Constructor in Python

* A constructor is a special type of method (function) which is used to initialize the instance members of the class.
* In C++ or Java, the constructor has the same name as its class, but it treats constructor differently in Python. It is used to create an object.
* Constructor definition is executed when we create the object of this class. Constructors also verify that there are enough resources for the object to perform any start-up task.
* In Python, the method the **\_\_init\_\_()** simulates the constructor of the class. This method is called when the class is instantiated. It accepts the **self**-keyword as a first argument which allows accessing the attributes or method of the class.
* We can pass any number of arguments at the time of creating the class object, depending upon the **\_\_init\_\_()** definition. It is mostly used to initialize the class attributes. Every class must have a constructor, even if it simply relies on the default constructor.
* For Example:

class emp():

def \_\_init\_\_(self,name,age):

self.name = name

self.age = age

def intro(self):

return “Hello my name is {self.name}, and my age is {age}”

employee1 = emp(“Charlie”, 23)

print(employee1.intro())

#OUTPUT: “Hello my name is Charlie, and my age is 23”

**Features of Construct:**

It is use to initialize the instance variable.

It returns nothing.

It only initialized by \_\_init\_\_.

No logical coding done inside it