Instance variables or Instance data members or object level variables

Instance variables define the properties of object.
Instance variables define the state of object.
Instance variables memory is allocated or created within object context or within object. These variables are bind or accessed with object name.

How to create instance variables?

1. Within class
2. Outside the class

Outside the class

Outside the class instance variables are created or accessed with object name.

Syntax1: object-name.instance-variable-name=<value>
Syntax2: object-name.instance-variable-name

Syntax1 is for creating instance variable
Syntax2 is for accessing instance variable

Example:	**Output**
class Student: pass stud1=Student() stud1.rollno=1 stud1.name="naresh" stud2=Student() stud2.rollno=2 stud2.name="suresh" print(stud1.rollno,stud1.name) print(stud2.rollno,stud2.name)	1 naresh 2 suresh

Outside the class instance variables can be created using a predefined property of every object called __dict__

Syntax: object-name.__dict__={‘variable-name’:value, ‘variable-name’:value,….}

Example:

```
class Employee:
   pass


emp1=Employee()
emp1.__dict__={'empno':101,'ename':'naresh','sal':50000}
print(emp1.empno,emp1.ename,emp1.sal)

emp2=Employee()
emp2.__dict__={'empno':102,'ename':'suresh','sal':65000}
print(emp2.empno,emp2.ename,emp2.sal)
```

Output

```
101 naresh 50000
102 suresh 65000
```

Creating instance variables within class

Instance variables or object level variables within class are bind with “self”.

What is “self”?

“self” is parameter name or local variable
This parameter refers to object (current object)

What is instance method?

A method defined inside class with first parameter “self” is called instance method. This method is bind with object name. This method cannot invoke without creating object.

This method is used for operating instance variables.

Syntax:

```
class class-name:
    def instance-method-name(self,param,param,param,…):
        statement-1
        statement-2
```

Constructor

Constructor is a special instance method or magic method.

What is magic method?
A method which is prefix and suffix with double underscore are called magic methods. These methods get executed automatically when performed some operations.
Magic methods are predefined methods; these are provided by python and implemented by programmer.

Constructor is a magic method used for creating instance variables.
This method gets executed automatically on creation of object.
In application development constructor for initialization of object.

Note:
Dunder methods, also known as magic methods, are special methods in Python that begin and end with double underscores

What is name of the constructor?
Name of the constructor is __init__

Constructor gets executed only once, on creation of object.
Constructor is defined,

1. Without parameters
2. With parameters

Syntax:

```
class class-name:
    def __init__(self,param,param,param,..):
            self.instance-variable=value
            self.instance-variable=value
```

Constructor without parameters

Constructor without parameters does not receives values.

Example:	**Output**

class Employee:
  def __init__(self):
  self.empno=0
  self.ename=None
 	
emp1=Employee()
emp2=Employee()
comp1=complex()
print(emp1.empno,emp1.ename)
print(emp2.empno,emp2.ename)
print(comp1.real,comp1.imag)
	0 None
	0 None
	0.0 0.0

Constructor with parameters

Constructor with parameters receives values at the time creating object.

Example:	**Output**

class Employee:
  def __init__(self,x,y):
  self.empno=x
  self.ename=y
 	
emp1=Employee(101,"naresh")
emp2=Employee(102,"suresh")
comp1=complex(1.2,3.5)
print(emp1.empno,emp1.ename)
print(emp2.empno,emp2.ename)
print(comp1.real,comp1.imag)
	101 naresh
	102 suresh
	1.2 3.5

Example	**Output**

class Account:
  def __init__(self,a,cn,b=0):
  self.accno=a
  self.cname=cn
  self.balance=b
 	
acc1=Account(101,"naresh",50000)
acc2=Account(102,"suresh")
 	
print(acc1.accno,acc1.cname,acc1.balance)
print(acc2.accno,acc2.cname,acc2.balance)
	101 naresh 50000
	102 suresh 0

Within class we can define only one constructor.