Пользовательские функции в PHP. Возвращаемые значения.

Пользовательские функции в PHP. Вызов функции и аргументы функции.

В предыдущей статье я описал каким образом можно вызвать PHP функцию, а также как передать в нее аргументы. Далее я расскажу, что может возвращать функция.
В принципе, функции можно условно разделить на два типа: функции возвращающие какие-либо значения, и функции, которые ничего не возвращают. Сначала рассмотрим функции, которые ничего не возвращают, а производят некоторую последовательность действий, к таким функциям можно отнести функцию hellow_user() из предыдущей статьи. Еще один пример – добавление записи в базу данных. Например, у нас есть таблица новостей, в которую надо добавить новость с описанием и названием:

```
<?php
function news_add($name, $description) {
	if (empty($name)) {
		echo 'Название новости не может быть пустым';
	}
	else if (empty($description)) {
		echo 'Текст новости не может быть пустым';
	}
	else {
		mysql_qeury(“INSERT INTO news SET name='$name',
description='$description'”);
	}
}
?>
```

Обратите внимание на то, что я добавил к функции проверку на названия и текста новости, чтобы избежать ошибок при добавлении. Таким образом, функция news_add() ничего не возвращает, а лишь вставляет строку в таблицу новостей.
Второй условный тип функций – функции, возвращающие какие-либо значения. Самый простой пример – сложение двух чисел:

```
<?php
function summ($a=0, $b=0) {
	$c = $a + $b;
	return $c;
}
?>
```

В качестве значения по умолчанию для аргументов функции я задал 0. Теперь если записать:

```
<?php
summ(2, 3);
?>
```

в браузере мы ничего не увидим, поскольку функция всего лишь складывает два числа 2 и 3. Запись:

```
<?php
echo summ(2, 3);
?>
```

Выведет в браузере число 5. Иногда нужно не просто вывести результат работы функции в браузер, а передать его в другую переменную:

```
<?php
$result = summ(2, 3);
echo $result;
?>
```

Такая запись тоже выведет в браузере число 5. Обратите внимание, что функция не может вернуть несколько переменных, т.е. запись:

```
<?php
function summ($a=0, $b=0) {
	$c = $a + $b;
	return $a;
	return $b;
	return $c;
}
?>
```

ошибочна. Чтобы вернуть несколько переменных, необходимо поместить их в массив:

```
<?php function summ($a=0, $b=0) {
	$c = $a + $b;
	return array($a, $b, $c);
}
?>
```

Использование нескольких операторов return возможно в случае, если они вызываются в условных операторах if, например:

```
<?php
function check_string($string) {
	$check_string = 'Проверочная строка';
	if ($string == $check_string) {
		return TДальше нам понадобится JavaScript функция, назовем ее
также user_online(), которая будет с некоторой периодичностью обращаться к PHP файлу, который и будет вызывать функцию user_online(). Для этого, нам понадобится AJAX и JQuery: RUE;
	}
	else {
		return FALSE;
	}
}
?>
```

Данная функция возвращает TRUE если введенная строка совпадает с проверочной, или FALSE в любом другом случае. Использовать такой подход можно при авторизации пользователей в проверке пароля:

```
<?php
if (check_string($password)) {
	echo 'Пароль верный';
}
else {
	echo 'Пароль неверный';
}
?>
```

Обратите внимание на двойной символ = (==) в функции. Неправильное использование знака = в условном операторе всегда приводит к ошибкам в его выполнении, так например такая запись:

```
if ($string = $check_string) {
                    return TRUE;
        }
        else {
                     return FALSE;
        }
```

Всегда будет возвращать TRUE, так как выражение в скобках оператора if присваивает значение $chek_string переменной $string. Всегда обращайте внимание на использование знака = в условном операторе.