

defined

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

defined — Checks whether a given named constant exists

Description ¶

defined(string \$constant_name): bool

Checks whether the given constant exists and is defined.

Note:

If you want to see if a variable exists, use [isset\(\)](#), as **defined()** only applies to [constants](#). If you want to see if a function exists, use [function_exists\(\)](#).

Parameters ¶

constant_name

The constant name.

Return Values ¶

Returns **true** if the named constant given by constant_name has been defined, **false** otherwise.

Examples ¶

Example #1 Checking Constants

```
<?php
/* Note the use of quotes, this is important. This example is checking
 * if the string 'TEST' is the name of a constant named TEST */
if (defined('TEST')) {
    echo TEST;
}
?>
```

See Also ¶

- [define\(\)](#) - Defines a named constant
- [constant\(\)](#) - Returns the value of a constant
- [get_defined_constants\(\)](#) - Returns an associative array with the names of all the constants and their values
- [function_exists\(\)](#) - Return true if the given function has been defined
- The section on [Constants](#)

 [add a note](#)

User Contributed Notes 16 notes

[up](#)

[down](#)

113

[daniel at neville dot tk ¶](#)

13 years ago

My preferred way of checking if a constant is set, and if it isn't - setting it (could be used to set defaults in a file, where the user has already had the opportunity to set their own values in another.)

```
<?php
```

```
defined('CONSTANT') or define('CONSTANT', 'SomeDefaultValue');
```

```
?>
```