O’ZBEKISTON RESPUBLIKASI OLIY VA O`RTA MAXSUS

TA’LIM VAZIRLIGI

BUXORO DAVLAT UNIVERSITETI

Fizika –matematika fakulteti

“Axborot texnologiyalari” kafedrasi

“Kompyuter grafikasi va web dizayn” fanidan

REFERAT

Mavzu: PHP takrorlash operatorlari va ulardan foydalanish.

Bajardi: **Hamroyeva M.**

Qabul qildi: **Xayatov X.**

Buxoro-2016

Mavzu: PHP takrorlash operatorlari va ulardan foydalanish.

Reja:

1. PHP tilining qisqacha tarixi.PHP tilining imkonyatlari

1994 yili PHP tilinig yaratuvchisi **Rasmus Lerdorf** o’zinig saytiga mehmonlar kirishini hisoblash uchun **Perl** dasturlash tilada maxsus qobiq yozib amalda qo’llagan. Ko’p o’tmay qobiqni ishlash unumdorligi juda past va sekinligi aniqlanganidan so’ng, dasturlarni yangidan “**C**” tilida yozib chiqishga to’g’ri keladi. Keyin, dastlabki dastur kodlari muallif tarafidan barchaga ko’rish uchun serverga nashr qilingan. Server foydalanuvchilari kodlar bilan qiziqib, uni ishlatish muxlislari ham paydo bo’lgan.
Hademay, bu dasturlar alohida loyihaga aylanib, 1995 yilning iyun oyida dasturiy mahsulot **PHP (Personal Home Page)**nomi bilan birinchi nashri chiqarildi.

1996 yil aprel oyida dasturlar jiddiy qayta ishlanganidan so’ng, **PHP/FI** (Personal Home Page / Forms Interpreter) nomi bilan mahsulotning ikkinchi nashri paydo bo’ldi. Bu mahsulot html-kod ichiga yozilib, html-formalarni qayta ishlab, hozirgi PHP dasturlash tilining tayanch imkoniytlarini ichiga olgan. **PHP/FI** kod yozilishi **Perl** tiliga juda oxshagan, lekin soddaroq bo’lgan.
1997 yili **PHP/FI 2.0** nashri chiqdi. O’sha paytda bu mahsulot bilan dunyo bo’yicha bir necha ming odam foydalanib, taxminan 50 ming domen bo’lib, Internetning 1%-ni tashkil qildi.

1997 yilda **Endi Gutmans** va **Ziv Suraski** PHP/FI kodini boshqatdan yozib chiqishdi, chunki eski kod ular ishlatayotgan elektron tijorat tizimlari uchun yaroqsiz edi. Eski kodning mualliflaridan yordam olish uchun ular birlashishni taklif etib,**PHP3** nomli loyihani **PHP/FI** -ni **rasmiy vorisi** deb e'lon qilishdi. Yangi loyiha uyushgandan keyin PHP/FI loyihasi ishlab chiqarilishi to'xtatilgan.

PHP 3.0 -ning eng kuchli taraflaridan biri uning kengaytirala olinadigan yadrosi(tizimning bosharuv qismi) bo'lib, bundan tashqari, ma'lumot jamg'armalar bilan, turli protokollar va interfeyslar bilan birgalikda ishlash keng imkoniyatlari yaratildi. Muvaffaqiyatga erishishga ancha ahamiyatli fakt bu yangi tilni boyligi va ob'ektlarga mo'ljallangan dasturlashni qo'llay olishi. Yangi loyiha bilan birga nafaqat tilni tashqi, ichki tuzulishi o'zgardi, balki o'zini nomi ham. Endi PHP qisqartmasi "**PHP: Hypertext Preprocessor**" ma'nosini anglatishi bildirildi.

1998 yilning oxirida PHP foydalanuvchilarning soni o'n minglardan oshdi. Yuz mingdan oshiq veb-saytlar bu tilni qo'llashini e'lon qilishdi. Taxminan Internetning 10% serverlarida **PHP 3.0** o'rnatilgan edi.

1998 yilning iyun oyda PHP3 to'qqiz oy ommaviy tekshiruvidan keyin rasman e'lon qilindi. Shu yilning qishida **Endi Gutmans** va **Ziv Suraski** PHP yadrosini qaytadan ishlab chiqarishni boshlashgan. Ularning asosiy vazifasi PHP tiziminig unumdorligini ko'tarish va kodning modullarini yaxshilash edi.

1999 yilning o'rtalarida birinchi marta taqdim qilingan yangi yadro "**Zend Engine**" deb nomlangan ("**Zend**": mualliflar "**Zeev**" va "**Andi**" ismlardan tashkil topgan). Uni asosida tuzilgan yangi til **PHP4** 2000 yilning may oyida rasman chiqarilgan. Unumdorlik yaxshilangandan tashqari, PHP 4.0 muhim yangiliklarga ega bo'lib, sessiyalarni qo'llash, buferli chiqarish, kiritilgan ma'lumotlarni havfsiz qayta ishlash va yana bir necha yangi til tuzuvchilarini paydo bo'lishidan iborat.

Hozirgi kunlarda "**Zend Engine**" qayta yaxshilanib **PHP5** tili ishlab chiqarildi. Asosiy o'zgarishlar ob'ektlarga mo'ljallangan dasturlash modelida bo'lib, tilning imkoniyatlari yanada kengaytirdi.

2. PHP da o'zgaruvchilar,o'zgarmaslar va ma'lumotlar turlari

Dastur ushlash davomida juda ko'p ma'lumotlardan foydalanadi, ularni qayta ishlaydi. Demakki bu ma'lumotlarni qayergadir va qandaydir usulda saqlashi kerak bo'ladi. Sistema dasturchilari kabi masalaga juda chuqur kirishmasdan bu ma'lumotlarga qanday murajaat qilish masalasini ko'rib chiqaylik. Foydalanuvchi murojaatini yengillashtirish maqsadida PHP da o'zgaruvchi tushunchasi kiritilgan. Aniqrog'i o'zgaruvchi ma'lumotni nomidir, qaysiki biror ma'lumotga murojaat qilmoqchi bo'lsak, qiymatgaga emas balki uning nomiga murojaat qilinadi. Dastur ishlash davomida o'zgaruvchiga bir necha marta qiymat berish, ya'ni uning qiymatini o'zgartirishimiz mumkin.
Dastur bajarilish davomida qiymati o'zgarishi mumkin bo'lgan miqdorlarga o'zgaruvchilar deyiladi.
O'zgaruvchilardan foydalanish to'g'risida bir necha misollar ko'ramiz:

```
$a = “20”;
$b = 20;
?>
```

Yuqoridagi misolda ikkita $a va $b o'zgaruvchilardan foydalanilgan, ularning birini qiymati “20” va ikkinchisining qiymati 20 ga teng.

```
$a=”500”;
$b=”200”;
$c=$a+$b;
$d=$a.$b;
Echo $c; //Natija 700
Echo $d; // Natija 500200
?>
```

O'zgaruvchini nomlash qoidalari:

1)O'zgaruvchi $ belgisidan boshlanadi;
2)nom-orasida probel belgisi qo'yilmaydi;
3)katta va kichik harflar farqlanadi;
$a–to'g'ri

$abc–to'g'ri

$new_array–to'g'ri

new_array–noto'g'ri

$newarray–noto'g'ri

$new_array va $New_array boshqa-boshqa o'zgaruvchilardir.

Dastur bajarilish davomida uning qiymati o'zgaradi:

```
$a = 21;
echo $a; // natija 21
$a = $a + 5;
echo $a; // natija 26
$a++;
echo $a; //natija 27
?>
```

Har bir o'zgaruvchi muayyan bir turga tegishli bo'ladi. Ko'pchilik hollarda o'zgaruvchining turini PHP avtomatik aniqlab oladi:

```
$a=5; // $a o'zgaruvchining turu butun son
$str = “5”; // $str o'zgaruvchining turi satrli miqdor
$dbl = 5.0; // $dbl o'zgaruvching turi haqiqiy son
$arr = array(“a”,”b”,”c”); // $arr o'zgaruvchi massiv (jadval ko'rinishidagi miqdor)
// uning 3ta elementi ham satrli miqdor
$bool=true // $bool mantiqiy turli o'zgaruvchi
// u faqat true va false qiymatlarni qabul qilishi mumkin
?>
```

PHP dasturda qaysidir o'zgaruvchi e'lon qilinganligini tekshirib ko'rish uchun isset(o'zgaruvchi) funcsiyasidan foydalaniladi:

```
if (isset($MyVar)) echo “bunday o'zgaruvchi mavjud, uning qiymati: $MyVar”;
?>
```

isset($MyVar) funksiyasi true qiymat qaytaradi, qachonki $MyVar mavjud bo'lsa, aks holda false qaytaradi.

Ko'p hollarda foydalanib bo'lingan o'zgaruvchini xotirani tejash maqsadida o'chirishga to'g'ri keladi. Buning uchun unset() funcsiyasidan foydalaniladi:

```
$MyVar=”QIYMAT”;
if (isset($MyVar)) {
echo “Bunday o'zgaruvchi mavjud, uning qiymasi:$MyVar”.”
”;
} else {
echo “Bunday o'zgaruvchi mavjud emas”;
} // Natija: Bunday o'zgaruvchi mavjud, uning qiymati:QIYMAT
unset($MyVar);
if (isset($MyVar)) {
echo “Bunday o'zgaruvchi mavjud, uning qiymasi:$MyVar”.”
”;
} else {
echo “Bunday o'zgaruvchi mavjud emas”;
} // Natija: Bunday o'zgaruvchi mavjud emas
?>
```

Ayrim hollarda o'zgaruvchi turini aniqlab olishga to'g'ri keladi. Bunday hollarda bir qator funksiyalardan foydalanishga to'g'ri keladi:

1) is_integer($MyVar) – TRUE qiymat qaytaradi, agar $MyVar butun son bo'lsa.

2) is_double($MyVar) – TRUE qiymat qaytaradi, agar $MyVar haqiqiy son bo'lsa.

3) is_string($MyVar) – TRUE qiymat qaytaradi, agar $MyVar satrli miqdor bo'lsa.

4) is_nomeric($MyVar) – TRUE qiymat qaytaradi, agar $MyVar sonli miqdor bo'lsabo'lsa(butunyokihaqiqiysonbo'lsa).

5) is_bool($MyVar) – TRUE qiymat qaytaradi, agar $MyVar mantiqiy turli o'zgaruvchibo'lsabo'lsa.

6) is_scalar($MyVar) – TRUE qiymat qaytaradi, agar $MyVar murakkab tur bo'lsa.

7) is_null($MyVar) – TRUE qiymat qaytaradi, agar $MyVar o'zgaruvchining

qiymatiNULLbo'lsa.

8) is_array($MyVar) – TRUE qiymat qaytaradi, agar $MyVar massiv bo'lsa.

9) is_object($MyVar) – TRUE qiymat qaytaradi, agar $MyVar obyektga bog'lanish bo'lsa.

10) gettype($MyVar) – integer, double, string, boolean, array, object, array, NULL yoki unknown type kabi natijajarni qaytaradi.

Konstantalar dastur bajarilish jarayonida qiymatini o'zgartirmaydi. Konstantalarni e'lon qilish uchun define() funksiyasidan foydalaniladi.

```
difine(“pi”,3.1416);
$a =sin(pi/4)
echo “Pi sonining qiymati: ”.pi; // Natija: Pi sonining qiymati: 3.1416
?>
```

PHPda ayrim bir o'zgarmaslar oldindan aniqlab qo'yilgan. Ularni ayrimlarini keltiramiz, to'liqroq ma'lumotlarni PHP ma'lumotnomalaridan olishingiz mumkin:

__FILE__ - ayni vaqtda qaysi fayldagi kod bajarilayotgan bo'lsa, shu fayl nomi saqlanadi.

__LINE__ ayni vaqtda bajarilayotgan satr raqamini saqlaydi.

3 PHP tilida dasturlash asoslari

PHP dasturlash tili tilida tuzilgan har qanday dastur **.php** kengaytmali fayllarda saqlanadi. Misol: uchun **functions.php**, **index.php**, **admin.php** kabi. Birnchi misolni ko'ramaiz. **test.php** nomli fayl tashkil etamiz va unga quyidagi kodlarni yozamiz:

```
<html>
<head>
<title>HTML hujjat</title>
<head>
<body>
Hujjat tanasi
</body>
```

```
</html>
```

Bu matn HTML asosida yozilgan. 1-misoldan ko'rinib turibdiki php hujjat HTML hujjat kabi shakllantirilmoqda. Lekin php hujjatda biz qo'shimcha imkoniyatlarga ega bo'lamiz.**test.php** faylga quyidagi o'zgarishlarni kiritamaiz:

```
<?php
Echo "<html>";
Echo "<head>";
Echo "<title>HTML hujjat</title>";
Echo "<head>";
Echo "<body>";
Echo "Bu matn HTML asosida yozilgan";
Echo "</body>";
Echo "</html>";
?>
```

2-misol ham 1-si kabi hujjat hosil qiladi. Farqi shundaki ikkinchi holda HTML teglarni PHP dasturi yordamida hosil qilinmoqda. Lekin ko'p hollarda HTML teglari butunicha PHP yordamida hosil qilinmay ora-oralarda PHP kodlari yoziladi:

```
<?php
$title = “HTML hujjat”;
$text = “Bu matn PHP asosida yaratilgan”;
?>
<html>
<head>
<title><?php echo $title; ?></title>
</head>
<body>
```

```
<?php echo $text; ?>
</body>
</html>
```

Yuqoridagi misolni bajarib ko'ring va internet sharhlovchidan HTML kodni ko'rish opsiyasini tanlang. Va siz yana yuqoridagidek HTML teglarini ko'rasiz. Bu yerda PHP kod HTML teglarni generatsiya qilyapti. Demak PHP dasturining vazifalaridan biri HTML teglarini generatsiya qilish ekanini bilib oldik. PHP ning bundan tashqari yana ko'plab imkoniyatlari mavjud bo'lib unga ma'lumotlar bazasiga murojaat qilish, turli hisob-kitoblarni amalga oshirish kabilar ham kiradi. Lekin PHPdan ko'pchilik aynan klent kompyuterida emas, balki serverda bajarilishi uchun foydalanadi. PHP dasuri tuzish haqida shu vaqtgacha bilib olgnlarimizni umumlashtiramiz:

1) *.php kengaytmali fayl yaratiladi;

2) faylda istalgancha HTML teglarini ishlatish mumkin;

3) PHP kodlarini ochish va yopish belgilari orasida yoziladi:

a) <?php –ochish belgizi va ?> -yopish belgisi;

b) <? –ochish belgisi va ?> - yopish belgisi;

c) <script language = "php"> -ochish belgisi va </script> - yopish belgisi;

d) <?= -ochish belgisi ba ?> yopish belgisi.

PHP kodlarini yozish bo'yicha yana bitta misol ko'ramiz:

```
<?php // birinchi usuldan foydalanilmoqda
$title = “HTML hujjat”;
$text = “Bu matn PHP asosida yaratilgan”;
?>
<html>
<head>
<title>
<? // ikkinchi usuldan foydalanilmoqda
```

```
echo $title;
?>
</title>
</head>
<body>
<script language = “php”> // uchunchi usuldan foydalanilmoqda
echo $text;
</script>
</body>
</html>
```

II BOB PHP tilining operatorlari

4. PHP konstruksiyasida takrorlanuvchi operatorlarni qo’llash

For

For sikli PHP ning eng muhim sikllaridan biridir.Uning sintaksisi quyidagichadir:

```
for (1-ifoda; shart; 2-ifoda) operator
```

1-ifoda bir marta sikl boshida bajariladi.

shart siklning har bir takrorlanishida tekshiriladi. Agar u TRUE qiymat qabul qilsa, sikl tanasi bajarilishi davom ettiriladi. Agar u FALSE qiymat qabul qilsa sikl o’z ishini to’xtatadi.

Ha bir sikl oxirida 2-ifoda bajariladi.

Yuqoridagi shart va ifodalarning har biri bo’sh bo’lishi mumkin. Agar shart berilmagan bo’lsa cheksiz sikl vujudga keladi. Bunday holatda sikl bajarilishini break operatori orqali to’xtatish mumkin.

Quyida 1 dan 5 gacha bo’lgan sonlarni ekranga chiqaruvchi to’rtta misolni ko’ramiz:

```
/* 1 - misol */
```

```
for ($i = 1; $i <= 10; $i++) {
    print $i;
}

/* 2 - misol */

for ($i = 1;;$i++) {
    if ($i > 10) {
        break;
    }
    print $i;
}

/* 3 - misol */

$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
}

/* 4 - misol */

for ($i = 1; $i <= 10; print $i, $i++);
```

For siklidan foydalanishning to’rt xil usulidan qaysidir birini ustun deyishimiz mumkin, lekin har bir usul joyida qulaylik olib kelishi mumkin.

Bundan tashqari FOR siklining alternativ sintaksisi ham mavjud:

```
for (1-ifoda; shart; 2-ifoda): operatorlar; endfor;
```

foreach

PHP assortativ massivlardan foydalanadi va bu foydalanishda katta ustunliklarni keltiradi. Lekin assortativ massivlar uchun sikllar tashkil qilish bir muncha noqulayliklarga sabab bo'lishi mumkin. Bunga asosiy sabab foydalanmoqchi bo'lgan massivimizning elementlari haqida oldindan tasavvurga ega bo'lmasligimizdir.

PHP4 versiyasidan boshlab foreach konstruksiyasidan foydalanish imkoniyati mavjud. U massivlar uchun sakl tashqil qilishni soddalashtiradi.Foydlanishning ikki xil sintaksisi mavjud:

```
foreach(array_expression as $value) statement
foreach(array_expression as $key => $value) statement
```

Birinchi ko'rinishda array_expression massivi uchun sikl tashkil qilinadi. Har bir o'tishda $value o'zgaruvchiga massivning joriy elemnti qo'yiladi va massivning ichki ko'rsatkichi bittaga oshiriladi.

Ikkinchi ko'rinish ham yuqoridagi kabi bo'lib, farqi joriy element indexsi/key $key o'zgaruvchiga o'zlashtiriladi va bundan foydalanish mumkin.

Quyoda bir qator misollar keltiramiz:

```
/* 1-misol */
reset ($arr);
while (list(, $value) = each ($arr)) {
    echo "Value: $value<br>\n";
}

foreach ($arr as $value) {
    echo "Value: $value<br>\n";
}
```

```
/* 2-misol */
reset ($arr);
while (list($key, $value) = each ($arr)) {
    echo "Key: $key; Value: $value<br>\n";
}

foreach ($arr as $key => $value) {
    echo "Key: $key; Value: $value<br>\n";
}
```

Yana bir qancha misollar keltiramiz

```
/* foreach 1-misol: faqat value */

$a = array (1, 2, 3, 17);

foreach ($a as $v) {
   print "Value = \$a: $v.\n";
}

/* foreach 2-misol: value (? key, ????????? ??? ???????????) */

$a = array (1, 2, 3, 17);

$i = 0; /* ?????? ??? ??????????? */

foreach($a as $v) {
    print "\$a[$i] => $v.\n";
```

```
    $i++;
}

/* foreach ?????? 3: key\???? ? value\???????? */

$a = array (
    "one" => 1,
    "two" => 2,
    "three" => 3,
    "seventeen" => 17
);

foreach($a as $k => $v) {
    print "\$a[$k] => $v.\n";
}

/* foreach ?????? 4: ??????????? ??????? */

$a[0][0] = "a";
$a[0][1] = "b";
$a[1][0] = "y";
$a[1][1] = "z";

foreach($a as $v1) {
    foreach ($v1 as $v2) {
        print "$v2\n";
    }
}
```

```
/* foreach ?????? 5: ???????????? ??????? */

foreach(array(1, 2, 3, 4, 5) as $v) {
    print "$v\n";
}
```

While

Whele sikli PHP ning oddiy siklidir Undan foydalanish sintaksisi quyida keltirilgan:

```
while (shart) operator
```

PHP while siklini bajarganda oldin shartni tekshirib ko'radi, agar shart TRUE qiymat qabul qilsa operator(lar)ni bajaradi. Bu jaraoyon toki shart FALSE qiymat qabul qilmaguncha davom etadi. Demak operator bir marta ham bajarilmasligi ham, yoki juda ko'p marta bajarilishi ham mumkin ekan. Dastur tuzish davomida e'tibor berishimiz kerak bo'lgan asosiy narsa takrorlanishning cheksiz davom etmasligini ta'minlashdir.

While operatori ham zarur bo'lganda if operatori kabi {} konteyneridan foydalanadi

```
while (expr) {
  1-operarot;
  2-operator;
  .....
  n-operator
}
```

Quyidagi misolda 1dan 5 gacha bo'lgan sonlar ekranga chiqariladi:

```
$i = 1;
while ($i <= 5) {
    print $i++;
}
```

While sikli zarur bo’lganda alternativ sintaksisni ham qo’llaydi

```
$i = 1;
while ($i <= 5):
    print $i+;
    $i++;
endwhile;
```

Do .. while

Do..while sikli while sikliga juda o’xshash, ammo bu yerda shart har bir sikl oxirida tekshiriladi. Va buning natijasida sikl hech bo’lmaganda bir matra bajarilishi ta’minlanadi.

do..while siklidan bir xil sintaksisda foydalanish mumkin:

```
$i = 0;
do {
   print $i;
} while ($i>0);
```

Yuqorida keltirilgan sikl faqat bir marta bajariladi, $i o’zgaruvchi 0 dan katta bo’lmagani uchun shart FALSE qiymat beradi hamda sikl bajarilishi to’xtaydi.

5. PHP konstruksiyasidagi boshqa operatorlar

switch operatori

switch operatori if operatorlari seriyasidan foydalanishga o'xshaydi. Ko'p hollarda bir o'zgaruvchini qiymatiga qarab dasturni turli qismlarini bajarishga to'g'ri keladi. Aynan bunday hollarda switch operatoridan foydalanish qulaydir.

Quyidagi ikkita misol bitta masalani hal qilsada birisi if va ikkinchisi switch konstruksiyasidan foydalanib tuzilgan.

```
if ($i == 0) {
print "i ning qiymati 0";
}
if ($i == 1) {
print "i ning qiymati 1";
}
```

```
if ($i == 2) {
print "i ning qiymati 2";
}
switch ($i) {
case 0:
print " i ning qiymati 0";
break;
case 1:
print "i ning qiymati 1";
break;
case 2:
print "i ning qiymati 2";
break;
}
```

Switch operatori PHP tomonidan qadam-baqadam bajariladi. Agar har bir case operatorining oxirida break ni yuqoridagi misoldagidak qo'llamasak bitta uchastkaning o'rniga bir nechta uchastka bajarilishi mumkin. Quyidagi misolda agar $i ning qiymati 0 bo'lsa "i ning qiymati0 i ning qiymati1 i ning qiymati2" kabi natija olamaiz.

```
switch ($i) {
case 0:
print "i ning qiymati 0";
case 1:
print " i ning qiymati 1";
case 2:
print " i ning qiymati 2";
}
```

Switch operatorida shart bir marta tekshiriladi, lekin bu ifoda har bir case operatori bilan solishtirib chiqiladi. Agar elseif operatorida shart qayta tekshirilishini hisobga olsak switch operatori if operatoridan tezroq ishlashini bilib olishimiz mumkin.

Case operatorlarini bo'sh qo'llash ham mumkin.

```
switch ($i) {
case 0:
case 1:
case 2:
print "i ning 3 dan kichik";
break;
case 3:
print "i ning qiymati 3 ga teng";
}
```

Agar case dagi ifodalarning birortasi ham mos bo'lmaganda default case dan foydalanish mumkin. U eng oxirgi o'rinda yoziladi:

```
switch ($i) {
case 0:
print " i ning qiymati 0 ga teng ";
break;
case 1:
print " i ning qiymati 1 ga teng ";
break;
case 2:
print " i ning qiymati 2 ga teng ";
break;
default:
print "i ning qiymati 0, 1, 2 ga teng emas";
}
```

Ehtiyoj bo'lganda switch operatorining alternativ sintaksisidan foydalanish mumkin:

```
switch ($i):
case 0:
print " i ning qiymati 3 ga teng ";
break;
case 1:
print " i ning qiymati 3 ga teng ";
break;
case 2:
print " i ning qiymati 3 ga teng ";
break;
default:
print "i ning qiymati 0, 1, 2 ga teng emas ";
endswitch;
```

break

break operatori **for, foreach, while, do..while** sikllaridan chiqish hamda switch konstruksiyasidan chiqish maqsadida foydalaniladi.

break shart bo’lmagan raqamli argumentni qabul qilishi mumkin, qaysiki u ichma-isch joylashgan qancha miqdorda konstruksiya ishini tugatishi kerakligini belgilaydi:

```
$arr = array ('one', 'two', 'three', 'four', 'stop', 'five');
while (list (, $val) = each ($arr)) {
    if ($val == 'stop') {
        break;    /* Bu yerda 'break 1;'  kabi yozish ham mumkin edi */
    }
    echo "$val<br>\n";
}

/* Shart bo’lmagan argumentdan foydalanish. */
```

```
$i = 0;
while (++$i) {
    switch ($i) {
    case 5:
        echo "At 5<br>\n";
        break 1;  /* switch dan chiqish. */
    case 10:
        echo "At 10; quitting<br>\n";
        break 2;  /* switch dan chiqish va while dan chiqish. */
    default:
        break;
    }
}
```

continue

continue operatori sikllarni boshqarishda foydalaniladi. Agar sikl ichida contiune operatori uchrasa siklning joriy iteratsiyasi ishini to’xtatadi, ammo shiklning keyingi iteratsiyasi bajarilishini davom ettiradi

continue shart bo’lmagan argumentni qo’llaydi, qaysiki siklning nechta iteratsiyasini bajarmasdan tashlab ketish kerakligini belgilaydi.

```
while (list ($key, $value) = each ($arr)) {
    if (!($key % 2)) { // juft iteratsiyalarni tashlab ketish
        continue;
    }
    do_something_odd ($value);
}

$i = 0;
while ($i++ < 5) {
    echo "Outer<br>\n";
    while (1) {
        echo "& & Middle<br>\n";
        while (1) {
            echo "& & Inner<br>\n";
            continue 3;
        }
        echo "This never gets output.<br>\n";
    }
    echo "Neither does this.<br>\n";
}
```

FOYDALANILGAN ADABIYOTLAR

1. Alimov S. ―PHP davrasida‖ Toshkent 2006. – 85 b.

2. Котеров Д. В. «Самоучитель PHP4» БХВ-Петербург, 2001. — 572 с

3. Кузнецов М. В. «PHP 5 на примерах» БХВ-Петербург, 2005. — 577 с

4. Томсон Лаура «Разработка Web-приложений на PHP и MySQL» ДиаСофтЮП, 2003. — 672 с

Internet tarmog’idagi manbalar:

1. www.google.co.uz

2. www.php.su

3. www.php.net

4. www.ziyonet.uz

5. www.library.tuit.uz/konspekty.htm