

# define

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

define — Defines a named constant

## Description ¶

**define**(string \$constant\_name, [mixed](#) \$value, bool \$case\_insensitive = **false**): bool

Defines a named constant at runtime.

## Parameters ¶

constant\_name

The name of the constant.

### Note:

It is possible to **define()** constants with reserved or even invalid names, whose value can (only) be retrieved with [constant\(\)](#). However, doing so is not recommended.

value

The value of the constant. In PHP 5, value must be a scalar value (int, float, string, bool, or **null**). In PHP 7, array values are also accepted.

### Warning

While it is possible to define resource constants, it is not recommended and may cause unpredictable behavior.

case\_insensitive

If set to **true**, the constant will be defined case-insensitive. The default behavior is case-sensitive; i.e. **CONSTANT** and **Constant** represent different values.

### Warning

Defining case-insensitive constants is deprecated as of PHP 7.3.0.

### Note:

Case-insensitive constants are stored as lower-case.

## Return Values ¶

Returns **true** on success or **false** on failure.

## Changelog ¶

Version	Description
7.3.0	<code>case_insensitive</code> has been deprecated and will be removed in version 8.0.0.
7.0.0	array values are allowed.

## Examples ¶

### Example #1 Defining Constants

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // outputs "Hello world."
echo Constant; // outputs "Constant" and issues a notice.

define("GREETING", "Hello you.", true);
echo GREETING; // outputs "Hello you."
echo Greeting; // outputs "Hello you."

// Works as of PHP 7
define('ANIMALS', array(
    'dog',
    'cat',
    'bird'
));
echo ANIMALS[1]; // outputs "cat"

?>
```

### Example #2 Constants with Reserved Names

This example illustrates the *possibility* to define a constant with the same name as a [magic constant](#). Since the resulting behavior is obviously confusing, it is not recommended to do this in practise, though.

```
<?php
var_dump(defined('__LINE__'));
var_dump(define('__LINE__', 'test'));
var_dump(constant('__LINE__'));
var_dump(__LINE__);
?>
```

The above example will output:

```
bool(false)
bool(true)
string(4) "test"
int(5)
```

## See Also ¶

- [defined\(\)](#) - Checks whether a given named constant exists
- [constant\(\)](#) - Returns the value of a constant
- The section on [Constants](#)