



- [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

[Выражения »](#)

[« Предопределённые константы](#)

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

Change language: Russian ▾

Магические константы

Есть девять магических констант, которые меняют своё значение в зависимости от контекста, в котором они используются. Например, значение `__LINE__` зависит от строки в скрипте, на которой эта константа указана. Все магические константы разрешаются во время компиляции, в отличие от обычных констант, которые разрешаются во время выполнения. Специальные константы нечувствительны к регистру и их список приведён ниже:

Магические константы PHP	
Имя	Описание
<code>__LINE__</code>	Текущий номер строки в файле.
<code>__FILE__</code>	Полный путь и имя текущего файла с развёрнутыми симлинками. Если используется внутри подключаемого файла, то возвращается имя данного файла.
<code>__DIR__</code>	Директория файла. Если используется внутри подключаемого файла, то возвращается директория этого файла. Это эквивалентно вызову <code>dirname(__FILE__)</code> . Возвращаемое имя директории не оканчивается на слеш, за исключением корневой директории.
<code>__FUNCTION__</code>	Имя функции или <code>{closure}</code> в случае анонимной функции.
<code>__CLASS__</code>	Имя класса. Это имя содержит название пространства имён, в котором класс был объявлен (например, <code>Foo\Bar</code>). При использовании в методах трейтов <code>__CLASS__</code> является именем класса, в котором эти методы используются.
<code>__TRAIT__</code>	Имя трейта. Это имя содержит название пространства имён, в котором трейт был объявлен (например, <code>Foo\Bar</code>).
<code>__METHOD__</code>	Имя метода класса.
<code>__NAMESPACE__</code>	Имя текущего пространства имён.
<code>ClassName::class</code>	Полное имя класса.

Смотрите также

- [::class](#)
- [get_class\(\)](#)
- [get_object_vars\(\)](#)
- [file_exists\(\)](#)
- [function_exists\(\)](#)

[+ add a note](#)

User Contributed Notes 3 notes

[up](#)
[down](#)

3

[Rich](#) 

7 months ago

```
<?php

namespace My\App {
    class Api {
        public static fetch() {
            print __FUNCTION__ . "\n"; // outputs fetch
            print __METHOD__ . "\n"; // outputs My\App\Api::fetch
        }
    }

    Api::fetch();
}

namespace {
    My\App\Api::fetch();
}
```

?>

__METHOD__ outputs a fully qualified method name; __FUNCTION__ when used in a method, outputs just the method name.

[up](#)

[down](#)

2

[chris at ocproducts dot com ¶](#)

9 months ago

Note that __CLASS__ and __METHOD__ both reference the class the code is written in, not whatever the object class is. E.g. if you have an object of class B inheriting from class A, any usage of __CLASS__ in class A is going to give "A".

[up](#)

[down](#)

1

[theking2 at king dot ma ¶](#)

1 year ago

If PHP is run inside a web server request there is an important difference between the __DIR__ constant and \$_SERVER['DOCUMENT_ROOT'].

Where __DIR__ of a PHP script contained within a sub-folder will include the complete server path
\$_SERVER['DOCUMENT_ROOT'] will contain a server path up to the _root_ of the application. This can be helpful when for instance an auto-loader is defined in an include file sitting inside a sub-folder and where the classes are located in another folder at the root of the application.

[+add a note](#)

- [Константы](#)
 - [Синтаксис](#)
 - [Предопределённые константы](#)
 - [Магические константы](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

