```
class myclass:
  x = 100 # class variable x definition
  def f1(): # class function or method f1 definition
    print('class method or function : f1( )')
    print('class variable x :',myclass.x)

  def f2(self): # object function or method f2 definition
    print('object method or function : f2( )')
    print('object variable x',self.x)
    print('current object memory id :',id(self))
```

```
myclass.f1()
```

```
class method or function : f1( )
class variable x : 100
```

```
m1 = myclass() # empty object created
```

```
m1.x = 'python' # instance or object variable x definition
```

```
m2 = myclass() # empty object created
```

```
m2.x = 'javascript' # instance or object variable x definition
```

```
print('m1 object memory id :',id(m1))
```

```
m1 object memory id : 139643465939216
```

```
print('m2 object memory id :',id(m2))
```

```
m2 object memory id : 139643465937488
```

```
m1.f2()
```

```
object method or function : f2( )
object variable x python
current object memory id : 139643465939216
```

```
m2.f2()
```

```
object method or function : f2( )
object variable x javascript
current object memory id : 139643465937488
```

object or instance method or function should have self as a first parameter

self contains current object reference

```
a1 = int(10)
print(a1)
print(type(a1))
print(id(a1))
```

```
10
<class 'int'>
139644518005328
```

```
a2 = float(12.23)
print(a2)
print(type(a2))
print(id(a2))
```

```
12.23
<class 'float'>
139643465221808
```

```
a3 = float()
print(a3)
print(type(a3))
print(id(a3))
```

```
0.0
<class 'float'>
139643465432048
```

constructor is a special function

constructor function name : _*init* _(self)

constructor is an instance or object method

Please do not call constructor method explicitly

during object creation python will call constructor method implictly

```
############################################################
class myclass:
  x = 100 # class variable x definition
  def f1(): # class function or method f1 definition
    print('class method or function : f1( )')
    print('class variable x :',myclass.x)

  def __init__(self): # constructor function
    print('object memory id :',id(self))

  def f2(self): # object function or method f2 definition
    print('object method or function : f2( )')
    print('object variable x',self.x)
    print('current object memory id :',id(self))

m1 = myclass()
```

```
object memory id : 139643464329248
```

```
print(id(m1))
```

```
139643464329248
```

```
m2 = myclass()
```

```
object memory id : 139643966066256
```

```
print(id(m2))
```

```
139643966066256
```

```
m3 = myclass()
```

```
object memory id : 139644023065808
```

```
print(id(m3))
```

```
139644023065808
```

constructor is used to initialize object or instance

```
class myclass:
  def __init__(self): # constructor definition
    self.x = 'python'

  def showdata(self):
```

```
    print(self.x)

m1 = myclass()

m2 = myclass()

m1.showdata()
```

```
    python
```

```
m2.showdata()
```

```
    python
```

constructor with parameters

```
class myclass:
  def __init__(self,data): # constructor definition
    self.name = data

  def showdata(self):
    print(self.name)

m1 = myclass('srikanth')

m2 = myclass('jagadeesh')

m1.showdata()
```

```
    srikanth
```

```
m2.showdata()
```

```
    jagadeesh
```

✓ 0s completed at 3:02 PM