

PHP – Static Methods

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. This chapter discusses static methods in a PHP class.

In a class definition, a function declared with a static qualifier becomes its static method.

```
class myclass {  
    public static function myStaticMethod() {  
        // ...  
    }  
}
```

You don't need to create the instance of the class to call its static method. The static method is called by the class name though the scope resolution operator. The syntax of a static method call is –

```
myclass::myStaticMethod();
```

As the static methods are callable without creating an instance of the class, the pseudo-variable `$this` is not available inside static methods. A static method is allowed to be called by an object, although calling an instance method as a static method raises error.

Example

Take a look at the following example –

[Open Compiler](#)

```
<?php  
class myclass {  
  
    /* Member variables */  
    static int $var1 = 0;  
    public static function mystaticmethod() {  
        echo "This is a static method". PHP_EOL;  
    }  
    public function myinstancemethod() {  
        echo "This is an instance method". PHP_EOL;  
    }  
}
```

```
}

myclass::mystaticmethod();
$obj = new myclass;
$obj->myinstancemethod();
$obj->mystaticmethod();
myclass::myinstancemethod();

?>
```

It will produce the following **output** –

```
This is a static method
This is an instance method
This is a static method
PHP Fatal error:  Uncaught Error: Non-static method
myclass::myinstancemethod() cannot be called statically
```

The "self" Keyword in Static Method

If you need to call a static method from inside an instance method defined in the same class, you have to use self keyword referring to the name of the class, followed by the scope resolution operator (such as self::mystaticmethod)

[Open Compiler](#)

```
<?php
class myclass {

    /* Member variables */
    static int $var1 = 0;
    public static function mystaticmethod() {
        echo "This is a static method". PHP_EOL;
    }
    public function myinstancemethod() {
        echo "This is an instance method". PHP_EOL;
        echo "calling static method from instance method" . PHP_EOL;
        self::mystaticmethod();
    }
}

$obj = new myclass;
```

```
$obj->myinstancemethod();  
?>
```

It will produce the following **output** –

```
This is an instance method  
calling static method from instance method  
This is a static method
```

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Using the "parent" Keyword

In case of inheritance, a static method defined in a base class may be called by an object of derived class, or from inside an instance method of the derived class, by referring it with the "parent" keyword.

Example

Take a look at the following example –

[Open Compiler](#)

```
<?php  
class myclass {  
  
    /* Member variables */  
    static int $var1 = 0;  
    public static function mystaticmethod() {  
        echo "This is a static method". PHP_EOL;  
    }  
    public function myinstancemethod() {  
        echo "This is an instance method". PHP_EOL;  
        echo "calling static method from instance method" . PHP_EOL;  
        self::mystaticmethod();  
    }  
}  
  
class mynewclass extends myclass {
```

```
public function myfunction() {  
    echo "This an instance method of the derived class" . PHP_EOL;  
    echo "Calling static method of the parent class" . PHP_EOL;  
    parent::mystaticmethod();  
}  
}  
$obj = new mynewclass;  
mynewclass::mystaticmethod();  
$obj->myfunction();  
?>
```

It will produce the following **output** –

```
This is a static method  
This an instance method of the derived class  
Calling static method of the parent class  
This is a static method
```

Static Method Inside Another Class

It is entirely possible to call the static method from one class in another. You have to qualify its name with its class name followed by the scope resolution operator.

Example

Take a look at the following example –

</>

Open Compiler

```
<?php  
class myclass {  
  
    /* Member variables */  
    static int $var1 = 0;  
    public static function mystaticmethod() {  
        echo "This is a static method". PHP_EOL;  
    }  
}  
  
#this is not a derived class  
class mynewclass {
```

```
public function myfunction() {  
    echo "This an instance method" . PHP_EOL;  
    echo "Calling static method of the another class" . PHP_EOL;  
    myclass::mystaticmethod();  
}  
}  
$obj = new mynewclass;  
$obj->myfunction();  
?>
```

It will produce the following **output** –

```
This an instance method  
Calling static method of another class  
This is a static method
```

Since **\$this** pseudo-variable is not available for a static method, the instance variables of an object cannot be accessed inside a static method. It can process only the static properties of the class.

Example

Take a look at the following example –

</>

Open Compiler

```
<?php  
class myclass {  
  
    /* Member variables */  
    static int $var1 = 0;  
    function __construct() {  
        self::$var1++;  
        echo "object number ". self::$var1 . PHP_EOL;  
    }  
    public static function mystaticmethod() {  
        echo "Number of objects available: " . self::$var1 . PHP_EOL;  
    }  
}  
  
for ($i=1; $i<=3; $i++) {
```

```
$obj = new myclass;  
}  
myclass::mystaticmethod();  
?>
```

It will produce the following **output** –

```
object number 1  
object number 2  
object number 3  
Number of objects available: 3
```