



- [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

[include »](#)

[« return](#)

- [Руководство по PHP](#)
- [Справочник языка](#)
- [Управляющие конструкции](#)

Change language: Russian

require

(PHP 4, PHP 5, PHP 7, PHP 8)

require аналогично [include](#), за исключением того, что в случае возникновения ошибки он также выдаст фатальную ошибку уровня **E_COMPILE_ERROR**. Другими словами, он остановит выполнение скрипта, тогда как [include](#) только выдал бы предупреждение **E_WARNING**, которое позволило бы скрипту продолжить выполнение.

Смотрите документацию по [include](#), чтобы узнать как он работает.

[+ add a note](#)

User Contributed Notes 3 notes

[up](#)

[down](#)

147

[chris at chrisstockton dot org ¶](#)

16 years ago

Remember, when using require that it is a statement, not a function. It's not necessary to write:

```
<?php
require('somefile.php');
?>
```

The following:

```
<?php
require 'somefile.php';
?>
```

Is preferred, it will prevent your peers from giving you a hard time and a trivial conversation about what require really is.

[up](#)

[down](#)

32

[Marcel Baur ¶](#)

2 years ago

If your included file returns a value, you can get it as a result from require(), i.e.

```
foo.php:
<?php
return "foo";
?>
```

```
$bar = require("foo.php");
echo $bar; // equals to "foo"
```

[up](#)

[down](#)

0

[jave dot web at seznam dot cz ¶](#)

29 days ago

Always use `__DIR__` to define path relative to your current `__FILE__`.
(Or another setting that is originally based on `__DIR__`/`__FILE__`)

try & catch - don't get confused by the words "fatal E_COMPILE_ERROR" - it's still just an internal Error that implements Throwable - it can be caught:

```
<?php
try {
require(__DIR__ . '/something_that_does_not_exist');
} catch (\Throwable $e) {
```

```
echo "This was caught: " . $e->getMessage();  
}  
echo " End of script.";   
?>
```

Note that this will still emit a warning "Failed to open stream: No such file or directory..." ...unless you prefix the require with "@" - which I strongly don't recommend as it would ignore/supress any diagnostic error (unless you have specified set_error_handler()). But even if you'd prefix the require with "@" it would still be caught.

[+add a note](#)

- [Управляющие конструкции](#)
 - [Введение](#)
 - [if](#)
 - [else](#)
 - [elseif/else if](#)
 - [Альтернативный синтаксис управляющих структур](#)
 - [while](#)
 - [do-while](#)
 - [for](#)
 - [foreach](#)
 - [break](#)
 - [continue](#)
 - [switch](#)
 - [match](#)
 - [declare](#)
 - [return](#)
 - [require](#)
 - [include](#)
 - [require_once](#)
 - [include_once](#)
 - [goto](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

