Constants

This lesson discusses all there is to know about constants, i.e., defining them, checking if they're defined and using them in PHP programs.

We'll cover the following

Defining Constants

Constants, by definition, are variables that cannot be modified during the execution of the script. They are created using the `const` statement or the `define` function.

> **Convention:** In PHP, the convention is to use uppercase letters for constant names, e.g., `A_CONSTANT` .

The following code snippet shows how constants are created:

```
<?php
const PI = 3.14; // float
define("EARTH_IS_FLAT", false); // boolean
const UNKNOWN = null; // null
define("APP_ENV", "dev"); // string
const MAX_SESSION_TIME = 60 * 60; // integer, using (scalar) expressions is ok
const APP_LANGUAGES = ["de", "en"]; // arrays
define("BETTER_APP_LANGUAGES", ["lu", "de"]); // arrays
?>
```

Define constant using explicit values

If you have already defined one constant, you can define another one based on it:

```php
<?php
const TAU = PI * 2;

define("EARTH_IS_ROUND", !EARTH_IS_FLAT);
define("MORE_UNKNOWN", UNKNOWN);
define("APP_ENV_UPPERCASE", strtoupper(APP_ENV)); // string manipulation is ok too
define("MAX_SESSION_TIME_IN_MINUTES", MAX_SESSION_TIME / 60);
const APP_FUTURE_LANGUAGES = [-1 => "es"] + APP_LANGUAGES; // array manipulations
define("APP_BETTER_FUTURE_LANGUAGES", array_merge(["fr"],BETTER_APP_LANGUAGES));
?>
```

Defining constant using another constant

> **Note:** Remember that you can not use a constant for a function call like this:
>
> ```php
> <?php
> const TIME = time(); // fails with a fatal error!
> ?>
> ```

Reserved Names

Some constant names are reserved by PHP and cannot be redefined. Try to print the values of constants defined below. The custom values assigned to these keywords won't be printed.

```php
<?php
define("true", false); // internal constant
define("false", true); // internal constant
define("CURLOPT_AUTOREFERER", "something"); // will fail if curl extension is loaded
// echo true;
// echo false;
// echo CURLOPT_AUTOREFERER;
?>
```

Reserved constants

Checking if constant is defined

To check if constant is defined, use the `defined` function.

> **Note:** The `defined` function doesn't care about the constant's value; it only cares if the constant exists or not. Even if the value of the constant is null or

> false the function will still return true.

The following code shows how you can use the `defined` function to check if the constants `GOOD` and `AWESOME` exist or not. Ignore the keywords `if` and `else` for now. You will learn about them later. For now, all you need to know is that `if` and `else` are used to decide if a particular part of the code needs to be executed.

```
<?php
define("GOOD", false);
if (defined("GOOD")) {
  echo "GOOD is defined.\n" ; // prints "GOOD is defined"

  if (GOOD)
    echo "GOOD is true." ; // does not print anything, since GOOD is false
  else
    echo "GOOD is false.";
}

if (!defined("AWESOME")) {
  define("AWESOME", true); // awesome was not defined. Now we have defined it
}
?>
```

Getting all defined constants

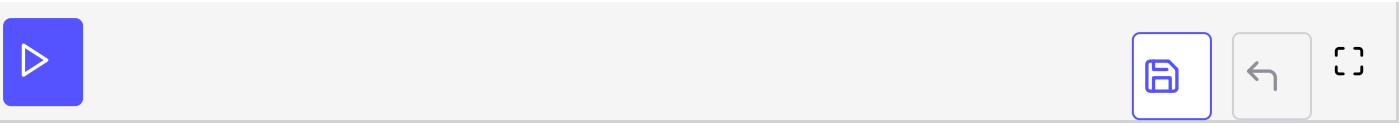
To get all defined constants, including those created by PHP, use the PHP function `get_defined_constants`. The following code snippet shows how this is done:

```
<?php
$constants = get_defined_constants();
var_dump($constants); // prints a very large list
?>
```

To get only those constants that were defined by your app call the function at the beginning and at the end of your script (normally after the bootstrap process). Take a look at the following code snippet:

```
<?php
$constants = get_defined_constants();
define("HELLO", "hello");
define("WORLD", "world");
$new_constants = get_defined_constants();
$myconstants = array_diff_assoc($new_constants, $constants); //compares array keys and values, and
```

```
var_export($myconstants); //return the structured value of $myconstants
?>
```



In line 5 of the above code, we used PHP's `array_diff_assoc` function. This function takes two parameters, compares the keys and values of both and returns the difference. In this case, it returned all the newly defined constants.

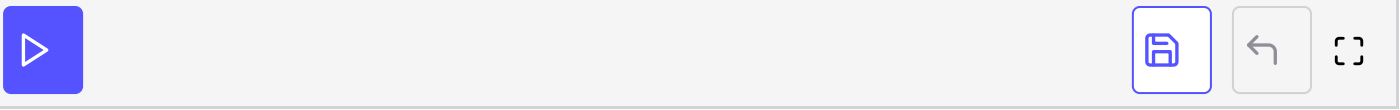
Using Constants

To use the constant simply use its name, like so:

```
<?php
//definition
define("EARTH_IS_FLAT", false); // boolean
define("APP_ENV_UPPERCASE", "DEV"); // string

//usage
if (EARTH_IS_FLAT)
  echo "Earth is flat.\n";
else
  echo "Earth is round.\n";

echo APP_ENV_UPPERCASE;
?>
```

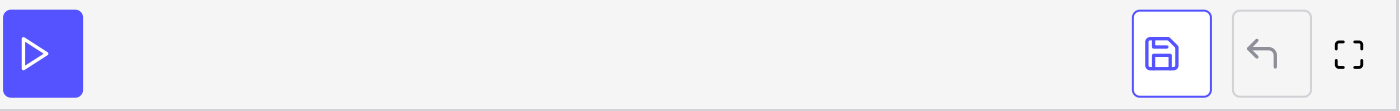


If you don't know the name of the constant in advance, use the constant function:

```
<?php
//definition
define("EARTH_IS_FLAT", false); // boolean
define("APP_ENV_UPPERCASE", "DEV"); // string

$const1 = "EARTH_IS_FLAT";
$const2 = "APP_ENV_UPPERCASE";
if (constant($const1))
  echo "Earth is flat.\n";
else
  echo "Earth is round.\n";

echo constant($const2);
?>
```



Let's practice our concepts by solving a challenge in the next lesson.