

# Inheritance in Python:

Inheritance is one of the powerful features of Python since it is an Object Oriented Programming Language.

## What is Inheritance in Python?

Inheritance in python is a way through which we can use the property of one class into another class.

This ensures the reusability of code and hence reduces the complexity of the program.

The class which inherits other class is called Base Class and the class which has been inherited from other class is known as derived class.

## Python Inheritance Block Diagram:

Base Class



Derive Class

## Python Inheritance Syntax:

```
class Base_Class:
```

```
    #statements
```

```
    pass
```

```
class Derived_Class(Base_Class):
```

```
    #statements
```

```
    Pass
```

**In Python, we have several types of Inheritance concepts, they are like,**

1. Single Level Inheritance
2. MultiLevel Inheritance
3. Multiple Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

## 1. Single Level Inheritance:

It is important to note that we should only inherit from another class if there is a 'IS A' relationship.

For example, Dog is an Animal.

Let's see this in action in with python program.

### **Example 1:**

To inherit or extend a class we need to declare the derived class name with the base class name inside the parenthesis.

#### **#Base Class**

```
class Animal:
    def __init__(self):
        print("I am an Animal")
    def eat(self):
        print("I can eat")
```

#### **#Derived Class**

```
class Dog(Animal):
    def __init__(self):
        super().__init__() #To call base class __init__ method
        print("I am a Dog")
    def sound(self):
        print("I can Bark")
```

#Creating only Derived class object

```
dog = Dog()
```

#calling methods using the object

```
dog.eat()
```

```
dog.sound()
```

#### **Output:**

I am an Animal

I am a Dog

I can eat

I can Bark

#### **Explanation:**

We have first created class 'Animal' then inherited it into 'Dog' class.

Using super() method we have called Animal class \_\_init\_\_() method from Dog class

\_\_init\_\_() method.

super() method returns the reference of Base class, hence allowing us to use base class into any derived class.

After which we have created Dog class object and called all the methods both of Animal and Dog class.

As you can see, that without any error we are able to call Animal class method's eat() using Dog class object, which proves that we have successfully inherited Animal class into Dog class i.e implemented inheritance.

----->> Let's see one more example of Single Level Inheritance.

### **Example2:**

```
class Demo:
```

```
    def show1(self):
```

```
        print('demo class show')
```

```
class Test(Demo):
```

```
    def show2(self):
```

```
        print('Test class show')
```

```
t1 = Test()
```

```
t1.show1()
```

```
t1.show2()
```

### **Output:**

```
demo class show
```

```
Test class show
```

### **Example 3:**

```
class Demo:
```

```
    def __init__(self):
```

```
        self.x = 10
```

```
        self.y = 20
```

```
def show(self):
    print('X value is :', self.x)
    print('Y value is :', self.y)
```

```
class Test(Demo):
    def sum(self):
        c = self.x + self.y
        print('Sum is :',c)
t1 = Test()
t1.show()
t1.sum()
print()
d = Demo()
d.show()
#d.sum() # AttributeError: 'Demo' object has no attribute 'sum'
```

### **Output:**

```
X value is : 10
Y value is : 20
Sum is : 30
X value is : 10
Y value is : 20
```

### **Example4:**

```
class A():
    def m1(self):
        print('iam from m1()')
    def m2(self):
        print('iam from m2()')

class B(A):
    def m3(self):
        print('iam from m3()')
    def m4(self):
        print('iam from m4()')
```

```
b = B() # child class object
b.m1()
b.m2()
b.m3()
b.m4()
print()
a = A()
a.m1()
a.m2()
#a.m3()
```

### **Output:**

```
iam from m1()
iam from m2()
iam from m3()
iam from m4()
```

```
iam from m1()
iam from m2()
```

### **Q. How to create Bank application for deposit() , withdraw() and chec\_balance() details.?**

Bank Example:

```
class Account:
    balance = 0 # class/static variable
    def __init__(self, balance):
        self.balance = balance

    def check_balance(self):
        print("Current Balance: ",self.balance)

    def deposit(self,amt):
        self.balance += amt
        print(amt," deposit Successfull")
```

```
def withdraw(self, amt):  
    if(amt > self.balance):  
        print("Insufficient Balance")  
        return  
    self.balance -= amt  
    print(amt, " withdraw Successfull")
```

```
class Saving_Account(Account):  
    rate = 6.9  
    def __init__(self, balance):  
        super().__init__(balance)  
  
    def interest(self, months):  
        print("Interest: %.2f" %(self.balance * (self.rate/100) * months))
```

```
account = Saving_Account(800)  
account.deposit(200)  
account.withdraw(300)  
account.check_balance()  
account.interest(5)
```

### **Output:**

```
200 deposit Successfull  
300 withdraw Successfull  
Current Balance: 700  
Interest: 241.50
```

----->> Hope now you will feel a little bit confident in writing inheritance related programs in python.

### **Multilevel Inheritance Concept:**

Multilevel inheritance is also possible in Python unlike other programming languages.

We can inherit a derived class from another derived class. This is known as multilevel inheritance. In python multilevel inheritance can be done at any depth. If we have intermediate class then it is called as 'multi-level' inheritance. Intermediate class means that class which act as parent class as well as child class. Multilevel inheritance requires three ranges of classes, i.e., a Base class , an Intermediate class, and sub class.

### **Syntax:**

**class Base:**

**Features of base class**

**pass**

**class Derived1(Base):**

**Features of base class + Derived1 class**

**pass**

**class Derived2(Derived1):**

**Features of base class + Derived1 class + Derived2 class**

**Pass**

### **Example1:**

**class A():**

**def m1(self):**

**print('iam from m1()')**

**class B(A):**

**def m2(self):**

**print('iam from m2()')**

**class C(B):**

**def m3(self):**

**print('iam from m3()')**

**print('using Base class A')**

**a = A()**

**a.m1()**

**#a.m2()**

```
#a.m3()
print()
print('using Intermediate class B')
b = B()
b.m1()
b.m2()
#b.m3()
print()
print('using Derive class C')
c = C()
c.m1()
c.m2()
c.m3()
```

### **Output:**

```
using Base class A
iam from m1()
using Intermediate class B
iam from m1()
iam from m2()
using Derive class C
iam from m1()
iam from m2()
iam from m3()
```

### **Example2: How to set Student and marks details? Display all student details?**

```
class student:
    def getStudent(self):
        self.name = input('Name :')
        self.age = input('age :')
        self.gender = input('gender :')

class test(student):
    def getMarks(self):
        self.stuClass = input('Class')
```

```
print('Enter the marks of respective subjects')
self.literature = int(input('Literature :'))
self.math = int(input('Math :'))
self.biology = int(input('Biology :'))
self.physics = int(input('Physics :'))
```

```
class marks(test):
    def display(self):
        print('\n\nName :',self.name)
        print('Age :',self.age)
        print('Gender :',self.gender)
        print('Study in :',self.stuClass)
        print('Total Marks :',self.literature + self.math + self.biology + self.physics)
```

```
m1 = marks()
m1.getStudent()
m1.getMarks()
m1.display()
```

### **Output:**

```
Name :Avinesh
age :30
gender :male
Class10th
Enter the marks of respective subjects
Literature :50
Math :60
Biology :55
Physics :46
Name : Avinesh
Age : 30
Gender : male
Study in : 10th
Total Marks : 211
```

### Example3:

```
class Animal:
    def eat(self):
        print('Eating..')
class Dog(Animal):
    def bark(self):
        print('Barking..')
class BabyDog(Dog):
    def weep(self):
        print('Weeping')
d = BabyDog()
d.eat()
d.bark()
d.weep()
```

### Output:

```
Eating..
Barking..
Weeping
```