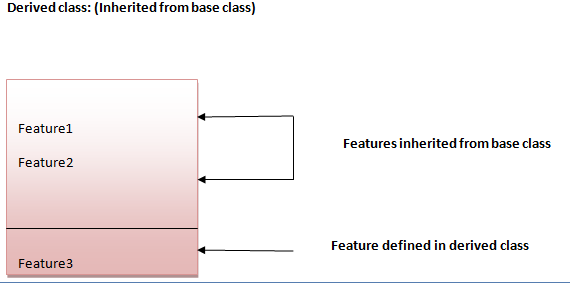
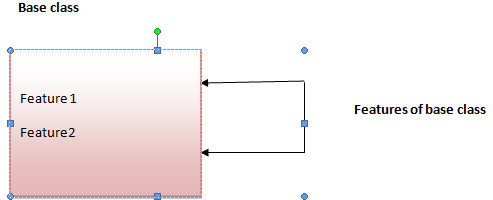
# Inheritance in Python

## What is Inheritance

Inheritance is used to specify that one class will get most or all of its features from its parent class. It is a feature of Object Oriented Programming. It is a very powerful feature which facilitates users to create a new class with a few or more modification to an existing class. The new class is called child class or derived class and the main class from which it inherits the properties is called base class or parent class.

The child class or derived class inherits the features from the parent class, adding new features to it. It facilitates re-usability of code.

**Image representation:**



**Syntax 1:**

1. class DerivedClassName(BaseClassName):
2. <statement-1>
3. .
4. .
5. .
6. <statement-N>

**Syntax 2:**

1. class DerivedClassName(modulename.BaseClassName):
2. <statement-1>
3. .
4. .
5. .
6. <statement-N>

## Parameter explanation:

The name BaseClassName must be defined in a scope containing the derived class definition. You can also use other arbitrary expressions in place of a base class name. This is used when the base class is defined in another module.

## Python Inheritance Example

Let's see a simple python inheritance example where we are using two classes: Animal and Dog. Animal is the parent or base class and Dog is the child class.

Here, we are defining eat() method in Animal class and bark() method in Dog class. In this example, we are creating instance of Dog class and calling eat() and bark() methods by the instance of child class only. Since, parent properties and behaviors are inherited to child object automatically, we can call parent and child class methods by the child instance only.

1. class Animal:
2. def eat(self):
3. print 'Eating...'
4. class Dog(Animal):
5. def bark(self):
6. print 'Barking...'
7. d=Dog()
8. d.eat()
9. d.bark()

**Output:**

1. Eating...
2. Barking...