



- [Downloads](#)
- [Documentation](#)
- [Get Involved](#)
- [Help](#)
- 

[Dutch PHP Conference 2024](#)

[Getting Started](#)

[Introduction](#)

[A simple tutorial](#)

[Language Reference](#)

[Basic syntax](#)

[Types](#)

[Variables](#)

[Constants](#)

[Expressions](#)

[Operators](#)

[Control Structures](#)

[Functions](#)

[Classes and Objects](#)

[Namespaces](#)

[Enumerations](#)

[Errors](#)

[Exceptions](#)

[Fibers](#)

[Generators](#)

[Attributes](#)

[References Explained](#)

[Predefined Variables](#)

[Predefined Exceptions](#)

[Predefined Interfaces and Classes](#)

[Predefined Attributes](#)

[Context options and parameters](#)

[Supported Protocols and Wrappers](#)

[Security](#)

[Introduction](#)

[General considerations](#)

[Installed as CGI binary](#)

[Installed as an Apache module](#)

[Session Security](#)

[Filesystem Security](#)

[Database Security](#)

[Error Reporting](#)

[User Submitted Data](#)

[Hiding PHP](#)

[Keeping Current](#)

[Features](#)

[HTTP authentication with PHP](#)

[Cookies](#)

[Sessions](#)

[Dealing with XForms](#)

[Handling file uploads](#)

[Using remote files](#)

[Connection handling](#)

[Persistent Database Connections](#)

[Command line usage](#)

[Garbage Collection](#)
[DTrace Dynamic Tracing](#)

[Function Reference](#)

[Affecting PHP's Behaviour](#)
[Audio Formats Manipulation](#)
[Authentication Services](#)
[Command Line Specific Extensions](#)
[Compression and Archive Extensions](#)
[Cryptography Extensions](#)
[Database Extensions](#)
[Date and Time Related Extensions](#)
[File System Related Extensions](#)
[Human Language and Character Encoding Support](#)
[Image Processing and Generation](#)
[Mail Related Extensions](#)
[Mathematical Extensions](#)
[Non-Text MIME Output](#)
[Process Control Extensions](#)
[Other Basic Extensions](#)
[Other Services](#)
[Search Engine Extensions](#)
[Server Specific Extensions](#)
[Session Extensions](#)
[Text Processing](#)
[Variable and Type Related Extensions](#)
[Web Services](#)
[Windows Only Extensions](#)
[XML Manipulation](#)
[GUI Extensions](#)

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

[Обращение к функциям через переменные »](#)
[« Аргументы функции](#)

- [Руководство по PHP](#)
- [Справочник языка](#)
- [Функции](#)

Change language: Russian

Возврат значений

Значения возвращаются при помощи необязательного оператора возврата. Возвращаемые значения могут быть любого типа, в том числе это могут быть массивы и объекты. Возврат приводит к завершению выполнения функции и передаче управления обратно к той строке кода, в которой данная функция была вызвана. Для получения более детальной информации ознакомьтесь с описанием [return](#).

Замечание:

Если конструкция [return](#) не указана, то функция вернёт значение `null`.

Использование выражения return

Пример #1 Использование конструкции [return](#)

```
<?php
function square($num)
{
    return $num * $num;
}
echo square(4); // выводит '16'.
?>
```

Функция не может возвращать несколько значений, но аналогичного результата можно добиться, возвращая массив.

Пример #2 Возврат нескольких значений в виде массива

```
<?php
function small_numbers()
{
    return [0, 1, 2];
}
// Деструктуризация массива будет собирать каждый элемент массива индивидуально
[$zero, $one, $two] = small_numbers();

// До версии 7.1.0 единственной эквивалентной альтернативой было использование конструкции list().
list($zero, $one, $two) = small_numbers();

?>
```

Для того, чтобы функция возвращала результат по ссылке, вам необходимо использовать оператор `&` и при описании функции, и при присвоении переменной возвращаемого значения:

Пример #3 Возврат результата по ссылке

```
<?php
function &returns_reference()
{
    return $someref;
}

$newref =& returns_reference();
?>
```

Для получения более детальной информации о ссылках обратитесь к разделу документации [Подробно о ссылках](#).

[+add a note](#)

User Contributed Notes 4 notes

[up](#)

[down](#)

[ryan dot jentsch at gmail dot com ¶](mailto:ryan.dot.jentsch@gmail.com)

7 years ago

PHP 7.1 allows for void and null return types by preceding the type declaration with a ? -- (e.g. function canReturnNullOrString(): ?string)

However resource is not allowed as a return type:

```
<?php
function fileOpen(string $fileName, string $mode): resource
{
    $handle = fopen($fileName, $mode);
    if ($handle !== false)
    {
        return $handle;
    }
}

$resourceHandle = fileOpen("myfile.txt", "r");
?>
```

Errors with:

Fatal error: Uncaught TypeError: Return value of fileOpen() must be an instance of resource, resource returned.

[up](#)

[down](#)

24

[rstaveley at seseit dot com ¶](mailto:rstaveley@seseit.com)

13 years ago

Developers with a C background may expect pass by reference semantics for arrays. It may be surprising that pass by value is used for arrays just like scalars. Objects are implicitly passed by reference.

```
<?php

# (1) Objects are always passed by reference and returned by reference

class Obj {
    public $x;
}

function obj_inc_x($obj) {
    $obj->x++;
    return $obj;
}

$obj = new Obj();
$obj->x = 1;

$obj2 = obj_inc_x($obj);
obj_inc_x($obj2);

print $obj->x . ', ' . $obj2->x . "\n";

# (2) Scalars are not passed by reference or returned as such

function scalar_inc_x($x) {
    $x++;
    return $x;
}

$x = 1;

$x2 = scalar_inc_x($x);
scalar_inc_x($x2);
```

```
print $x . ', ' . $x2 . "\n";
```

(3) You have to force pass by reference and return by reference on scalars

```
function &scalar_ref_inc_x(&$x) {  
    $x++;  
    return $x;  
}
```

```
$x = 1;
```

```
$x2 =& scalar_ref_inc_x($x); # Need reference here as well as the function sig  
scalar_ref_inc_x($x2);
```

```
print $x . ', ' . $x2 . "\n";
```

(4) Arrays use pass by value semantics just like scalars

```
function array_inc_x($array) {  
    $array['x']++;  
    return $array;  
}
```

```
$array = array();  
$array['x'] = 1;
```

```
$array2 = array_inc_x($array);  
array_inc_x($array2);
```

```
print $array['x'] . ', ' . $array2['x'] . "\n";
```

(5) You have to force pass by reference and return by reference on arrays

```
function &array_ref_inc_x(&$array) {  
    $array['x']++;  
    return $array;  
}
```

```
$array = array();  
$array['x'] = 1;
```

```
$array2 =& array_ref_inc_x($array); # Need reference here as well as the function sig  
array_ref_inc_x($array2);
```

```
print $array['x'] . ', ' . $array2['x'] . "\n";
```

[up](#)

[down](#)

10

[bgalloway at citycarshare dot org](#)

15 years ago

Be careful about using "do this thing or die()" logic in your return lines. It doesn't work as you'd expect:

```
<?php  
function myfunc1() {  
    return('thingy' or die('otherthingy'));  
}  
function myfunc2() {  
    return 'thingy' or die('otherthingy');  
}  
function myfunc3() {  
    return('thingy') or die('otherthingy');
```

```
}  
function myfunc4() {  
return 'thingy' or 'otherthingy';  
}  
function myfunc5() {  
$x = 'thingy' or 'otherthingy'; return $x;  
}  
echo myfunc1(). "\n". myfunc2(). "\n". myfunc3(). "\n". myfunc4(). "\n". myfunc5(). "\n";  
?>
```

Only myfunc5() returns 'thingy' - the rest return 1.

[up](#)

[down](#)

8

[nick at itomic.com ¶](#)

20 years ago

Functions which return references, may return a NULL value. This is inconsistent with the fact that function parameters passed by reference can't be passed as NULL (or in fact anything which isnt a variable).

i.e.

```
<?php
```

```
function &testRet()  
{  
return NULL;  
}  
  
if (testRet() === NULL)  
{  
echo "NULL";  
}  
?>
```

parses fine and echoes NULL

[+add a note](#)

- [Функции](#)
 - [Функции, определяемые пользователем](#)
 - [Аргументы функции](#)
 - [Возврат значений](#)
 - [Обращение к функциям через переменные](#)
 - [Встроенные функции](#)
 - [Анонимные функции](#)
 - [Стрелочные функции](#)
 - [Синтаксис callable-объектов первого класса](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

