

CODECRAFT

PHP

Урок 4

PHP цикл `do...while`

```
do {  
    выполнять код;  
} while (условие правильное);
```

В отличие от *while* в функции *do..while* код сначала выполняется а потом `$x` увеличивается и проверяется условие

Пример :

```
<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++; // в каждом цикле значение $x увеличивается на 1
} while ($x <= 5); // и после выполнения кода если $x
превышает
//5 цикл заканчивается
?>
```

```
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
```

Пример :

Например в данном случае, $x = 6$ уже задан, но все же условие **do...while** сначала заходит в цикл, выполняет код, а потом проверяет условие.

То есть независимо от условия код внутри цикла **сначала выполняется**, а **потом проверяется** условие

```
<?php
$x = 6;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```

```
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
```

PHP 5 циклы "for"

В отличии от цикла **while**, цикл **for** повторно выполняет один и тот же код определенное количество раз. То есть мы знаем заранее сколько раз код должен быть выполнен.

Структура:

```
for (базовый счетчик; условие для счетчика; увеличить значение счетчика) {  
    выполнить код;  
}
```

Счетчик считает сколько раз должен быть выполнен код.

Базовый счетчик - задает начальное значение счетчика.

Условие для счетчика - проверяет условие в каждом цикле. И если условие неправильное цикл останавливается.

увеличить значение счетчика - если условие правильное увеличить или же уменьшить значение счетчика.

Более подробно рассмотрим позже ---□>>>>>>

Пример :

```
// цикл от 0 до 10
```

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

счетчик начинается с 0

если значение меньше либо равно 10 повторять цикл

увеличить значение счетчика если условие правильное

```
The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5
```

Цикл PHP "foreach"

В отличие от цикла `for`, цикл `foreach` работает только для перебора массивов.

И код выполняется в цикле для каждого элемента массива. Для выполнения `foreach` нам требуется ключ и значение каждого элемента массива.

Ключ - название массива


```
foreach ($array as $value) {  
    code to be executed;  
}
```

значение - каждого элемента массива

Пример на следующей странице --->>>

Пример :

```
<?php
$colors = array("red", "green", "blue", "yellow");
foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```



```
red
green
blue
yellow
```

PHP - Функции

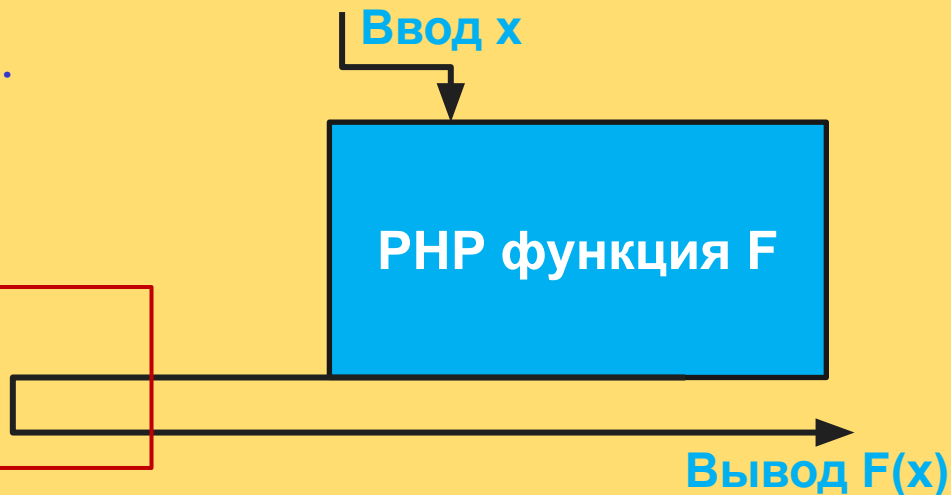
Функция – это просто блок кода, которые могут быть использованы повторно в программе.

*PHP имеет более 1000 встроенных функций.

- **Функция** не будет выполняться сразу после загрузки страницы.
- Для этого надо его **вызвать**

Синтаксис:

```
function названиеФункции() {  
    код. . . ;  
}
```



PHP - ФУНКЦИИ

Синтаксис:

```
function названиеФункции() {  
    код. . . ;  
}
```

Пример:

```
<?php  
function myFunc() {  
    echo "Hello world!";  
}  
  
myFunc(); // Вызов функции myFunc()  
?>
```

PHP - Функции с аргументами (параметрами)

Синтаксис:

```
function названиеФункции($параметр1, $параметр2, ...) {  
    код. . .;  
}
```

Пример:

```
<?php  
function codeCraftFunc($courseName) {  
    echo "Codecraft $courseName курсы.<br>";  
}  
  
codeCraftFunc("Web");  
codeCraftFunc("PHP");  
codeCraftFunc("Scratch");  
codeCraftFunc("App Inventor");  
?>
```

PHP - Функции с аргументами (параметрами)

Синтаксис:

```
function названиеФункции($параметр1, $параметр2, ...) {  
    код. . . ;  
}
```

Пример:

```
<?php  
function codeCraftFunc($courseName, $num) {  
    echo "Codecraft $courseName курсы. $num учеников <br>";  
}  
  
codeCraftFunc("Web", "25");  
codeCraftFunc("PHP", "10");  
codeCraftFunc("Scratch", "20");  
codeCraftFunc("App Inventor", "20");  
?>
```

PHP - Функции с аргументами по умолчанию

Синтаксис:

```
function названиеФункции($параметр1=значение1, $параметр2=значение2, ...) {  
    код. . .;  
}
```

Пример:

```
<?php  
function setScore($minScore = 50) {  
    echo "Баллы: $minScore <br>";  
}  
  
setScore(100);  
setScore(); // будет использовать значения 50 по умолчанию  
setScore(85);  
setScore(63);  
?>
```

PHP - Возвращение значения функцией

Синтаксис:

```
function названиеФункции($параметр1=значение1, $параметр2=значение2, ...) {  
    код. . .;  
    return $переменная;  
}
```

Пример:

```
<?php  
function summa($x, $y) {  
    $z = $x + $y;  
    return $z;  
}  
  
echo "5 + 10 = " . summa(5, 10) . "<br>";  
echo "7 + 13 = " . summa(7, 13) . "<br>";  
echo "2 + 4 = " . summa(2, 4);  
?>
```

Домашнее Задание