



- [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

[Область видимости переменной »](#)

[« Основы](#)

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

Change language: Russian

## Предопределённые переменные

Любому запускаемому скрипту PHP предоставляет большое количество предопределённых переменных. Однако многие из этих переменных не могут быть полностью задокументированы, поскольку они зависят от запускающего скрипта сервера, его версии и настроек, а также других факторов. Некоторые из этих переменных недоступны, когда PHP запущен из [командной строки](#). Смотрите [список зарезервированных предопределённых переменных](#) для получения дополнительной информации.

PHP предоставляет дополнительный набор предопределённых массивов, содержащих переменные сервера (если они доступны), окружения и пользовательского ввода. Эти массивы являются особыми, поскольку они становятся глобальными автоматически - то есть, автоматически доступны в любой области видимости. По этой причине они также известны как 'автоглобальные' или 'суперглобальные' переменные. (В PHP нет механизма определяемых пользователем суперглобальных переменных.) Смотрите [список суперглобальных переменных](#) для получения дополнительной информации.

### Замечание: Переменные переменных

Суперглобальные переменные не могут быть [переменными переменных](#) внутри функций или методов класса.

Если некоторые из переменных в [variables\\_order](#) не установлены, соответствующие им предопределённые массивы также останутся пустыми.

[+add a note](#)

## User Contributed Notes 4 notes

[up](#)

[down](#)

111

[johnphayes at gmail dot com ¶](#)

17 years ago

I haven't found it anywhere else in the manual, so I'll make a note of it here - PHP will automatically replace any dots ('.') in an incoming variable name with underscores ('\_'). So if you have dots in your incoming variables, e.g.:

```
example.com/page.php?chuck.norris=nevercries
```

you can not reference them by the name used in the URI:

```
//INCORRECT
```

```
echo $_GET['chuck.norris'];
```

instead you must use:

```
//CORRECT
```

```
echo $_GET['chuck_norris'];
```

[up](#)

[down](#)

12

[DD32=theonly DD32\[&\]yahoo.com.au ¶](#)

17 years ago

I have this function in my main files, it allows for easier SEO for some pages without having to rely on .htaccess and mod\_rewrite for some things.

```
<?php
```

```
function long_to_GET(){
```

```
/**
```

```
* This function converts info.php/a/1/b/2/c?d=4 TO
```

```
* Array ( [d] => 4 [a] => 1 [b] => 2 [c] => )
```

```
**/
```

```
if(isset($_SERVER['PATH_INFO']) && $_SERVER['PATH_INFO'] != ''){
```

```
//Split it out.
```

```
$tmp = explode('/',$_SERVER['PATH_INFO']);
```

```
//Remove first empty item
```

```
unset($tmp[0]);
//Loop through and apend it into the $_GET superglobal.
for($i=1;$i<=count($tmp);$i+=2){ $_GET[$tmp[$i]] = $tmp[$i+1];}
}
}
?>
```

Its probably not the most efficient, but it does the job rather nicely.

DD32

[up](#)

[down](#)

10

[\*\*\*lopez dot on dot the dot lists at yellowspace dot net ¶\*\*\*](#)

**21 years ago**

- Security Issue and workaround -

If You use "eval()" to execute code stored in a database or elsewhere, you might find this tip useful.

Issue:

By default, all superglobals are known in every function.

Thus, if you eval database- or dynamically generated code (let's call it "potentially unsafe code"), it can use all the values stored in any superglobal.

Workaround:

Whenever you want to hide superglobals from use in evaluated code, wrap that eval() in an own function within which you unset() all the superglobals. The superglobals are not deleted by php in all scopes - just within that function. eg:

```
function safeEval($evalcode) {
unset($GLOBALS);
unset($_ENV);
// unset any other superglobal...
return eval($evalcode);
}
```

(This example assumes that the eval returns something with 'return')

In addition, by defining such a function outside classes, in the global scope, you'll make sure as well that the evaluated ('unsafe') code doesn't have access to the object variables (\$this-> ...).

[up](#)

[down](#)

0

[\*\*\*LouisGreen at plig dot freeserve dot co dot uk ¶\*\*\*](#)

**20 years ago**

It seems that when you wish to export a variable, you can do it as return \$variable, return an array(), or globalise it. If you return something, information for that variable can only travel one way when the script is running, and that is out of the function.

```
function fn() {
$variable = "something";

return $variable;
}
```

echo fn();

OR

```
$newvariable = fn();
```

Although if global was used, it creates a pointer to a variable, whether it existed or not, and makes whatever is created in the function linked to that global pointer. So if the pointer was global \$variable, and then you set a value to \$variable, it would then be accessible in the global scope. But then what if you later on in the script redefine that global to equal something else. This means that whatever is put into the global array, the information that is set in the pointer, can be set at any point (overiden). Here is an example that might make this a little clearer:

```
function fn1() {

global $variable; // Pointer to the global array
$variable = "something";
}

fn1();
echo $variable; // Prints something
$variable = "12345";
echo $variable; // Prints 12345

function fn2() {

global $variable; // Pointer to the global array
echo $variable;
}

fn2(); // echos $variable which contains "12345"
```

Basically when accessing the global array, you can set it refer to something already defined or set it to something, (a pointer) such as variable you plan to create in the function, and later possibly over ride the pointer with something else.

[+add a note](#)

- [Переменные](#)
  - [ОСНОВЫ](#)
  - [Предопределённые переменные](#)
  - [Область видимости переменной](#)
  - [Переменные переменных](#)
  - [Переменные извне PHP](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

