



# Python Classes and Objects

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

## Python Classes/Objects

Python is an object oriented programming language.

Almost everything in Python is an object, with its properties and methods.

A Class is like an object constructor, or a "blueprint" for creating objects.

## Create a Class

To create a class, use the keyword `class` :

### Example

[Get your own Python Server](#)

Create a class named MyClass, with a property named x:

```
class MyClass:
    x = 5
```

[Try it Yourself »](#)

## Create Object

Now we can use the class named MyClass to create objects:

### Example

Create an object named p1, and print the value of x:

```
p1 = MyClass()
print(p1.x)
```

[Try it Yourself »](#)

## The \_\_init\_\_() Function

The examples above are classes and objects in their simplest form, and are not really useful in real life applications.

To understand the meaning of classes we have to understand the built-in `__init__()` function.

All classes have a function called `__init__()`, which is always executed when the class is being initiated.

Use the `__init__()` function to assign values to object properties, or other operations that are necessary to do when the object is being created:





```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

[Try it Yourself »](#)

**Note:** The `__init__()` function is called automatically every time the class is being used to create a new object.

## The `__str__()` Function

The `__str__()` function controls what should be returned when the class object is represented as a string.

If the `__str__()` function is not set, the string representation of the object is returned:

### Example

The string representation of an object WITHOUT the `__str__()` function:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

p1 = Person("John", 36)

print(p1)
```

[Try it Yourself »](#)

### Example

The string representation of an object WITH the `__str__()` function:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"{self.name}({self.age})"

p1 = Person("John", 36)

print(p1)
```

[Try it Yourself »](#)



Let us create a method in the Person class:

## Example

Insert a function that prints a greeting, and execute it on the p1 object:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def myfunc(self):
        print("Hello my name is " + self.name)

p1 = Person("John", 36)
p1.myfunc()
```

Try it Yourself »

**Note:** The `self` parameter is a reference to the current instance of the class, and is used to access variables that belong to the class.

## The self Parameter

The `self` parameter is a reference to the current instance of the class, and is used to access variables that belongs to the class.

It does not have to be named `self`, you can call it whatever you like, but it has to be the first parameter of any function in the class:

## Example

Use the words *mysillyobject* and *abc* instead of *self*:

```
class Person:
    def __init__(mysillyobject, name, age):
        mysillyobject.name = name
        mysillyobject.age = age

    def myfunc(abc):
        print("Hello my name is " + abc.name)

p1 = Person("John", 36)
p1.myfunc()
```

Try it Yourself »

## Modify Object Properties

You can modify properties on objects like this:

## Example

Set the age of p1 to 40:

```
p1.age = 40
```





## Delete Object Properties

You can delete properties on objects by using the `del` keyword:

### Example

Delete the age property from the p1 object:

```
del p1.age
```

[Try it Yourself »](#)

## Delete Objects

You can delete objects by using the `del` keyword:

### Example

Delete the p1 object:

```
del p1
```

[Try it Yourself »](#)

## The pass Statement

`class` definitions cannot be empty, but if you for some reason have a `class` definition with no content, put in the `pass` statement to avoid getting an error.

### Example

```
class Person:  
    pass
```

[Try it Yourself »](#)

## Test Yourself With Exercises

### Exercise:

Create a class named MyClass:

```
MyClass:  
    x = 5
```





Tutorials ▾

Exercises ▾

Services ▾



✦ Plus

⌕ Spaces

🛒 Get Certified

Sign Up

Log in



CSS

JAVASCRIPT

SQL

PYTHON

JAVA

PHP

HOW TO

W3.CSS

C

C++

C#

BOOTSTRAP

REACT

MYSQL

[Start the Exercise](#)[< Previous](#)[Next >](#)

W3schools Pathfinder  
Track your progress - it's free!

Sign Up

Log in

Summer Sale

50%  
OFF

On all courses &amp; certificates

Check it out!



COLOR PICKER



SPACES

UPGRADE

AD-FREE

NEWSLETTER

GET CERTIFIED

CONTACT US

## Top Tutorials

HTML Tutorial  
CSS Tutorial  
JavaScript Tutorial  
How To Tutorial  
SQL Tutorial  
Python Tutorial  
W3.CSS Tutorial  
Bootstrap Tutorial  
PHP Tutorial  
Java Tutorial  
C++ Tutorial  
jQuery Tutorial

## Get Certified

HTML Certificate  
CSS Certificate  
JavaScript Certificate  
Front End Certificate  
SQL Certificate  
Python Certificate  
PHP Certificate




## Top References


HTML Reference  
CSS Reference  
JavaScript Reference  
SQL Reference  
Python Reference  
W3.CSS Reference  
Bootstrap Reference  
PHP Reference  
HTML Colors  
Java Reference  
Angular Reference  
jQuery Reference

## 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



Tutorials▼Exercises▼Services▼

Plus</> SpacesGet CertifiedSign UpLog in

CSSJAVASCRIPTSQLPYTHONJAVAPHPHOW TOW3.CSSCC++C#BOOTSTRAPREACTMYSQL



FORUMABOUTCLASSROOM

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 1999-2024 by Refsnes Data. All Rights Reserved. W3Schools is Powered by W3.CSS.

