

# Class Constants ¶

It is possible to define [constants](#) on a per-class basis remaining the same and unchangeable. The default visibility of class constants is `public`.

## Note:

Class constants can be redefined by a child class.

It's also possible for interfaces to have constants. Look at the [interface documentation](#) for examples.

It's possible to reference the class using a variable. The variable's value can not be a keyword (e.g. `self`, `parent` and `static`).

Note that class constants are allocated once per class, and not for each class instance.

## Example #1 Defining and using a constant

```
<?php
class MyClass
{
    const CONSTANT = 'constant value';

    function showConstant() {
        echo self::CONSTANT . "\n";
    }
}

echo MyClass::CONSTANT . "\n";

$classname = "MyClass";
echo $classname::CONSTANT . "\n";

$class = new MyClass();
$class->showConstant();

echo $class::CONSTANT . "\n";
?>
```

The special `::class` constant allows for fully qualified class name resolution at compile time, this is useful for namespaced classes:

## Example #2 Namespaced `::class` example

```
<?php
namespace foo {
    class bar {

        echo bar::class; // foo\bar
    }
}
?>
```

## Example #3 Class constant expression example

```
<?php
const ONE = 1;
class foo {
    const TWO = ONE * 2;
    const THREE = ONE + self::TWO;
    const SENTENCE = 'The value of THREE is ' . self::THREE;
}
?>
```

**Example #4 Class constant visibility modifiers, as of PHP 7.1.0**

```
<?php
class Foo {
    public const BAR = 'bar';
    private const BAZ = 'baz';
}
echo Foo::BAR, PHP_EOL;
echo Foo::BAZ, PHP_EOL;
?>
```

Output of the above example in PHP 7.1:

bar

Fatal error: Uncaught Error: Cannot access private const Foo::BAZ in ...

**Note:**

As of PHP 7.1.0 visibility modifiers are allowed for class constants.