**Лекция 15.**

**Тема: Ссылки**

В PHP есть два вида ссылок: жёсткие и символические. Жёсткая ссылка на переменную - это просто синоним.

**Пример** жёсткой ссылки.

$a = 5;

$b = &$a; //$b - синоним (жёсткая ссылка) переменной $a

print "\$b = $b <BR>"; //напечатается $b = 5

$a = 10;

print "\$a = $a, \$b = $b <BR>"; //напечатается $a=10, $b = 10

Символические ссылки весьма необычны. Приведём сначала пример символической ссылки, а потом дадим разъяснения.

**Пример** символической ссылки

$a = 10;

$z ='a';

//$z - скалярная переменная, а $$z - символическая ссылка на $a

print "\$z = $z, \$\$z =".$$z."<BR>";

//напечатается "$z = a, $$z =10"

Имя переменной состоит из префикса "$" и идентификатора. В примере значение переменной $z совпадает с идентификатором переменной $a. Даже если это совпадение произошло случайно, $z одновременно и просто переменная и символическая ссылка на переменную $a, т. е. $$z - это $a. Выполнение конструкции $$z можно считать состоящей из двух этапов:

1. Конкатенация

'$'.$z

Получается строка "$a"

2. Cтрока "$a" интерпретируется (интерполируется) как переменная $a, т.е.

вместо неё подставляется значение (в примере $a=10).

**Передача значения по ссылке**

Если в функции нужно изменять при любом обращении одни и те же глобальные переменные, то достаточно воспользоваться объявлением *global*, как это сделано в приведённом выше [примере](https://i.voenmeh.ru/kafi5/Kam.loc/PHPzadan/LEKCII_PHP.html#glob). А как поступить, если в функции при разных обращениях нужно изменять разные глобальные переменные? Можно так:

$a=5;$b=2; $c=1;

list($a,$b,$c)=F1($a,$b,$c);

print "a=$a,b=$b, c=$c<BR>";

$d=50;$e=20; $f=10;

list($d,$e,$f)=F1($d,$e,$f);

print "d=$d,e=$e, f=$f<BR>";

function F1($x,$y,$z)

{ $x--;$y++;$z++;

return array($x,$y,$z);

}

/\*Результат

a=4, b=3, c=2

d=49, e=21, f=11

\*/

Перепишем этот пример так:

$a=5;$b=2; $c=1;

F1($a,$b,$c);

print "a=$a,b=$b,c=$c<BR>";

$d=50;$e=20; $f=10;

F1($d,$e,$f);

print "d=$d,e=$e,f=$f<BR>";

function F1(&$x,&$y,&$z)

{ $x--;$y++;$z++;

}

/\*Результат

a=4,b=3, c=2

d=49, e=21, f=11

\*/

В новом варианте примера &$x означает, что на время выполнения функции F1() локальная переменная $x служит синонимом глобальной переменной из списка фактических параметров. При первом обращении к F1() $x ≡ $a, а при втором $x ≡ $d. Таким образом, префикс & делает формальный параметр жёсткой ссылкой на соответствующий фактический.

**Символическая ссылка на функцию**

В PHP, если переменная содержит имя функции, то можно вызывать функцию, используя вместо имени функции имя переменной, например, если $y = 'sin', то вместо sin($x), можно написать $y($x).

**Пример**

$func ='sin';

$y = 30;

$x = $y/180\*pi();

//1. Выполнение функции, имя которой задано в переменной

$z = $func($x);

print"$func($y<SUP>o</sup>)= $z<BR>"; //sin(30o) = 0.5

//2. Передача имени функции в качестве параметра в другую функцию

print ex\_val($func,$x)."<BR>"; //1.5

print ex\_val('sin',$x)."<BR>"; //1.5

function ex\_val($f,$t)

{ $x = $f($t)+1;

return $x;

}

**Функция eval() и символические ссылки**

Параметром функции eval() служит выражение, задаваемое в виде строки. Это выражение интерполируется и выполняется.

**Пример**

$f ='sin';

$x=30;

eval("\$z=$f($x/180\*pi());");

//Выражение в кавычках в результате интерполяции примет вид оператора

// $z=sin(30/180\*pi();

// Этот оператор выполнится и получится $z=0.5

print "$z<BR>"; //$z=0.5

С помощью функции eval пользователь может вносить изменения в программу в процессе её выполнения. Этим свойством eval() могут воспользоваться и люди с недобрыми намерениями. Ниже приведён пример сайта, состоящего из двух страниц. Первая страница (рис. 2) служит для ввода пользователем фамилии и оклада

|  |
| --- |
| https://i.voenmeh.ru/kafi5/Kam.loc/PHPzadan/eval_1.png |
| Рис. 2. Форма для ввода |

На вторую страницу (рис. 3) выводятся результаты работы скрипта. Скрипт просто выводит полученные данные.

|  |
| --- |
| **Использование eval() для приёма параметров**  Фамилия - Петров Оклад 40000 рублей |
| Рис. 3. Форма для вывода результатов |

*Текст (код) первой HTML-страницы*

<HTML><HEAD>

<meta http-equiv="Content-Type" content="text/html; charset=WINDOWS-1251">

<TITLE>eval и симв. ссылки</title>

</head>

<BODY>

<h2>Функция eval() и символические ссылки</h2>

<FORM METHOD=GET action ='eval\_.php'>

ФИО . . . . <INPUT type=text name=FIO><BR>

Оклад . . . <INPUT type=text name=x><BR>

<INPUT type=submit value='выполнить'>

</form>

</body>

</html>

*Текст (код) второй HTML-страницы с PHP-скриптом*

<html><body>

<h2>Использование eval() для приёма параметров</h2>

<?php

//В цикле перебираются все поступившие с браузера параметры

//В данномм примере всего два параметра FIO и x

foreach ($\_GET as $ke =>$val)

{ eval("\$$ke='$val';");}//Результат интерполяции выражения

//"\$$ke='$val';" при $ke='FIO' $val='Петров'-

//$FIO='Петров'. Этот оператор выполнится

//функцией eval

print "Фамилия=$FIO<BR>оклад=$x рублей";

?>

</body></html>

Строка адреса в браузере при отображении второй страницы будет иметь вид

http://primery/PHPkucha/eval\_.php?FIO=%CF%E5%F2%F0%EE%E2&x=45

После знака вопроса перечисляюся пары "имя переменной= значение".Пары разделяются знаком амперсанта. (%CF - код буквы П в кодировке windows-1251, %E5 - е и %CF%E5%F2%F0%EE%E2 - 'Петров'). Попробуем похулиганить. Изменим значения параметров в адресной строке браузера следующим образом

http://primery/PHPkucha/eval\_.php?FIO='.sin(30/180\*pi()).'&x=XA-XA-XA

и перезагрузим страницу. Результат на рис. 4.

|  |
| --- |
| **Использование eval() для приёма параметров**  Фамилия - 0.5 Оклад - XA-XA-XA |
| Рис. 4. Результаты изменения значений параметров |

Это конечно шутка, мелкое хулиганство, но при определённых навыках таким приёмом можно нанести серьёзный вред. Можно заставить скрипт выполнить любую последовательность команд. Чтобы не дать злоумышленнику этой возможности, заменим eval() на символическую ссылку:

foreach ($\_GET as $ke =>$val)

{ //eval("\$$ke='$val';");

$$ke=$val;

}

В окно браузера будет выведена испорченная информация

Фамилия - '.sin(30/180\*pi()).'

Оклад - XA-XA-XA рублей ,

но её будет видеть только злоумышленник, а функция sin() в нашем примере выполнена не будет. В общем случае не будет выполнена введённая в адресную строку последовательность команд.

**Пример использования символических ссылок**

Задача. Вводится римскими цифрами век. Нужно напечатать, кто царствовал в этом веке. Например, ввели "XVI", нужно вывести  
"В XVI веке царствовал Иван Васильевич".

Не будем показывать HTML-страницу с вводом века. Рассмотрим скрипт, решающий поставленную задачу

<HTML>

. . . . .

<?php

$vek=$\_GET['vek']

$XVI="Иван Васильевич";

$XVIII="Пётр Алексеевич";

$XIX="Николай Павлович";

//Если $vek="XIX", то $$vek - ccылка на переменную $XIX

print "В $vek веке царствовал ". $$vek."<BR>";

//В XIX веке царствовал Николай Павлович

?>

</body>

</html>