_close();
```

```
print "Uzildik";
```

```
?>
```

Bu yerda biz avval ulandik keyin esa uzdik bog'lanishni. Bu yerda biz ulanish ko'rsatgichini ko'rsatmadik sababi bizda faqatgina bitta ulanish bor.

mysql_query()

Bu funktsiyamiz bazadan o'qish yoki bazaga yozish ishlari uchun, yani query uchun ishlatiladi.

Int mysql_query(string query [, int ulanish_korsatgichi])

query SQL tilidagi komanda yoki komandalar toplami bo'ladi.

Misolni keyingi funktsiyada ko'rib chiqamiz.

mysql_affected_rows()

Agarda biz bazaga yozsak mysql_query() bilan va biz qancha qatorlarni o'zgarganini bilmoqchi bo'lsak unda shu funktsiya bizga yordam beradi.

Int mysql_affected_rows([int ulanish_korsatgichi])

misol:

<?

```
@mysql_connect ("localhost", "user", "user") or die("Could not connect to Server!");
```

```
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
```

```
$query="update guestbook set name='test' where email='sinch@uz'";
```

```
$result=mysql_query($query);
```

```
print "Ozgargan qatorlar soni:", mysql_affected_rows();
```

```
mysql_close();
```

?>

Shunda oynangizga bitta o'zgargan yani "Ozgargan qatorlar soni: 1" deb chiqadi.

mysql_num_rows()

Funktsiyamiz Select komandasidan qaytgan qatorlar sonini beradi.

Int mysql_num_rows(int qaytish_raqami)

Qaytish raqami oldindan ishlatilgan query'ning raqamidir.

Misol:

<?

```
@mysql_connect ("localhost", "user", "user") or die("Could not connect to Server!");
```

```
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
```

```
$query="select name from guestbook";
```

```
$result=mysql_query($query);
```

```
print "Tanlangan qatorlar soni:", mysql_num_rows($result);
```

```
mysql_close();
```

?>

Shunda oynamizga qancha qatorlar tanlangan bo'lsa shularni sonini qaytaradi.

mysql_result()

Bu funksiya mysql_query() bilan ishlatiladi va Select komandasi uchundir. Select'dan qaytgan ma'lumotni ko'rish uchundir.

Int mysql_result(int qaytish_raqami, int yozuv [, mixed joy])

misol:

```
<?
@mysql_connect ("localhost", "user", "user") or die("Could not connect to Server!");
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
$query="select * from guestbook";
$result=mysql_query($query);
$x=0;
while($x<mysql_num_rows($result)):
$name=mysql_result($result, $x, "name");
$email=mysql_result($result, $x, "email");
$comments=mysql_result($result, $x, "comments");
print "$name $email $comments";
print "<br>";
$x++;
endwhile;
mysql_close();
?>
```

Shunda oynamizga ism email comments ketma ket chiqadi.

Misol uchun shu kabi

sinch uzinfo@inbox.ru hayot gozal

azmi azmi@yahoo.com men keldim

...

mana shu kabi.

mysql_fetch_row()

mysql_result() kabidir faqat farqi shundaki birdaniga bitta qatorni hammasini o'zgarmasga yozadi.

Array mysql_fetch_row(int qaytish_raqami)

keling misolda ko'raylik.

```
<?
@mysql_connect ("localhost", "user", "user") or die("Could not connect to Server!");
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
$query="select * from guestbook";
$result=mysql_query($query);
while($row=mysql_fetch_row($result)):
print "$row[name] $row[email] $row[comments]";
```

```
print "<br>";  
endwhile;  
mysql_close();  
</?>
```

Yuqoridagi funktsiya kabi chiqaradi.

mysql_fetch_array()

Aynan mysql_fetch_row() kabidir, ammo farqi shundaki bu funktsiyada siz o'zgarmaslarga qanday yozishni belgilaysiz, yani ismi bilan, yoki raqamli yoki ham raqam ham ismi bilan yozishi mumkin. Array mysql_fetch_array(int qaytish_raqami [, int index_turi])

index_turi: MYSQL_ASSOC – ismi kabi yozish (\$row["name"])

MYSQL_NUM – raqamli (\$row[1])

MYSQL_BOTH – ikkalasi

agarda hech qaysisini ko'rsatmasangiz unda MYSQL_ASSOC ni ishlatadi.

Misol:

```
<?  
@mysql_connect ("localhost", "user", "user") or die("Could not connect to Server!");  
@mysql_select_db("uzinfo") or die("Could not connect to Database!");  
$query="select * from guestbook";  
$result=mysql_query($query);  
while($row=mysql_fetch_array($result)):  
print "$row[name] $row[email] $row[comments]";  
print "<br>";  
endwhile;  
mysql_close();  
</?>
```

Mana yuqoridagini ayana qaytardi.

ODBC va PHP

O'zingiz yahshi bilasiz agarda siz MS Access bilan ishlamoqchi bo'lsangiz yoki boshqa bir baza bilan ishlamoqchi bo'lsangiz unda ODBC orqali ishlash eng qulayidir. PHP ham ODBC bilan ishlay oladi. ODBC bilan ishlash komandalari pastda birma bir yozib o'tilgandir.

odbc_connect()

mysql_connect() kabidir, farqi bu yaratilgan ODBC bilan ulanishdir.

Int odbc_connect (string odbc_ismi, string login, string password [, int cursor_turi])

keling misolga:

```
<?  
odbc_connect("accessDB", "user", "user") or die("Could not connect to ODBC");  
</?>
```

Mana shu kabi ishlatiladi.

odbc_close()

Ochilgan ulanishni uzadi, yani yopadi.

Void odbc_close([int ulanish_raqami])

odbc_prepare()

Qandaydir query'ni ishlatishdan oldin ODBC'ga siz tayorlanish buyrug'ini yuborishingiz kerak, yani kerakli query'ni shu funktsiya orqali yuborasiz va ODBC ishlatishga tayorlanib turadi, agarda ishlata olmasa unda FALSE qaytaradi, agarda ishlata olsa unda tayorlik_raqami'ni qaytaradi.

Int odbc_prepare (string ulanish_raqami, string query)

odbc_execute()

Query'ni ishlatish uchundir.

Int odbc_execute(int tayorlik_raqami [, array qoshimchalar])

qoshimchalar: Bu yerda siz agar query bilan birga yana qoshimcha ma'lumot yubormoqchi bo'lsangiz unda ishlatishingiz mumkindir.

Misol:

```
<?
$connect=@odbc_connect("accessDB", "user", "user") or die ("Could not connect to ODBC");
$query="update customers set cust_id=\"Chicago\" where cust_id= \" NYC\"";
$result=odbc_prepare($connect, $query) or die ("Could not prepare query!");
$result=odbc_execute($result) or die ("Could not execute!");
odbc_close($connect);
?>
```

mana ko'rganingzidek ODBC bilan ishlash ham juda oson. Keyingi funktsiyamizga qarang.

odbc_exec()

Bu funktsiyamiz odbc_prepare() va odbc_execute() funktsiyalarini birlashtiradi, yani bittada ishlatadi.

Int odbc_exec(int ulanish_raqami, string query)

misol:

```
<?
$connect=@odbc_connect("accessDB", "user", "user") or die("Could not connect to ODBC");
$query="select *from customers";
$result=odbc_exec($connect, $query) or die ("Could not execute query!");
odbc_close($connect);
?>
```

odbc_result_all()

Bu funktsiyamiz odbc_exec, yoki odbc_execute dan qaytgan ma'lumotlarni ko'rsatadi.

Int odbc_result_all(int qaytish_raqami [, string tablitsani_korinishi])

PHP Davrasida kitobi Open Source

misol:

```
<?
$connect=@odbc_connect("accessDB", "user", "user") or die("Could not connect to ODBC");
$query="select *from customers";
$result=odbc_exec($connect, $query) or die ("Could not execute query!");
odbc_result_all($result, "border='1'");
odbc_close($connect);
?>
```

Shunda quyidagi kabi oynaga chiqadi

cust_name	cust_adress
Engineering LTD	support@engine.com
BAT C.O.	admin@bat.com

odbc_free_result()

Butun qilgan ishlarigizdan so'ng hotirani bo'shatish uchun shu funktsiyadan foydalanasiz. Iloji boricha shu funktsiyani ishlatib, sababi butun qilingan ishlar kompyuteringizni hotirasida yoztmaydi, siz ulani olib tashlaysiz.

Int odbc_free_result(int qaytish_raqami)

```
<?
$connect=@odbc_connect("accessDB", "user", "user") or die("Could not connect to ODBC");
$query="select *from customers";
$result=odbc_exec($connect, $query) or die ("Could not execute query!");
odbc_result_all($result, "border='1'");
odbc_free_result($result);
odbc_close($connect);
?>
```

Mana shu bilan Bazalar bilan ishlashni tugatamiz va sizga kichik bir vazifa, sahifangizga odamlarga yoqgan site'larni nomini yozib qoldiruvchi va category'yaga bo'lingan holda chiqishini va yozilishi mumkin bo'lgan php sahifa yozing.

Ishingizda omad!!!

Yana kichik misol qilib sizga sahifaga kirilgan insonlarni butun ma'lumotlarini bazaga yozuvchi sahifa

PHP Davrasida kitobi Open Source

yo zamiz.

<?

```
$host = "localhost";
```

```
$user = "user";
```

```
$pswd = "user";
```

```
$database = "visitors";
```

```
$visitors_table = "visitors";
```

```
@mysql_connect($host, $user, $pswd) or die("Couldn't connect to MySQL server!");
```

```
@mysql_select_db($database) or die("Couldn't select $database database!");
```

```
$header_color = "#cbda74";
```

```
$table_color = "#000080";
```

```
$row_color = "#cOcOcO";
```

```
$font_color = "#000000";
```

```
$font_face = "Arial, Times New Roman, Verdana";
```

```
$font_size = "-1";
```

```
function recordUser() {
```

```
GLOBAL $visitors_table, $HTTP_USER_AGENT, $REMOTE_ADDR;
```

```
$timestamp = date("Y-m-d H:i:S");
```

```
$query = "INSERT INTO $visitors_table VALUES('$HTTP_USER_AGENT', '$REMOTE_ADDR',  
'$timestamp')";
```

```
$result = @mysql_query($query); }
```

```
// recordUser
```

```
function viewStats() {
```

```
GLOBAL $visitors_table, $table_color, $header_color;
```

```
GLOBAL $row_color, $font_color, $font_face, $font_size;
```

PHP Davrasida kitobi Open Source

```
$query = "SELECT browser, ip, timeofvisit FROM $visitors_table ORDER BY TimeofVisit desc";

$result = mysql_query($query);

print "<table cellpadding=\"2\" cellspacing=\"1\" width = \"700\" border = \"0\"
bgcolor=\"$table_color\">";

print "<tr bgcolor= \"\$header_color\"><th>Browser</th><th>IP</th><th>TimeofVisit</th></tr>";

while($row = mysql_fetch_array($result)):

    $op_sys = $row["browser"];
    print "<tr bgcolor=\"$row_color\">";

    print "<td><font color=\"$font_color\" face=\"$font_face\" size=\"$font_size\">$op_sys</font></td>";
    $rowed=$row["ip"];
    print "<td><font color=\"$font_color\" face=\"$font_face\" size=\"$font_size\">$rowed</font></td>";

    print "<td><font color=\"$font_color\" face=\"$font_face\" size=\"$font_size\">";

    print $row["timeofvisit"]."</font></td>";

    print "</tr>";

endwhile;

print "</table>"; }

// viewStats

?>
```

Mana shu ko'dni ko'rib o'rganib chiqing, bu sizga juda ham katta yordam beradi.

Cookies

Cookie – Bu internet'ning sizga beradigan qo'shimcha asbobi, yani qo'shimcha xususiyatidir. Ko'p insonlar cookie nima ekanligini bilmaydi va ularni ishlatmaydi. Agarda siz sahifangiz har bir kirgan inson uchun, unga yoqgan ranglarda yoki ko'rinishda yaratmoqchi bo'lsangiz unda cookie siz uchun eng kerakli va eng kuchli asbob bo'la oladi. Agarda siz kirgan insonni ismini so'rasangiz va uni cookie orqali yozib qo'ysangiz unda shu inson sahifangizga qayta kirganida siz misol uchun “Salom Mr. Deamon” deb kutib olishingiz mumkindir. Bu juda ham yahshi va qulay vositadir. Yoki ranglar, sahifa ranglarini insonga ko'ra o'zgartirish. Kirgan inson birinchi marotabada o'ziga yoqgan ranglarni tanlaydi va keyingi kirishida shu ranglar sahifangiz ko'rinishda bo'ladi.

Cookie har bir sahifa uchun alohida bo'ladi va faqatgina bitta o'zini ko'di bilan yoziladi. Shu ko'd orqali internet server sizni kimligingizni biladi va sizga mos o'zgartirishlarni kiritib sahifani ko'rsatib beradi.

Siz cookie'ni **setcookie** funksiyasi orqali yozishingiz mumkindir.

setcookie()

Shu funktsiya orqali siz sahifa mushtarisining kompyuteriga kerakli ma'lumotlarni yozishingiz mumkindir.

boolean setcookie (string ismi [, string ma'lumot [, int tugash_vaqi, [string yo'l, [string domain [, int security]]]])

misol:

```
setcookie("Testing_cookie", $value);  
setcookie ("Testing_cookie", $value, time()+3600); //1 soatdan so'ng ma'lumot ochib ketadi.  
setcookie ("Testing_cookie", $value, time()+3600, "/guest", "uzinfo.cjb.cc");  
//Bu yerda biz qaysi server va qaysi ichki joydan ekanligini ham belgilab berdik
```

a yozilgan cookie esa siz ismi bo'yicha qayta olishingiz mumkidir, yani aynan o'zgarmas kabi qayta olasiz. Keling kattaroq bir misolda ko'raylik.

fayl ismi cook.php dir

```
<?php  
if ($BeenSubmitted){  
    setcookie ("BGColor", "$NewBGColor");  
    setcookie ("TextColor", "$NewTextColor");  
    $BGColor=$NewBGColor;  
    $TextColor=$NewTextColor;  
}  
else {  
    if(!$BGColor){ $BGColor="White";}  
    if(!$TextColor) {$TextColor="Black";}  
}  
?>  
  
<html>  
<head>  
<title>File Upload</title>  
</head>  
<?  
print ("<body bgcolor=$BGColor text=$TextColor>\n");  
?>  
Currently Your page looks like this!
```

```
<FORM action="cook.php" method="post">
Select Background color:
<select name="NewBGColor">
<option value="white">white</option>
<option value="black">black</option>
<option value="blue">blue</option>
<option value="red">red</option>
<option value="green">green</option>
</select>
Select a new text color:
<select name="NewTextColor">
<option value="white">white</option>
<option value="black">black</option>
<option value="blue">blue</option>
<option value="red">red</option>
<option value="green">green</option>
</select>
<input type="hidden" name="BeenSubmitted" value=true>
<input type="submit" name="SUBMIT" value="Submit!!!">
</form>
</body>
</html>
```

Bu misolda biz qanday qilib biz ranglar berishni ko'rdik, yani sahifani orqa rangi va harflar rangini belgilab berishni va keyin esa qayta ularni o'qish orqali sahifa rangini keyingi kirishda o'zgartirishni ko'rdik.

Agarda siz cookie'ni o'chirib tashlash niyatingiz bo'lsa unda vaqtga manfiy ma'lumot yubirng.

Misol:

```
setcookie("TestCookie", $value, time()-60);
```

Mana shunda sizning yozgan ma'lumotingiz o'chib ketadi.

Ha yatgancha cookie'ni siz sahifa boshida belgilab yozishingiz shart, bo'lmasam ishlamaydi.

Mana shu bilan biz boshlovchilar uchun yozilgan PHP kitobimizni tugatdik.

Lekin bu degani PHP'ning hamma funktsiya va komandalarini ko'rib chiqdik degani emas, biz ko'rib chiqmagan yana komandalar va funktsiyalar bor. Niyat qilamanki bu kitobim PHP olamiga bo'lgan sizning birinchi qadamingizda katta yordam berdi.