Inheritance

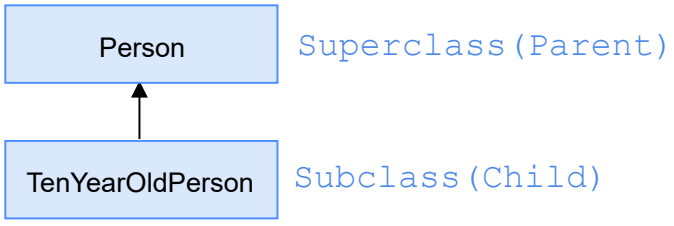
This lesson discusses inheritance, i.e., one class inheriting attributes and methods of another class.

WE'LL COVER THE FOLLOWING

Class Inheritance

Inheritance is an essential part of object-oriented programming. Inheritance is a process in which a **subclass** can inherit the attributes and methods of another class, allowing it to rewrite some of the **super class**'s functionalities. For instance, the `Person` class in the first lesson we could permit us to create a subclass for people at 10 years of age.

1 of 3



2 of 3

The following codes demonstrates how this works in Python:

```
class Person:
      def __init__(self, name, age): # Person's constructor
          self.name = name # Person's attribute
          self.age = age # Person's attribute

      def greet(self): # Person's method
          print("Hello, my name is %s!" % self.name)

class TenYearOldPerson(Person): # TenYearOldPerson inherits from Person

      def __init__(self, name): # TenYearOldPerson's constructor
          Person.__init__(self, name, 10) # accesses Person's constructor

      def greet(self): # rewrites the greet method
          print("I don't talk to strangers!!")

tyo = TenYearOldPerson("Jack") # instance of TenYearOldPerson
tyo.greet() # call greet method of the TenYearOldPerson
```

The indication that the `TenYearOldPerson` class is a subclass of `Person` is given on line 9. Then, we rewrote the constructor of the subclass only to receive the name of the person, but we will eventually call the super class's constructor with the name of the 10-year-old and the age hardcoded as 10. Finally, we reimplemented the `greet` method.

Now that you know the basics of inheritance, let's move on to slightly more advanced concepts of inheritance in the next lesson.