PHP - Constants

A constant in PHP is a name or an identifier for a simple value. A constant value cannot change during the execution of the PHP script.

- By default, a PHP constant is case-sensitive.
- By convention, constant identifiers are always uppercase.
- A constant name starts with a letter or underscore, followed by any number of letters, numbers, or underscore.
- There is no need to write a dollar sign ($) before a constant, however one has to use a dollar sign before a variable.

Examples of Valid and Invalid Constant Names in PHP

Here are some examples of valid and invalid constant names in PHP –

```
// Valid constant names
define("ONE",     "first thing");
define("TWO2",    "second thing");
define("THREE_3", "third thing");
define("__THREE__", "third value");

// Invalid constant names
define("2TWO",    "second thing");
```

Difference between Constants and Variables in PHP

- Constants cannot be defined by simple assignment; they can only be defined using the define() function.
- Constants may be defined and accessed anywhere without regard to variable scoping rules.
- Once the Constants have been set, they may not be redefined or undefined.

Defining a Named Constant

The define() function in PHP library is used to define a named constant at runtime.

```
define(string $const_name, mixed $value, bool $case = false): bool
```

Parameters

- **const_name** – The name of the constant.
- **value** – The value of the constant. It can be a scalar value (int, float, string, bool, or null) or array values are also accepted.
- **case** – 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.

The define() function returns "true" on success and "false" on failure.

Example 1

The following example demonstrates how the define() function works –

</> Open Compiler

```
<?php
   define("CONSTANT", "Hello world.");

   echo CONSTANT;
   // echo Constant;
?>
```

The first echo statement outputs the value of CONSTANT. You will get the following **output** –

```
Hello world.
```

But, when you uncomment the second echo statement, it will display the following error –

```
Fatal error: Uncaught Error: Undefined constant "Constant" in hello.php: on line 5
```

If you set the case parameter to False, PHP doesn't differentiate upper and lowercase constants.

Example 2

You can also use an array as the value of a constant. Take a look at the following example –

</> Open Compiler

```
<?php
   define(
      $name="LANGS",
      $value=array('PHP', 'Java', 'Python')
   );
   var_dump(LANGS);
?>
```

It will produce the following **output** –

```
array(3) {
  [0]=>
  string(3) "PHP"
  [1]=>
  string(4) "Java"
  [2]=>
  string(6) "Python"
}
```

Using the constant() Function

The echo statement outputs the value of the defined constant. You can also use the constant() function. It returns the value of the constant indicated by name.

```
constant(string $name): mixed
```

The constant() function is useful if you need to retrieve the value of a constant, but do not know its name. I.e. it is stored in a variable or returned by a function.

</> Open Compiler

```php
<?php
   define("MINSIZE", 50);

   echo MINSIZE;
   echo PHP_EOL;
   echo constant("MINSIZE");    // same thing as the previous line
?>
```

It will produce the following **output** –

```
50
50
```

Using the defined() Function

The PHP library provides a defined() function that checks whether a given named constant exists. Take a look at the following example –

</> Open Compiler

```php
<?php
   define('MAX', 100);

   if (defined('MAX')) {
      echo MAX;
   }
?>
```

It will produce the following **output** –

```
100
```

PHP also has a function called "get_defined_constants()" that returns an associative array of all the defined constants and their values.