

Compare define() vs const in PHP

As we know both **define()** and **const** are used to declare a constant in PHP script.

Syntax



```
<?php
const VAR = 'F00'
define('VAR', 'F00');
?>
```

Let's discuss the difference between these two.

- The basic difference between these two is that **const** defines constants at compile time, whereas **define()** defines them at run time.
- We can't use the **const** keyword to declare constant in conditional blocks, while with **define()** we can achieve that.
- ```
<?php
if(){
const VAR = 'F00'; // invalid
}
if(){
define('VAR', 'F00'); //valid
}
?>
```
- **const** accepts a static scalar(number, string or other constants like true, false, null, \_\_FILE\_\_), whereas **define()** takes any expression.
- **consts** are always case sensitive, whereas **define()** allows you to define case insensitive constants by passing true as the third argument.
- **const** can also be utilized within a class or interface to declare a class constant or interface constant, while **define()** can't be utilized for this reason
- ```
<?php
class abc{
const VAR = 2; // valid
echo VAR; // valid
}
// but
class xyz{
define('XUV', 2); // invalid
echo XUV; // invalid
```

```
}  
?>
```

- The above example shows that we can declare constant inside the class with the **const** keyword but **define()** can't be used for declaring constant inside a class.