

Classes and Objects in Python

The basic idea behind an
object-oriented programming (OOP) is
to combine
both data and associated procedures (known as
methods)
into a single *unit*
which operate on the data.
Such a *unit* is called an object.

Classes and Objects in Python

- Classes:
 - Classes provide a means of **bundling data and functionality together**.
 - Class creates a **user-defined data structure**, which holds its own data members and member functions, which can be accessed and used by creating **an instance of that class**.
 - A class is like a blueprint for an object.
 - Classes are **created by keyword class**.



instance

/ˈɪnst(ə)ns/

noun

plural noun: **instances**

an example or single occurrence of something.
"a serious instance of corruption"

Similar:

example

occasion

occurrence

case

representative case



Classes and Objects in Python

Class Definition Syntax:

```
class ClassName:  
    # Statement-1  
    .  
    .  
    .  
    # Statement-N
```

Classes and Objects in Python

- Object:
 - An Object is **an instance of a Class**.
 - A class is like a blueprint while an instance is a copy of the class with *actual values*.
 - An object consists of :
 - **State** : It is represented by attributes of an object. It also reflects the **properties of an object**.
 - **Behavior** : It is represented by **methods of an object**. It also reflects the response of an object with other objects.
 - **Identity** : It gives a **unique name to an object** and enables one object to interact with other objects.

Classes and Objects in Python

- **Declaring Objects (Also called **instantiating a class**)**
 - When an object of a class is created, the class is said to be instantiated.
 - All the instances share the attributes and the behavior of the class.
 - But the values of those attributes, i.e. the state are unique for each object.
 - A **single class may have any number of instances.**

Classes and Objects in Python

- Syntax for declaring an object:

object_name = Classname()

Classes and Objects in Python

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

    def details(self):
        print("Employee Name:", self.name)
        print("Employee Age:", self.age)
```

```
emp1 = Employee("Sam", 36)
emp1.details()
emp2 = Employee("Edie", 43)
emp2.details()
|
```

First object called
emp1 is created

Second object called
emp2 is created

Classes and Objects in Python

- **The self:**
 - Class methods **must** have an extra first parameter in method definition. We do not give a value for this parameter when we call the method, Python provides it.
 - If we have a method which takes no arguments, then we still have to have one argument.
 - The **self parameter is a reference to the current instance of the class**, and is used to access variables that belongs to the class.
- When we call a method of this object as `myobject.method(arg1, arg2)`, this is automatically converted by Python into `MyClass.method(myobject, arg1, arg2)`

Classes and Objects in Python

- **`__init__` method**

- Class functions that begins with double underscore (`__`) are called special functions as they have special meaning.
- This special function “`__init__`” gets called **whenever a new object of that class is instantiated.**
- This **type of function is also called constructors** in Object Oriented Programming (OOP). **We normally use it to initialize all the variables.**

Classes and Objects in Python

Create a class in python called Book which will take following attributes:

ID, Title, Author, Number of pages.

Include a method in the class which will output the ID of the book, title of the book, author of the book and number of pages contained in the book.

Create two objects of this class and your output should look like this:

```
The unique id of the book is: 1
The title of the book is: Computer Network
The author of the book is: William Stallings
The number of pages in the book is: 789
The unique id of the book is: 2
The title of the book is: Indian Music
The author of the book is: Some Singer
The number of pages in the book is: 850
```

Classes and Objects in Python

- Class and instance variables
 - Instance variables are variables used for data that is **unique to a particular instance**.
 - Class Variables are variables that are **shared by all instances** of a class.

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.

```
class Person:  
    pass
```