PHP – Static Properties

The "static" keyword in PHP is used to define static properties and static methods in a PHP class. It may be noted that the static keyword is also used to define static variable, and static anonymous functions. Read this chapter to learn about the static properties in a PHP class.

In a class definition, a variable declared with a static qualifier becomes its static property. The static keyword may appear before or after the access modifier.

```php
static private $var1;
public static $var2;
```

If you want to use type hints, the type must not be before the **static** keyword.

```php
static private string $var1;
public static float $var2;
```

The value of the static property in a class is not accessible by its object (with the -> operator). Doing so will result in a notice stating **Accessing static property myclass::$var1 as non static**. Instead, the static properties are accessed using the Scope Resolution Operator represented by the "::" symbol.

Example

Take a look at the following example −

```
</>                                                        Open Compiler

<?php
   class myclass {
      static string $var1 = "My Class";
      function __construct() {
         echo "New object declared" . PHP_EOL;
      }
   }
   $obj = new myclass;
   echo "accessing static property with scope resolution operator: " . myclass::$v
```

```php
   echo "accessing static property with -> operator: ". $obj->var1 . PHP_EOL;
?>
```

It will produce the following **output** −

```
New object declared
accessing static property with scope resolution operator: My Class
PHP Notice:  Accessing static property myclass::$var1 as non static in hello.php on line 1
```

The "self" Keyword

To access the static property from inside a method, refer to the current class with the self keyword. In the following example, the class has an integer static property, which is incremented every time a new object is declared.

Open Compiler

```php
<?php
   class myclass {

      /* Member variables */
      static int $var1 = 0;
      function __construct(){
         self::$var1++;
         echo "object number ". self::$var1 . PHP_EOL;
      }
   }
   for ($i=1; $i<=3; $i++) {
      $obj = new myclass;
   }
?>
```

It will produce the following **output** −

```
object number 1
object number 2
object number 3
```

The "parent" Keyword

The static property of a base class can be used inside a function of the inherited class by referring to the base by parent keyword. You need to use the "parent::static_property" syntax.

Example

Take look at the following example −

```php
<?php
   class myclass {

      /* Member variables */
      static int $var1 = 0;
      function __construct() {
         self::$var1++;
         echo "object number ". self::$var1 . PHP_EOL;
      }
   }

   class newclass extends myclass{
      function getstatic() {
         echo "Static property in parent class: " . parent::$var1 . PHP_EOL;
      }
   }
   $obj = new newclass;
   $obj->getstatic();
?>
```

It will produce the following **output** −

```
object number 1
Static property in parent class: 1
```