# FunctionPHPОписание функций

# Функция explode

**explode --** Разбивает строку на подстроки

## Описание

array **explode** ( string separator, string string)

Возвращает массив строк, полученных разбиением строки string с использованием separator в качестве разделителя.

Если separator - пустая строка (""), **explode()** возвращает **FALSE**. Если separator не содержится в string, то **explode()** возвращает массив, содержащий один элемент string.

<?php  
// Пример 1  
$number  = "num1 num2 num3 num4 num5 num6";  
$numbers = **explode**(" ", $number);  
echo $numbers[0]; // num1  
echo $numbers[1]; // num2

?>

# Цикл перебора массивов foreach

Синтаксис цикла **foreach** выглядит следующим образом:

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

Здесь команды циклически выполняются для каждого элемента массива, при этом очередная пара **ключ=>значение** оказывается в переменных **$ключ** и **$значение**. Приведем пример работы цикла **foreach**:

<?php  
$names["Иванов"] = "Андрей";  
$names["Петров"] = "Борис";  
$names["Волков"] = "Сергей";  
$names["Макаров"] = "Федор";  
**foreach ($names as $key => $value)**

{  
 echo "<b>$value $key</b><br>";  
}  
?>

Рассмотренный сценарий выводит:

Андрей Иванов  
Борис Петров  
Сергей Волков  
Федор Макаров

У цикла **foreach** имеется и другая форма записи, которую следует применять, когда нас не интересует значение ключа очередного элемента. Выглядит она так:

**foreach (массив as $значение)  
команды;**

В этом случае доступно лишь значение очередного элемента массива, но не его ключ. Это может быть полезно, например, для работы с массивами-списками:

<?php  
$names[] = "Андрей";  
$names[] = "Борис";  
$names[] = "Сергей";  
$names[] = "Федор";  
**foreach ($names as $value)**

{  
 echo "<b>$value</b><br>";  
}  
?>

# Конструкция включений require

Конструкция **require** позволяет включать файлы в PHP сценарий **до выполнения сценария** PHP.

Общий синтаксис **require** такой:

**require имя\_файла;**

**При запуске** (именно при запуске, а не при исполнении!) программы интерпретатор просто заменит инструкцию на содержимое файла имя\_файла

# Конструкция включений include

Конструкция **include** предназначена для включения файлов в код сценария PHP во время **исполнения** **сценария PHP**.

В отличие от конструкции [**require**](http://php.su/functions/?require) конструкция **include** позволяет включать файлы в код PHP скрипта во время выполнения сценария. Синтаксис конструкции include выглядит следующим образом:

**include имя\_файла;**

Целесообразнее использовать конструкцию [**require**](http://php.su/functions/?require) там, где не требуется динамическое включение файлов в сценарий, а конструкцию **include** использовать только с целью динамического включения файлов в код PHP скрипта.

*Источник информации*  <http://php.su/functions>