



PHP OOP - Classes and Objects

[< Previous](#)[Next >](#)

A class is a template for objects, and an object is an instance of class.

OOP Case

Let's assume we have a class named Fruit. A Fruit can have properties like name, color, weight, etc. We can define variables like \$name, \$color, and \$weight to hold the values of these properties.

When the individual objects (apple, banana, etc.) are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

Define a Class

A class is defined by using the `class` keyword, followed by the name of the class and a pair of curly braces (`{}`). All its properties and methods go inside the braces:

Syntax

[Get your own PHP Server](#)

```
<?php
class Fruit {
    // code goes here...
}
?>
```



Tutorials ▼

Exercises ▼

Services ▼



Sign Up

Log in

 . CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C

```
<?php
class Fruit {
    // Properties
    public $name;
    public $color;

    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
}
?>
```

Note: In a class, variables are called properties and functions are called methods!

Define Objects

Classes are nothing without objects! We can create multiple objects from a class. Each object has all the properties and methods defined in the class, but they will have different property values.

Objects of a class are created using the **new** keyword.

In the example below, \$apple and \$banana are instances of the class Fruit:

Example



Tutorials ▼

Exercises ▼

Services ▼



Sign Up

Log in

 [CSS](#) [JAVASCRIPT](#) [SQL](#) [PYTHON](#) [JAVA](#) [PHP](#) [HOW TO](#) [W3.CSS](#) [C](#)

```
public $name;  
public $color;  
  
// Methods  
function set_name($name) {  
    $this->name = $name;  
}  
function get_name() {  
    return $this->name;  
}  
}  
  
$apple = new Fruit();  
$banana = new Fruit();  
$apple->set_name('Apple');  
$banana->set_name('Banana');  
  
echo $apple->get_name();  
echo "<br>";  
echo $banana->get_name();  
?>
```

[Try it Yourself »](#)

In the example below, we add two more methods to class Fruit, for setting and getting the \$color property:

Example

```
<?php  
class Fruit {  
    // Properties  
    public $name;  
    public $color;  
  
    // Methods
```



Tutorials ▼

Exercises ▼

Services ▼



Sign Up

Log in

 [CSS](#) [JAVASCRIPT](#) [SQL](#) [PYTHON](#) [JAVA](#) [PHP](#) [HOW TO](#) [W3.CSS](#) [C](#)

```
        return $this->name;
    }
    function set_color($color) {
        $this->color = $color;
    }
    function get_color() {
        return $this->color;
    }
}

$apple = new Fruit();
$apple->set_name('Apple');
$apple->set_color('Red');
echo "Name: " . $apple->get_name();
echo "<br>";
echo "Color: " . $apple->get_color();
?>
```

[Try it Yourself »](#)

PHP - The \$this Keyword

The \$this keyword refers to the current object, and is only available inside methods.

Look at the following example:

Example

```
<?php
class Fruit {
    public $name;
}
```



Tutorials ▼

Exercises ▼

Services ▼



Sign Up

Log in

≡ . CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C

So, where can we change the value of the \$name property? There are two ways:

1. Inside the class (by adding a set_name() method and use \$this):

Example

```
<?php
class Fruit {
    public $name;
    function set_name($name) {
        $this->name = $name;
    }
}
$apple = new Fruit();
$apple->set_name("Apple");

echo $apple->name;
?>
```

Try it Yourself »

2. Outside the class (by directly changing the property value):

Example

```
<?php
class Fruit {
    public $name;
}
$apple = new Fruit();
$apple->name = "Apple";

echo $apple->name;
?>
```



[Tutorials ▼](#)[Exercises ▼](#)[Services ▼](#)[Sign Up](#)[Log in](#)[≡](#) [.](#) [CSS](#) [JAVASCRIPT](#) [SQL](#) [PYTHON](#) [JAVA](#) [PHP](#) [HOW TO](#) [W3.CSS](#) [C](#)

PHP - instanceof

You can use the **instanceof** keyword to check if an object belongs to a specific class:

Example

```
<?php
$apple = new Fruit();
var_dump($apple instanceof Fruit);
?>
```

[Try it Yourself »](#)[◀ Previous](#)[Next ▶](#)

Track your progress - it's free!

[Sign Up](#)[Log in](#)

[Tutorials](#) ▼[Exercises](#) ▼[Services](#) ▼[Sign Up](#)[Log in](#)[CSS](#) [JAVASCRIPT](#) [SQL](#) [PYTHON](#) [JAVA](#) [PHP](#) [HOW TO](#) [W3.CSS](#) [C](#)

COLOR PICKER

[PLUS](#)[SPACES](#)[GET CERTIFIED](#)[FOR TEACHERS](#)[FOR BUSINESS](#)[CONTACT US](#)

Top Tutorials

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)
- [How To Tutorial](#)
- [SQL Tutorial](#)

Top References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)
- [SQL Reference](#)
- [Python Reference](#)

[Tutorials ▼](#)[Exercises ▼](#)[Services ▼](#)[Sign Up](#)[Log in](#)[CSS](#)[JAVASCRIPT](#)[SQL](#)[PYTHON](#)[JAVA](#)[PHP](#)[HOW TO](#)[W3.CSS](#)[C](#)[jQuery Tutorial](#)

Top Examples

[HTML Examples](#)
[CSS Examples](#)
[JavaScript Examples](#)
[How To Examples](#)
[SQL Examples](#)
[Python Examples](#)
[W3.CSS Examples](#)
[Bootstrap Examples](#)
[PHP Examples](#)
[Java Examples](#)
[XML Examples](#)
[jQuery Examples](#)

[jQuery Reference](#)

Get Certified

[HTML Certificate](#)
[CSS Certificate](#)
[JavaScript Certificate](#)
[Front End Certificate](#)
[SQL Certificate](#)
[Python Certificate](#)
[PHP Certificate](#)
[jQuery Certificate](#)
[Java Certificate](#)
[C++ Certificate](#)
[C# Certificate](#)
[XML Certificate](#)

[FORUM](#) [ABOUT](#) [ACADEMY](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our [terms of use](#), [cookie and privacy policy](#).

Copyright 1999-2025 by Refsnes Data. All Rights Reserved. [W3Schools is Powered by W3.CSS](#).

