

Python Class

Python classes provide all the standard features of object oriented programming, the class inheritance mechanism allows multiple base classes, a derived class can override any methods of its base class and a method can call the method of a base class with the same name. The following syntax of python class.

```
class ClassName:
    <statement-1>
    .
    .
    .
    <statement-N>
```

Class Object

Class object support two kinds of operations - attribute references and instantiation. attribute references use the standard syntax used for all attribute references in python - obj.name. Valid attribute names are all the names that were in the class's namespace when the class object was created. So, if the class definition looked like this.

```
class MyClass:
    a = 10

    def f(self):
        return 'Welcome'
```

Example

```
class Fruit(object):
    """A class that makes various tasty fruits."""
    def __init__(self, name, color, flavor, poisonous):
        self.name = name
        self.color = color
        self.flavor = flavor
        self.poisonous = poisonous

    def description(self):
        print "I'm a %s %s and I taste %s." % (self.color, self.name, self.flavor)

    def is_edible(self):
        if not self.poisonous:
            print "Yep! I'm edible."
        else:
            print "Don't eat me! I am super poisonous."

lemon = Fruit("lemon", "yellow", "sour", False)

lemon.description()
lemon.is_edible()
```

Output

```
I'm a yellow lemon and I taste sour.  
Yep! I'm edible.  
None
```