

№14 лабораториялық жұмыс

PHP тіліндегі *айнымалы* аты доллар (\$) таңбасымен басталып жазылады. Мысалы:

\$my_var

Айнымалының аты регистрге тәуелді, мыс: **\$my_var** және **\$My_var** әр түрлі *айнымалылар* болып саналады.

PHP тіліндегі барлық *айнымалылар* аттары идентификатор арқылы жазылып, жоғарыдағы ережеге бағынуы тиіс: айнымалының дұрыс аты кез келген әріптен басталатын әріптер, сандар немесе астын сызу таңбалары тізбегінен тұрады.

1.1 Тіркестік айнымалылармен жұмыс істеу

Сөз тіркестері арасындағы . нүкте таңбасы олардың бір-біріне жалғаса шығарылатынын, яғни “біріктіру” (конкатенация) белгісі екендігін білдіреді.

1 мысал.

```
<?php
$first = ' Text '; // Присваиваем $first значение ' Text '
$second = $first; // Присваиваем $second значение переменной $first
$first = ' New text '; // Изменяем значение $first на ' New text '
echo "Переменная с именем first " .
    "равна $first <br>";
    // выводим значение $first
echo "Переменная с именем second " .
    "равна $second";
    // выводим значение $second
?>
```

Осы скрипт жұмысының нәтижесі төмендегідей болады:

Переменная с именем first равна New text

Переменная с именем second равна Text

1.2 Тұрақтылармен жұмыс істеу

Script орындау барысында тұрақты шамаларды, яғни мәні өзгермейтін шамаларды сақтау үшін **константалар** немесе **тұрақтылар** пайдаланылады. Константаларға математикалық *тұрақтылар*, құпия сөздер (парольдар), файл адрестері және т.б. жатады. *Тұрақтының айнымалыдан* негізгі айырмашылығы – оған мән бір-ақ рет беріледі және оның жарияланған мәнін өзгертілмейді. Бұған қоса тұрақтылар аты алдына доллар таңбасы жазылмайды және оларға мәнді қарапайым меншіктеу арқылы беруге болмайды. Тұрақтыларды анықтау үшін **define()** арнайы функциясы қолданылады. Оның синтаксисі мынадай:

```
Define("Тұрақты_аты", "Тұрақты_мәні", ["Регистрден_тәуелсіздігі"])
```

Келісім бойынша тұрақтылар аттары регистрге тәуелді, бірақ әрбір тұрақты үшін қосымша аргумент ретінде **Регистрден_тәуелсіздігі** мәнін **True** мәніне өзгертіп, бұл қасиетті жоюға болады. Келісім бойынша әрқашанда тұрақты аты жоғарғы регистрде (бас әріптермен) жазылады.

Тұрақтының мәні оның \$ таңбасысыз жазылатын атын көрсету арқылы пайдаланылады. Бұған қоса тұрақты мәнін алу үшін аргументі тұрақты атына сәйкес **constant()** функциясын пайдалануға болады.

2 мысал.

```
<?php // определяем константу PASSWORD
define("PASSWORD","qwerty");
    // определяем регистронезависимую константу PI = 3.14
define("PI","3.14", True);
    // выведет значение константы PASSWORD, т.е. qwerty
echo (PASSWORD);
    // сл. строка тоже выведет qwerty
echo constant("PASSWORD");
echo (password); /* выведет password и предупреждение, поскольку мы
    ввели регистрозависимую константу PASSWORD */
echo pi; // выведет 3.14, т.к. константа PI
    // регистронезависима по определению
?>
```

1.3 Boolean типі (бульдік немесе логикалық тип)

Бұл қарапайым *тип* мәннің ақиқат екендігін, яғни осы *типтегі* айнымалының тек 2 мәнді ғана – TRUE (ақиқат) және FALSE (жалған) қабылдайтынын білдіреді.

Бульдік *типті* анықтау үшін TRUE немесе FALSE түйінді сөздері қолданылады. Бұл екеуі де регистрден тәуелсіз мәндер болып табылады.

Логикалық *айнымалылар* әр түрлі *басқарушы конструкцияларда* (циклдерде, шарттарда және т.б.) пайдаланылады. *Логикалық типті*, яғни екі ғана мәнді – ақиқат немесе жалған мәндерін, кейбір *операторлар* да қолдана алады (мысалы, теңдік түріндегі *оператор*). Сонымен қатар олар кез келген бір шартты тексеру кезінде жиі пайдаланылады. Мысалы, шартты нұсқауда *оператордың* немесе *айнымалының* ақиқаттық мәні тексеріледі де, тексеру нәтижесіне байланысты әр түрлі әрекеттер орындалады. Бұл жерде шарттың ақиқаттығы немесе жалғандығы *айнымалының* немесе *оператордың логикалық типте* екендігін көрсетеді.

3 мысал.

```
<?php //Оператор '==' проверяет равенство и выдает булево значение
if ($know == False) { // если $know имеет значение false
echo"Изучай PHP!";
}
if (!$know) { // то же самое, что и выше, т.е. проверка
    // имеет ли $know значение false
echo"Изучай PHP!";
}
/* оператор == проверяет, совпадает ли значение переменной $action
со строкой "Изучить PHP". Если совпадает, то возвращает true,
иначе - false. Если возвращено true, то выполняется то, что внутри
фигурных скобок */
if ($action == "Изучить PHP")
{ echo"Начал изучать"; }
?>
```

1.4 Integer (бүтін сан) типі

Бұл *тип* бүтін сандар жиынын анықтайды $Z=\{\dots, -2,-1,0,1,2, \dots\}$. Бүтін сандар «-» немесе «+» таңбаларынан басталатын ондық, оналтылық немесе сегіздік санау жүйесінде болуы мүмкін. Егер сегіздік санау жүйесі пайдаланылса,

сан міндетті түрде 0 санынан басталады, ал оналтылық санау жүйесінде ол 0х таңбаларынан басталуы тиіс.

4 мысал.

```
<?php
$a = 1234;
$b = -1.2e3;
$c = 0123; // восьмеричное число 83 в 10-ой системе
$d = 0x1A; // шестнадцатеричное число 26 в 10-ой системе
echo "a = $a <br>";
echo "b = $b <br>";
echo "c = $c <br>";
echo "d = $d <br>";
?>
```

1.5 Float (жылжымалы түктемен берілеті сандар) типі

Жылжымалы түктемен берілетін сандарды (олар екі еселенген дәлдікте немесе нақты сандар түрінде бола береді) келесідей түрдегі кез келген ережелердің бірімен анықтауға болады:

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Бүтін сан мөлшері компьютерлер класына (16- не 32-разрядты) байланысты болады, бірақ көбінесе мұндағы ең үлкен сан $\sim 1.8e308$ шамасында, ал дәлдігі үтірден кейінгі 14 ондық разрядпен беріледі.

5 а мысал.

```
<?php
$a = 1.234;
$b = 1.2e-3;
$c = 7E-10;
$d = 1.4e2;
$e=0.1*($a+$b)/$d;
echo "a = $a <br>";
echo "b = $b <br>";
echo "c = $c <br>";
echo "d = $d <br>";
echo "e = $e <br>";
?>
```

5 ә мысал.

```
<?php
$a = ($b=4)+5; // $a будет равна 9, а $b=4
$a += 5; //аналогично $a = $a + 5;
$b = "Привет ";
$b .= "всем!";
// аналогично $b="Привет всем!"
echo "a = $a <br>";
echo "b = $b <br>";
?>
```

1.6 String (сөз тіркесі) типі

Сөз тіркесі – бұл символдар тізбегі. PHP тілінде символ дегеніміз – бір байт көлеміндегі мәлімет, яғни ол әр түрлі 256 символдардан тұра алады. Бұл PHP тілінің Unicode жүйесін сүйемелдемейтінін білдіреді. PHP тілінде *сөз тіркесінің* ұзындығына шек қойылмайды, сондықтан олардың ұзындығы жайлы ойламай жұмыс істей беруге болады.

PHP тілінде *сөз тіркесі* 3 түрлі тәсілмен:

- ★ *жалқы тырнақша* (апостроф) көмегімен;
- ★ *қос тырнақша* көмегімен;
- ★ *heredoc-синтаксисі* арқылы анықталады.

Сөз тіркесін анықтаудың ең қарапайым тәсілі – бұл оны ' *апостроф* немесе " *қостырнақша* ішіне алып жазу. Бұл екеуі арасындағы негізгі айырмашылық – қостырнақша ішіндегі айнымалы идентификаторының мәні шығарылады да, апостроф ішіндегі айнымалы идентификаторының аты шығарылады. Мысалы, мына екі жол нәтижесі бірдей:

```
$food = "et";
```

```
$food = 'et' ;
```

Ал келесі екі жол арасында айырмашылықтар бар:

```
$tagam1 = "Менің жақсы көретінім - $food";
```

```
$tagam2 = 'Менің жақсы көретінім - $food' ;
```

\$tagam1 айнымалысына мынадай мән беріледі:

Менің жақсы көретінім - et

\$tagam2 айнымалысының мәні:

Менің жақсы көретінім - \$food

PHP тілінде кері слэш таңбасынан басталатын басқару комбинациялары да қолданылады, олардың арнайы мәндері бар. Мысалы:

Символдар тіркесі	Мағынасы
\n	Жаңа жолға көшу
\r	Курсорды жол басына көшіру
\t	Горизонталь табуляция
\\	Кері қиғаш сызық
\\$	Доллар белгісі

Қостырнақша ішінде бұлар мағынасына сәйкес басқару символдары болады да, ал апостроф ішінде тек “\” және “\\” тізбегі ғана өңделеді.

Жалқы тырнақша арқылы анықтау

Сөз тіркесін анықтаудың ең қарапайым тәсілі – бұл оны ' *жалқы тырнақша* ішіне алу. Ал сол сөз тіркесі ішінде жалқы тырнақшаны пайдалану үшін, Си тіліндегі тәрізді, оның алдына кері қиғаш сызықша (слэш) таңбасын «\» қою керек, яғни оны экрандау қажет. Егер кері сызықша жалқы тырнақшаның алдында немесе сөз тіркесінің соңында тұруы керек болса, онда ол қайталанып жазылуы тиіс «\\».

Егер жалқы тырнақша ішіндегі сөз тіркесінде кез келген басқа таңбалар алдында кері слэш «\» кездессе, онда ол қарапайым символ сияқты анықталады және өзгеріссіз шығарылады. Сондықтан кері слэш таңбасын тек ол сөз тіркесінің соңында, яғни жабылатын тырнақша алдында тұрғанда ғана экрандау керек.

PHP тілінде кері слэш таңбасынан басталатын басқару комбинациялары апострофтар ішінде кездескенмен өнделмейді.

6 мысал. *Басқару тізбектерін пайдалану*

```
<?php
```

```
echo 'Вы можете вставлять в строки символ новой строки как <br>,
      это считается нормальным переходом к сл. строке';
```

```
// Чтобы вывести ' надо перед ней поставить \
```

```
echo 'Чтобы вывести \' надо перед ' .
```

```
'ней поставить \\' ;
```

```
// Сл.строка выведет: Вы хотите удалить C:\*.*?
```

```
echo '<br> Вы хотите удалить C:\*.*?';
```

```
// Сл.строка выведет: Это не вставит: \n новую строку
```

```
echo '<br> Это не вставит: \n новую строку';
```

```
// Сл.строка: Переменные $exprand также $either не подставляются
echo '<br>Переменные $exprand также $either ' .
    'не подставляются'
?>
```

Heredoc-синтаксисін қолдану

Сөз тіркесін анықтаудың басқа бір тәсілі – бұл *heredoc-синтаксисін* пайдалану. Мұнда сөз тіркесі <<< символдарынан басталып, осыдан кейін идентификатор тұрады.

Сөз тіркесі аяқталарда да осы идентификатор қайта жазылады. Жабылатын идентификатор осы жолдағы бірінші сөз болуы қажет. Идентификатор PHP тіліндегі басқа атаулар сияқты жазылуы тиіс: тек әріптерден басталып, сандар мен әріптер (астын сызу таңбасынан да) тізбегінен тұрады.

Heredoc-мәтін тура қос тырнақша ішіндегі сөз тіркесі сияқты болады, бірақ қостырнақшаны жазу қажет емес. Сол себепті тырнақшаны экрандау да керек емес. Бірақ мұнда да жоғарыда айтылған басқару тізбектерін пайдалануға болады. Heredoc тізбегі ішінде тұрған айнымалы мәні де шығарылады.

7 мысал. Heredoc-синтаксисін пайдалану

```
<?php
$str = <<<EOD
Пример строки в несколько строчек
с использованием heredoc-синтаксиса
EOD;
// ЗДЕСЬ идентификатор - EOD. Ниже идентификатор EOT
$name = "Бекзат";
$name2="Азат";
echo <<<EOT
Его зовут "$name", а другого "$name2".
EOT;
// Выводит: Его зовут "Бекзат", а другого "Азат".
?>
```

Тапсырмалар

1 мысал. PHP тіліне код енгізіп орындау

```
<html> <head> <title> Listing 1 </title> </head>
<body>
<?php echo "Здравствуй! Это мой скрипт!"; ?>
</body>
</html>
```

2 мысал. Кесте шығару:

```
<?php
print "<table align=\"center\"><tr><td colspan=\"2\" align=\"center\">";
print "Заголовок таблицы</td></tr>";
print "<tr><td>Что-нибудь</td><td>Еще что-нибудь</td></tr></table>";
?>
```

НЕМЕСЕ мынадай код:

```
<?php
print "<table align=\"center\"><tr><td colspan=\"2\" align=\"center\">
Заголовок таблицы</td></tr>
<tr><td>Что-нибудь</td><td>Еще что-нибудь</td></tr></table>";
?>
```

Былай жазсақ ше:

```
<html>
<head>
<title>Что-нибудь</title>
</head>
<body>
<table align="center">
<tr>
<td colspan="2" align="center">
<?php echo "Заголовок таблицы"; ?>
</td>
</tr>
<tr>
<td><?php echo "Что-нибудь"; ?></td>
<td><?php echo "Еще что-нибудь"; ?></td>
</tr>
</table>";
</body>
</html>
```

Нәтижелері бірдей, неге?

3 мысал. Сандар типін және мәнін шығару:

```
$large_number = 2147483647;
var_dump($large_number);
// на выходе: int(2147483647)
$large_number = 2147483648;
var_dump($large_number);
// на выходе: float(2147483648)
// это подходит также для 16-ричных чисел:
var_dump( 0x80000000 );
// на выходе: float(2147483648)
$million = 1000000;
$large_number = 50000 * $million;
var_dump($large_number);
// на выходе: float(500000000000)
```

4 мысал. Сандар шығару:

```
<?php
$a = 1234;
$b = -1.2e3;
$c = 0123; // восьмеричное число 83 в 10-ой системе
$d = 0x1A; // шестнадцатеричное число 26 в 10-ой системе
echo "a = $a <br>";
echo "b = $b <br>";
echo "c = $c <br>";
echo "d = $d <br>";
?>
```

5 мысал. Сөз тіркестерін шығару:

```
<?php
echo 'Также вы можете вставлять в строки символ новой строки таким
      образом, поскольку это нормально<br>';
// Выведет: Чтобы вывести ` надо перед ней поставить \
echo 'Чтобы вывести ` надо перед ней поставить ` `';
//Выведет: Вы хотите удалить C:\*.*?
echo '<br>Вы хотите удалить C:\\*.*?';
//Выведет: Это не вставит: \n новую строку
```

```
echo '<br>Это не вставит: \n новую строку';
// Выведет: Переменные $exrand также $either не подставляются
echo '<br>Переменные $exrand также $either не подставляются';
?>
```

6 мысал. Қос тырнақшалар мен апострофтар қолдану:

```
<?php
$text="2+2=";
$correct="4";
print "2+2=$correct";
print "<br>";
print $text.$correct;
?>
```

Нәтижесі:

2+2=4

2+2=4

Апострофтарды пайдалану мысалы:

```
<?php
$text="2+2=";
$correct="4";
print '2+2=$correct';
print "<br>";
print $text.$correct;
?>
```

Нәтижесі:

2+2=\$correct

2+2=4

Келесі мысал:

```
<?php
$one="1"; // Это строка
$two="4"; // И это строка
$three=5; // А это число
$result1=$one+$two+$three;
$result2=$one.$two.$three;
print $result1;
print "<br>";
print $result2;
?>
```

Нәтижесі:

10

145

Неге бұлай, түсіндіріңіздер.

СҰРАҚТАР

1. PHP тілінде программа қалай жазылады және орындалады?
2. Константаларды анықтау және олардың мәндерін алу.
3. Экранға мәлімет шығару жолдары.
4. Комментарийлер түрлері және оларды пайдалану.
5. Логикалық тип және оны пайдалану.
6. PHP тіліндегі сандар типтері және оларды пайдалану.
7. Тіркестік мәліметтерлі анықтау тәсілдері.
8. Сөз тіркестеріндегі қос тырнақшалар мен апострофтар айырмашылығы.
9. Heredoc-синтаксисті пайдалану тәсілі.