Search

Keyboard Shortcuts

?
: This help

j
: Next menu item

k
: Previous menu item

g p
: Previous man page

g n
: Next man page

G
: Scroll to bottom

g g
: Scroll to top

g h
: Goto homepage

g s
: Goto search
(current page)

/
: Focus search box

Аргументы функции »
« Функции

- Руководство по PHP
- Справочник языка
- Функции

Change language: Russian

Функции, определяемые пользователем

Приведём пример синтаксиса, используемого для описания функций:

Пример #1 Псевдокод для демонстрации использования функций

```
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
echo "Пример функции.\n";
return $retval;
}
?>
```

Внутри функции можно использовать любой корректный PHP-код, в том числе другие функции и даже объявления классов.

Имена функций следуют тем же правилам, что и другие метки в PHP. Корректное имя функции начинается с буквы или знака подчёркивания, за которым следует любое количество букв, цифр или знаков подчёркивания. В качестве регулярного выражения оно может быть выражено так: `^[a-zA-Z_\x80-\xff][a-zA-Z0-9_\x80-\xff]*$`.

Подсказка

Смотрите также Руководство по именованию.

Функции не обязаны быть определены до их использования, *исключая* тот случай, когда функции определяются условно, как это показано в двух последующих примерах.

В случае, когда функция определяется в зависимости от какого-либо условия, например, как это показано в двух приведённых ниже примерах, обработка описания функции должна *предшествовать* её вызову.

Пример #2 Функции, зависящие от условий

```
<?php

$makefoo = true;

/* Мы не можем вызвать функцию foo() в этом месте,
поскольку она ещё не определена, но мы можем
обратиться к bar() */

bar();

if ($makefoo) {
function foo()
{
echo "Я не существую до тех пор, пока выполнение программы меня не достигнет.\n";
}
}

/* Теперь мы благополучно можем вызывать foo(),
поскольку $makefoo была интерпретирована как true */

if ($makefoo) foo();

function bar()
{
echo "Я существую сразу с начала старта программы.\n";
}

?>
```

Пример #3 Вложенные функции

```php
<?php
function foo()
{
function bar()
{
echo "Я не существую пока не будет вызвана foo().\n";
}
}

/* Мы пока не можем обратиться к bar(),
поскольку она ещё не определена. */

foo();

/* Теперь мы можем вызвать функцию bar(),
обработка foo() сделала её доступной. */

bar();

?>
```

Все функции и классы PHP имеют глобальную область видимости - они могут быть вызваны вне функции, даже если были определены внутри и наоборот.

PHP не поддерживает перегрузку функции, также отсутствует возможность переопределить или удалить объявленную ранее функцию.

> **Замечание**: Имена функций регистронезависимы для символов ASCII от A до Z, тем не менее, предпочтительнее вызывать функции так, как они были объявлены.

Функции PHP поддерживают как списки аргументов переменной длины, так и значения аргументов по умолчанию. Смотрите также описания функций func_num_args(), func_get_arg() и func_get_args() для более детальной информации.

Можно вызывать функции PHP рекурсивно.

Пример #4 Рекурсивные функции

```php
<?php
function recursion($a)
{
if ($a < 20) {
echo "$a\n";
recursion($a + 1);
}
}
?>
```

> **Замечание**: Рекурсивный вызов методов/процедур с глубиной более 100-200 уровней рекурсии может вызвать переполнение стека и привести к аварийному завершению скрипта. В частности, бесконечная рекурсия будет считаться программной ошибкой.

User Contributed Notes

There are no user contributed notes for this page.