

#9 – Функции. Область видимости

Функции в PHP выполняют важную роль. Они обеспечивают сокращение кода за счет вынесения его в отдельные места. За урок мы научимся создавать функции и передавать в них значения.

Функции можно назвать небольшими подпрограммами, куда можно вынести повторяющийся код и обращаться к нему, когда это будет нужно. Функции значительно облегчают построение программ, так как нам не надо копировать однотипный код множество раз, а можно просто воспользоваться одной общей функцией.

Многие путают функции и методы и не понимают отличий между ними. На самом деле отличий нет, так как что методы, что функции являются одним и тем же. Функции что записаны вне классов называют функциями, а функции что записаны внутри классов называются методами.

Точно такая же ситуация обстоит с переменным. В классах переменные называются полями, а вне классов - переменными.

В PHP функции создаются при помощи ключевого слова `function`. Каждая функция может иметь какие-либо параметры или же не иметь их вовсе. Функции способны что-либо возвращать в ходе выполнения кода, если это требуется.

Создание функции

На основе всех данных наша функция будет выглядеть следующим образом:

```
function test() {  
    echo "Вывод чего-либо в консоль";  
}
```

Функция выше не принимает никаких параметров и ничего не возвращает. Она просто пишет слово в консоль. Давайте разнообразим функцию и добавим параметр:

```
function test($word) {  
    echo $word;  
}
```

Теперь функция принимает параметр, который будет отображен в консоли.

Если функция должна что-либо вернуть, то прописываем тип данных который будет возвращен. Для возвращения данных используем ключевое слово `return`:

```
function test($some_number) {  
    $some_number *= 2;  
    return $some_number;  
}
```

Область видимости

В PHP есть несколько полей видимости: локальная и глобальная. Если записать глобальную переменную, то такая переменная будет видна повсюду и с ней можно работать отовсюду в документе. Если записать локальную переменную, то такая переменная будет видна лишь в той области, где она записана.

Для создания глобальной переменной её необходимо прописать вне функции, класса или же объекта. Для создания локальных переменных вам достаточно поместить переменную в функцию. Переменная в функции будет видна лишь внутри блока с функцией и нигде более.

Если из функции мы хотим получить доступ к переменной записанной вне функции, то необходимо прописывать ключевое слово `global`:

```
$some = "Тест";
```

```
function getValue() {  
    // echo $some; // Ничего не выведет  
    echo global $some; // Выведет слово "Тест"  
}
```

Помимо этого можно создавать статические переменные, которые не будут становиться пустыми при новом их использовании:

```
function getNewValue() {  
    // Создаем статическую переменную  
    static $count = 0;  
    $count += 4;  
    echo $count;  
}
```

```
getNewValue(); // Выведет 4  
getNewValue(); // Выведет 8  
getNewValue(); // Выведет 12  
// и так далее
```

Исходной код примеров(выполнить):

Сумма элементов

```
<?php  
function summary($arr) {  
    $summa = 0;  
    foreach($arr as $value) {  
        $summa += $value;  
    }  
    return $summa;  
}
```

```
$list = [5, 7, 3];  
echo summary($list). "<br>";  
echo summary([5, 2, 3]). "<br>";  
?>
```

Область видимости

```
<?php  
function info() {  
    global $x;  
    $x = 0;  
}  
  
function click() {  
    static $count;  
    $count++;  
    echo $count. '<br>';  
}  
  
click();  
click();  
click();  
  
// $x = 10;  
// info();  
// echo $x;  
?>
```

Простые функции

```
<?php  
function info($word) {  
    echo "$word<br>";  
}  
  
function math($x, $y) {  
    return $x + $y;  
}  
  
$res_1 = math(4, 6);  
$res_2 = math(5, 6);  
  
info($res_1);  
info($res_2);  
?>
```

Самостоятельная работа

Проверка возраста

Создайте функцию с одним параметром.

Если параметр будет больше за число 18, то возвращайте значение true. Иначе возвращает окно с вопросом: «Родители разрешили?».

Подсчет километров

Создайте функцию, что будет принимать 2 параметра: время и скорость. Функция должна возвращать расстояние, которое проедет пользователь за время что было передано в функцию. Полученный результат выведите в консоль.

Количество вызовов функции

Создайте глобальную переменную со значением ноль. Создайте функцию, которая будет выводить суммарное количество вызовов этой функции.