

Конструкции языка PHP

Выполнил: ст.гр. ЭАС-305

Михайлова Т.Д.

Проверил: Файзрахманова К.Э.



Основные группы управляющих конструкций РНР

- Условные операторы
- Циклы
- Конструкции выбора
- Конструкции объявления
- Конструкции возврата значений
- Конструкции включений
- Альтернативный синтаксис
конструкций РНР

Инструкции, рассмотренные в презентации:

- Инструкция if-else
- Цикл while
- Цикл for
- Цикл foreach
- Инструкции break и continue
- Конструкция switch-case
- Инструкции require и include

Инструкция if-else

if (логическое_выражение) инструкция_1;

```
<?php  
if (логическое выражение) оператор;  
?>
```

if (логическое_выражение)

инструкция_1;

else

инструкция_2;

```
<?php  
if ($a > $b) {  
    echo "а больше, чем b";  
} else {  
    echo "а НЕ больше, чем b";  
}  
?>
```

Инструкция if-else

if (логическое_выражение_1)

оператор_1;

elseif (логическое_выражение_2)

оператор_2;

else

оператор_3;

```
<?php
if ($a > $b) {
    echo "а больше, чем b";
} elseif ($a == $b) {
    echo "а равен b";
} else {
    echo "а меньше, чем b";
}
?>
```



Использование альтернативного синтаксиса

<?if (логическое_выражение):?>

...HTML-код...

<?else (другое_логическое_выражение):?>

...HTML-код...

<?endif?>

```
<?if(@$name):?>
Привет, <?=$name?>!
<?else:?>
<form action=<?=$REQUEST_URI?> method=get>
Ваше имя: <input type=text name=name><br>
<input type=submit value="Отослать!">
<?endif?>
```

Циклы в PHP

- Цикл с предусловием (**while**);
- Цикл с постусловием (**do-while**);
- Цикл со счетчиком (**for**);
- Специальный цикл перебора массивов (**foreach**).

Цикл с предусловием while

while (логическое_выражение)
инструкция;

```
<?php
$x=0;
while ($x<10)
{
    $x++; // Увеличение счетчика
    echo $x;
}
// Выводит 12345678910
?>
```

Альтернативный синтаксис для цикл с предусловием while

```
<?while (логическое_выражение):?>  
...HTML-код...  
<?endwhile;?>
```

```
<?while($x<10):?>  
<h2><?echo $x; $x++;?></h2>  
<?endwhile;?>
```

Цикл с постусловием do while

```
do
{
тело_цикла;
}
while (логическое_выражение);
```

```
<?php
$x = 1;
do {
    echo $x;
} while ($x++<10);
?>
```

Рассмотренный сценарий выведет на экран:
12345678910



Универсальный цикл for

for (инициализирующие_команды;
условие_цикла; команды_после_итерации) {

```
<?php  
for ($x=0; $x<10; $x++) echo $x;  
?>
```

Данный сценарий выводит: 0123456789

```
<?php  
for ($x=0, $y=0; $x<10; $x++, $y++) echo $x;  
// Выводит 0123456789  
?>
```



Альтернативный синтаксис для for

for(инициализирующие_команды;
условие_цикла; команды_после_итерации):
операторы;
endfor;

```
<?for($z=1;$z<=10;$z++):?>  
<h2><?echo $z;?></h2>  
<?endfor;?>
```

Цикл foreach

foreach (массив as \$ключ=>\$значение) команды;

```
<?php
$names["Иванов"] = "Андрей";
$names["Петров"] = "Борис";
$names["Волков"] = "Сергей";
$names["Макаров"] = "Федор";
foreach ($names as $key => $value) {
echo "<b>$value $key</b><br>";
}
?>
```

foreach (массив as \$значение) команды;

```
<?php
$names[] = "Андрей";
$names[] = "Борис";
$names[] = "Сергей";
$names[] = "Федор";
foreach ($names as $value) {
echo "<b>$value</b><br>";
}
?>
```

Инструкция break

break; // По умолчанию

break(номер_цикла); // Для вложенных циклов
(указывается номер прерываемого цикла)

```
<?php
$x=0;
while ($x++<10) {
if ($x==3) break;
echo "<b>Итерация $x</b><br>";
}
// Когда $x равен 3, цикл прерывается
?>
```

Рассмотренный сценарий выводит:

```
Итерация 1
Итерация 2
```

Инструкция continue

```
<?php
$x=0;
while ($x++<5) {
  if ($x==3) continue;
  echo "<b>Итерация $x</b><br>";
}
// Цикл прервется только на третьей итерации
?>
```

Рассмотренный сценарий выводит:

```
Итерация 1
Итерация 2
Итерация 4
Итерация 5
```

Конструкция switch-case

```
switch(выражение) {  
case значение1: команды1; [break;]  
case значение2: команды2; [break;]  
...  
case значениеN: командыN; [break;]  
[default: команды_по_умолчанию; [break]]  
}
```

```
<?php  
$x="Яблоко";  
switch ($x) {  
case "Яблоко":  
    echo "Это Яблоко";  
    break;  
case "Груша":  
    echo "Это Груша";  
    break;  
case "Арбуз":  
    echo "Это Арбуз";  
    break;  
}  
?>
```

Альтернативный синтаксис switch-case

switch (выражение):

case значение1: команды1; [break;]

...

case значениеN: командыN; [break;] [default:
команды_по_умолчанию; [break;]]

endswitch;

Конструкция declare

- declare (директива)
инструкция;

```
<?php
// Вы можете равнозначно использовать следующие методы

// Так:
declare(ticks=1) {
    // Здесь полный сценарий
}

// Или так:
declare(ticks=1);
// Здесь полный сценарий
?>
```

Конструкция return

```
<?php
function retfunct()
{
    return 7;
}
echo retfunct(); // выводит '7'.
?>
```

```
<?php
function numbers()
{
    return array (0, 1, 2);
}
list ($zero, $one, $two) = numbers();
echo $zero;
echo $one;
echo $two;
// выводит '012'
?>
```

Инструкции require и include

require имя_файла;
имя_файла;

include

```
Файл header.html:  
<html>  
<head><title>It is a title</title></head>  
<body bgcolor=green>
```

```
Файл footer.html:  
&copy; Home Company, 2005.  
</body></html>
```

```
Файл script.php  
<?php  
require "header.htm";  
// Сценарий выводит само тело документа  
require "footer.htm";  
?>
```

```
<?php  
// Создаем цикл, в теле которого конструкция include  
for($i=1; $i<=10; $i++) {  
include "$i.txt";  
}  
// Включили десять файлов: 1.txt, 2.txt, 3.txt ... 10.txt  
// Результат - вывод 12345678910  
?>
```

Инструкции однократного включения `require_once` и `include_once`

```
<?php  
include_once("a.php"); // включит файл a.php  
include_once("A.php"); // включит файл a.php  
require_once("a.php"); // включит файл a.php  
require_once("A.php"); // включит файл a.php  
?>
```