

In [5]:

```
1  #Prog 1
2  #Class Attribute
3  class Dog:
4      species = 'mammal'
5      #Initializer / Instance Attributes
6      def __init__(self,name,age):
7          self.name = name
8          self.age = age
9      #instance method
10     def description(self):
11         return "{} is {} years old".format(self.name,self.age)
12     #instance method
13     def speak(self,sound):
14         return "{} says {}".format(self.name,sound)
15 #Instantiat the Dog object
16 pike = Dog("Pike", 6)
17
18 #Call our instance methods
19 print(pike.description())
20 print(pike.speak("Woof Woof!"))
```

Pike is 6 years old

Pike says Woof Woof!

In [6]:

```
1  #Prog 2
2  #Parent class
3  class Dog:
4      #class attribute
5      species='mammal'
6
7      #Initializer/Instance attributes
8      def __init__(self,name,age):
9          self.name = name
10         self.age = age
11
12         #instance method
13         def description(self):
14             return "{} is {} years old".format(self.name,self.age)
15
16         #instance method
17         def speak(self,sound):
18             return "{} says {}".format(self.name,sound)
19
20     #Child class (inherits from Dog class)
21     class RussellTerrier(Dog):
22         def run(self,speed):
23             return "{} runs {}".format(self.name,speed)
24     #Child class(inherits from Dog class)
25     class Bulldog(Dog):
26         def run(self,speed):
27             return "{} runs {}".format(self.name,speed)
28
29     #Child classes inherit attributes and behaviors from the parent class
30     thunder = Bulldog("Thunder",9)
31     print(thunder.description())
32
33     #Child classes have specific attributes and behaviors as well
34     print(thunder.run("Slowly"))
35
36     spinter = Bulldog("Spinter",12)
37     print(spinter.description())
38     print(spinter.run("Fast"))
39
40     roger = RussellTerrier("Roger",5)
41     print(roger.description())
42     print(roger.run("quickly"))
```

Thunder is 9 years old
Thunder runs Slowly
Spinter is 12 years old
Spinter runs Fast
Roger is 5 years old
Roger runs quickly

```

In [4]: 1 # Prog 3(with error)
        2 # Parent class
        3 class Dog:
        4     # Class attribute
        5     species = 'mammal'
        6     # Initializer / Instance attributes
        7     def __init__(self, name, age):
        8         self.name = name
        9         self.age = age
        10    # instance method
        11    def description(self):
        12        return "{} is {} years old".format(self.name, self.age)
        13    # instance method
        14    def speak(self, sound):
        15        return "{} says {}".format(self.name, sound)
        16    # Child class (inherits from Dog() class)
        17    class RussellTerrier(Dog):
        18        def run(self, speed):
        19            return "{} runs {}".format(self.name, speed)
        20    # Child class (inherits from Dog() class)
        21    class Bulldog(Dog):
        22        def run(self, speed):
        23            return "{} runs {}".format(self.name, speed)
        24    # Child classes inherit attributes and behaviors from the parent class
        25    thunder = Bulldog("Thunder", 9)
        26    print(thunder.description())
        27    # Child classes have specific attributes and behaviors as well
        28    print(thunder.run("slowly"))
        29    # Is thunder an instance of Dog()?
        30    print(isinstance(thunder, Dog))
        31    # Is thunder_kid an instance of Dog()?
        32    thunder_kid = Dog("ThunderKid", 2)
        33    print(isinstance(thunder, Dog))
        34    # Is Kate an instance of Bulldog()
        35    Kate = RussellTerrier("Kate", 4)
        36    print(isinstance(Kate, Dog))
        37    # Is thunder_kid and instance of kate?
        38    print(isinstance(thunder_kid, Kate))
        39    print("Thanks for understanding the concept of OOPs")

```

Thunder is 9 years old

Thunder runs slowly

True

True

True

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-4-9d04014cdcd2> in <module>
    36 print(isinstance(Kate, Dog))
    37 # Is thunder_kid and instance of kate?
--> 38 print(isinstance(thunder_kid, Kate))
    39 print("Thanks for understanding the concept of OOPs")

```

TypeError: isinstance() arg 2 must be a type or tuple of types

```

In [1]: 1 # Prog 3(without error)
        2 # Parent class
        3 class Dog:
        4     # Class attribute
        5     species = 'mammal'
        6     # Initializer / Instance attributes
        7     def __init__(self, name, age):
        8         self.name = name
        9         self.age = age
        10    # instance method
        11    def description(self):
        12        return "{} is {} years old".format(self.name, self.age)
        13    # instance method
        14    def speak(self, sound):
        15        return "{} says {}".format(self.name, sound)
        16
        17    # Child class (inherits from Dog() class)
        18    class RussellTerrier(Dog):
        19        def run(self, speed):
        20            return "{} runs {}".format(self.name, speed)
        21    # Child class (inherits from Dog() class)
        22    class Bulldog(Dog):
        23        def run(self, speed):
        24            return "{} runs {}".format(self.name, speed)
        25
        26    # Child classes inherit attributes and behaviors from the parent class
        27    thunder = Bulldog("Thunder", 9)
        28    print(thunder.description())
        29
        30    # Child classes have specific attributes and behaviors as well
        31    print(thunder.run("slowly"))
        32
        33    # Is thunder an instance of Dog()?
        34    print(isinstance(thunder, Dog))
        35
        36    # Is thunder_kid an instance of Dog()?
        37    thunder_kid = Dog("ThunderKid", 2)
        38    print(isinstance(thunder, Dog))
        39
        40    # Is Kate an instance of Bulldog()
        41    Kate = RussellTerrier("Kate", 4)
        42    print(isinstance(Kate, Dog))
        43
        44    print("Thanks for understanding the concept of OOPs")

```

```

Thunder is 9 years old
Thunder runs slowly
True
True
True
Thanks for understanding the concept of OOPs

```

In []:

1

