

# #7 – Цикл for, while и do while.

## Операторы циклов

За счет циклов вы можете создавать код, что будет циклично выполнен определенное количество раз. Мы в уроке рассмотрим создание циклов в языке PHP. Мы изучим циклы: for, while, do while и foreach. Дополнительно нами будут изучены операторы для циклов.

В языке **PHP**, как и в большинстве других языков, существует 3 основных вида циклов. Каждый из них выполняет одну и ту же роль, но записывается по-разному. Рассмотрим все три цикла.

### Цикл For

В цикле `for` все условия записываются в одном месте, что очень удобно во многих случаях. Стандартная запись такого цикла выглядит следующим образом:

```
for ($i = 0; $i < 10; $i++)  
    echo $i;
```

В объявлении цикла записывается следующее: переменная цикла, её начальное значение и диапазон. В примере выше будут выведены числа от 0 до 10.

Если в теле цикла всего одна строка кода, то фигурные скобки можно пропустить и не записывать их.

### Цикл While

Суть цикла `while` не особо отличается от цикла `for`. Единственное отличие заключается в способе записи цикла. В `while` необходимо прописать лишь условие, а все остальные параметры записываются вне цикла:

```
$i = 1; // Создание переменной  
while ($i <= 10) { // Здесь только условие  
    echo $i;  
    $i++; // Увеличение переменной  
}
```

### Цикл Do While

Цикл схож с циклом `while` по форме написания, но при этом работает немного по-другому. Цикл **do..while** будет выполнен один раз сто процентов, а дальше проверит условие и если оно верно, то цикл будет выполняться дальше:

```
$x = 13;
```

```
do {  
    $x--;  
    echo $x;  
} while ($x > 10);
```

Как видно из примера, цикл изначально неверный, но это не мешает ему сработать один раз.

## Операторы для работы в циклах

Существует три основных оператора для работы в циклах:

- Оператор `break` - служит для выхода из цикла полностью;
- Оператор `continue` - пропускает лишь одну итерацию и не выходит из цикла.

## Исходный код

### Операторы в циклах

```
for($el = 100; $el > 10; $el /= 2) {  
    if($el < 15)  
        break;  
  
    if($el % 2 == 0)  
        continue;  
  
    echo $el.'<br>';  
}
```

### Вывод массива

```
$list = [5, 7, 3, 8, "some", 45.7];  
  
for($i = 0; $i < count($list); $i++)  
    echo "Element $i: $list[$i]<br>";
```

### Цикл «foreach»

```
<?php  
$list = ["age" => 45, "name" => "Alex", "hobby" => "Football"];  
$arr = [5, 6, 8, 9];  
  
//     foreach ($list as $item => $value) {  
//         echo "Key: $item. Value: $value.<br>";  
//     }  
foreach ($arr as $i => $value)  
    echo "Index: $i. Value: $value.<br>";  
?>
```

### Примеры циклов

```
<?php  
// Цикл For  
// for($i = 100; $i >= 20; $i -= 5)
```

```
//      echo $i.'<br>';

// Цикл While
//      $isHasHouse = true;
//      while($isHasHouse) {
//          echo "Это бесконечный цикл, не стоит его запускать!";
//      }
// Цикл Do While
//      $i = 100;
//      do {
//          echo $i;
//      } while($i < 10);
?>
```

## Самостоятельная работа

### Вывести в цикле

Выведите столбец чисел от 16 до 23 с пропуском числа 21. Используйте цикл for, а также сделайте тоже самое с циклом while.

### Вывод в цикле

Выведите столбец чисел от 11 до 33. Используйте цикл for, а также сделайте тоже самое с циклом while.

Посмотреть ответ

### Пропуск данных

Выведите числа от -30 до 0.

При выводе отображайте лишь каждое третье число с пропуском чисел -27, -21 и -15.

Сделайте вывод чисел в циклах «for» и «do while».