Instance method

Note: A function defined inside class is called method.

A method defined inside class with first argument as **“self”** is called instance method or object level method. This method is bind with object name and this method cannot invoke without creating object. This method defines behavior of object.

Syntax:

```
class <class-name>:
        def <method-name>(self,parameter,parameter,…):
                statement-1
                statement-2
```

What is “self”?

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.

In Python, self serves as a reference to the current class instance, enabling access and modification of its attributes and methods.

Example:

```
class Car:  # Model or Blueprint
    def start(self): # instance method or object level method
        print("Car Start....")
    def stop(self): # instance method or object level method
        print("Car Stop...")
```

```
audi=Car() # object of car
benz=Car() # object of car

audi.start()
audi.stop()

benz.start()
benz.stop()
```

Output

```
Car Start....
Car Stop...
Car Start....
Car Stop...
```

Note: in the place of “self” other parameter name can be defined.

```
class Person: # class
    def walk(self): # instance method or object level method
        print("Walk...")
    def run(self): # instance method or object level method
        print("Run...")


person1=Person()
person2=Person()

person1.walk()
person1.run()

person2.walk()
person2.run()
```

Output

```
Walk...
Run...
Walk...
Run...
```

Creating or accessing instance variables using instance method

Inside a class any variable bind with **“self”** is called instance variable. Instance variables are accessible or created inside only instance methods.

Syntax:

```
class <class-name>/<class-type-name>:
        def <instance-method>(self,param,param,param):
                variable-name=value # L.V
                self.<variable-name>=value # I.V or OLV
                self.<variable-name>=value # I.V or OLV
                statement-1
                statement-2
        def <instance-method>(self,param,param,param):
                variable-name=value # L.V
                self.<variable-name>=value # IV or OLV
                self.<variable-name>=value # IV or OLV
                statement-1
                statement-2
```

Example:

```
class Employee:
    def create_properties(self):
        self.empno=101
        self.ename="naresh"
        self.salary=5000
    def print_properties(self):
        print(self.empno)
        print(self.ename)
        print(self.salary)

emp1=Employee()
emp1.create_properties()
emp2=Employee()
emp2.create_properties()
emp1.print_properties()
emp2.print_properties()
emp3=Employee()
```

Output

```
101
naresh
5000
101
naresh
5000
```

Example:

```
class Student:
    def create_attributes(self):
        self.rollno=None
        self.name=None


stud1=Student()
stud1.create_attributes()
print(stud1.rollno,stud1.name)
stud1.rollno=101
stud1.name="naresh"
```

```
print(stud1.rollno,stud1.name)
stud2=Student()
stud2.create_attributes()
print(stud2.rollno,stud2.name)
```

Output

```
None None
101 naresh
None None
```