

Static and Clone

Sometimes, it is useful if we can access methods and properties in the context of a `class` rather than an object. To do this, you can use `static` keyword.

To add a static method to a class, you use the `static` keyword as follows:

```
1 public static function static_method(){  
2     // method implementation  
3 }
```

You can put the `static` keyword before or after the method's visibility. However, by convention, the visibility is declared first.

To add a static property to a class, you also use the `static` keyword as the following syntax:

```
1 private static $static_property;
```

The static methods and static properties are not linked to any particular object of the class but the class itself.

To call a static method outside the class, you use the `::` operator as follows:

```
1 MyClass::static_method();
```

To access a public static property outside the class, you also use the `::` operator:

```
1 MyClass::$static_property;
```

However to access static methods and static properties from within an instance of the class, you use `self` instead of `$this` as follows:

```
1 <?php
2 class MyClass{
3     private static $static_property;
4
5     public static function static_method(){
6         //...
7     }
8
9     public function method(){
10         self::$static_property;
11         self::static_method();
12     }
13 }
```

Developers, who are new to PHP object-oriented programming, are sometimes confused between `self` and `$this`. Let's make the concepts clear.

\$this and self

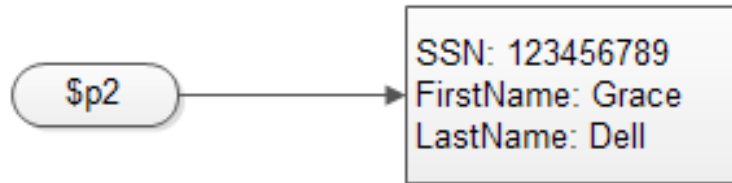
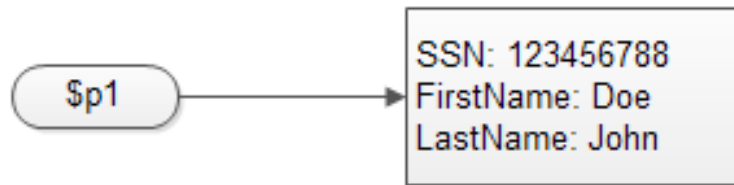
self vs. *\$this*

<code>\$this</code>	<code>self</code>
Represents an instance of the class or object	Represents a class
Always begin with dollar (\$) sign	Never begin with dollar(\$) sign
Is followed by the <code>-></code> operator	Is followed by the <code>::</code> operator
The property name after the <code>-></code> operator does not take dollar (\$) sign, e.g., <code>\$this->property</code> .	The property name after the <code>::</code> operator always take the dollar (\$) sign.

When to use?

- PHP static methods are often used in *utility* classes of PHP frameworks. A utility class is a class that contains only static methods.

Clone



```
$p2 = $p1;
```

