Home PHP MySQL Laravel WordPress Magento 2 CodeIgniter

⇧ SCROLL TO TOP

Magic Constants

Magic constants are the predefined constants in PHP which get changed on the basis of their use. They start with double underscore (__) and ends with double underscore.

They are similar to other predefined constants but as they change their values with the context, they are called **magic** constants.

There are **nine** magic constants in PHP. In which eight magic constants start and end with double underscores (__).

1. __LINE__
2. __FILE__
3. __DIR__
4. __FUNCTION__
5. __CLASS__
6. __TRAIT__
7. __METHOD__
8. __NAMESPACE__
9. ClassName::class

All of the constants are resolved at compile-time instead of run time, unlike the regular constant. Magic constants are case-insensitive.

Changelog

Version	Description
5.3.0	Added **__DIR__** and **__NAMESPACE__** magic constant
5.4.0	Added **__TRAIT__** magic constant
5.5.0	Added ::class magic constant

All the constants are defined below with the example code:

1. __LINE__

It returns the current line number of the file, where this constant is used.

Example:

```
<?php
   echo "<h3>Example for __LINE__</h3>";
   // print Your current line number i.e;4
   echo "You are at line number " . __LINE__ . "<br><br>";
?>
```

Output:

```
Example for __LINE__


You are at line number 4
```

2. __FILE__:

This magic constant returns the full path of the executed file, where the file is stored. If it is used inside the include, the name of the included file is returned.

Example:

```
<?php
   echo "<h3>Example for __FILE__</h3>";
   //print full path of file with .php extension
   echo __FILE__ . "<br><br>";
?>
```

Output:

```
Example for __FILE__


D:\xampp\htdocs\program\magic.php
```

3. __DIR__:

It returns the full directory path of the executed file. The path returned by this magic constant is equivalent to dirname(__FILE__). This magic constant does not have a trailing slash unless it is a root directory.

Example:

```
<?php
    echo "<h3>Example for __DIR__</h3>";
    //print full path of directory where script will be placed
    echo __DIR__ . "<br><br>";
    //below output will equivalent to above one.
    echo dirname(__FILE__) . "<br><br>";
?>
```

Output:

```
Example for __DIR__

D:\xampp\htdocs\program

D:\xampp\htdocs\program
```

4. __FUNCTION__:

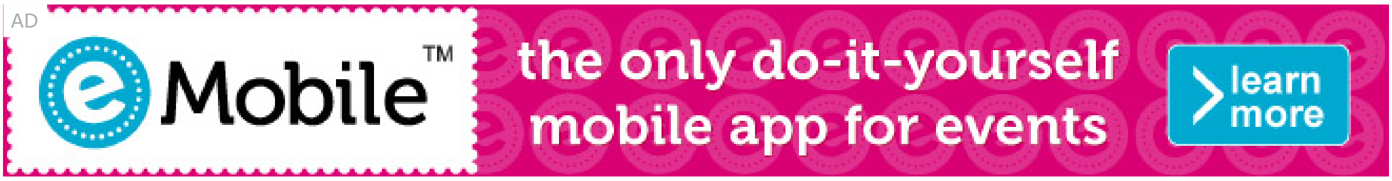
This magic constant returns the function name, where this constant is used. It will return blank if it is used outside of any function.

Example:

```
<?php
    echo "<h3>Example for __FUNCTION__</h3>";
    //Using magic constant inside function.
    function test(){
        //print the function name i.e; test.
        echo 'The function name is '. __FUNCTION__ . "<br><br>";
    }
    test();

    //Magic constant used outside function gives the blank output.
    function test_function(){
       echo 'Hie';
    }
    test_function();
    //give the blank output.
    echo  __FUNCTION__ . "<br><br>";
?>
```

Output:



```
Example for __FUNCTION__


The function name is test


Hie
```

5. __CLASS__:

It returns the class name, where this magic constant is used. __CLASS__ constant also works in traits.

Example:

```
<?php
    echo "<h3>Example for __CLASS__</h3>";
    class JTP
    {
       public function __construct() {
          ;
    }
    function getClassName(){
        //print name of the class JTP.
        echo __CLASS__ . "<br><br>";
        }
    }
    $t = new JTP;
    $t->getClassName();

    //in case of multiple classes
    class base
    {
    function test_first(){
            //will always print parent class which is base here.
            echo __CLASS__;
        }
    }
    class child extends base
    {
        public function __construct() {
            ;
        }
    }
    $t = new child;
    $t->test_first();
?>
```

Output:

```
Example for __CLASS__

JTP

base
```

6. __TRAIT__:

This magic constant returns the trait name, where it is used.

Example:

```
<?php
  echo "<h3>Example for __TRAIT__</h3>";
  trait created_trait {
    function jtp(){
      //will print name of the trait i.e; created_trait
      echo __TRAIT__;
    }
  }
  class Company {
    use created_trait;
    }
  $a = new Company;
  $a->jtp();
?>
```

Output:

```
Example for __TRAIT__

created_trait
```

7. __METHOD__:

It returns the name of the class method where this magic constant is included. The method name is returned the same as it was declared.

Example:

```
<?php
  echo "<h3>Example for __METHOD__</h3>";
  class method {
    public function __construct() {
      //print method::__construct
        echo __METHOD__ . "<br><br>";
      }
    public function meth_fun(){
      //print method::meth_fun
        echo __METHOD__;
    }
  }
  $a = new method;
  $a->meth_fun();
?>
```

Output:

```
Example for __METHOD__

method:: construct
method:: meth_fun
```

8. __NAMESPACE__:

It returns the current namespace where it is used.

Example:

```
<?php
  echo "<h3>Example for __NAMESPACE__</h3>";
  class name {
    public function __construct() {
      echo 'This line will print on calling namespace.';
    }
  }
  $class_name = __NAMESPACE__ . '\name';
  $a = new class_name;
?>
```

Output:

```
Example for __NAMESPACE__

This line will print on calling namespace.
```

9. ClassName::class:

This magic constant does not start and end with the double underscore (__). It returns the fully qualified name of the ClassName. ClassName::class is added in **PHP 5.5.0**. It is useful with namespaced classes.

Example:

```
<?php
    namespace Technical_Portal;
    echo "<h3>Example for CLASSNAME::CLASS </h3>";
    class javatpoint {
    }
    echo javatpoint::class;    //ClassName::class
?>
```

Output:

```
Example for ClassName::class

Technical_Portal\javatpoint
```

> Note: Remember namespace must be the very first statement or after any declare call in the script, otherwise it will generate Fatal error.

← Prev Next →

For Videos Join Our Youtube Channel: Join Now

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share

Learn Latest Tutorials

Splunk	SPSS	Swagger	Transact-SQL
Tumblr	ReactJS	Regex	Reinforcement Learning
R Programming	RxJS	React Native	Python Design Patterns

R Programming

RxJS

React Native

Python Design Patterns

Python Pillow

Python Turtle

Keras

Preparation

Aptitude

Reasoning

Verbal Ability

Interview Questions

Company Questions

Trending Technologies

Artificial Intelligence

AWS

Selenium

Cloud Computing

Hadoop

ReactJS

Data Science

Angular 7

Blockchain

Git

Machine Learning

DevOps

B.Tech / MCA

- DBMS
- Data Structures
- DAA
- Operating System
- Computer Network
- Compiler Design
- Computer Organization
- Discrete Mathematics
- Ethical Hacking
- Computer Graphics
- Software Engineering
- Web Technology
- Cyber Security
- Automata
- C Programming
- C++
- Java
- .Net
- Python tutorial

  Python
- List of Programs

  Programs
- Control Systems tutorial

  Control System
- Data Mining Tutorial

  Data Mining
- Data Warehouse Tutorial

  Data Warehouse