Class and Module

Sunday, February 05, 2017 9:55 AM

1. **Class and Object**

   An object combines variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially templates for creating your objects. You can think of an object as a single data structure that contains data as well as functions. Functions of objects are called methods.

   a. Define a Class

```
Class MyClass:
    variable=12

    Def foo(self):#we'llexplainselfparameterlaterintask4
        print("Hellofromfunctionfoo")

my_object=MyClass()

my_object.foo()#will print "Hello from functionfoo"
print(my_object.variable)#will print the value 12 we assigned before
```

   b. Class variable

```
Class Calculator:
    current=0

    Def add(self,amount):
        self.current=self.current+amount

    Def get_current(self):
        Return self.current


my_cal=Calculator()
my_cal.add(123)
print(my_cal.current)
```

2. **Module and Packages**

   Modules in Python are simply Python files with the .py extension containing Python definitions and statements. Modules can be handy when you want to use your function in a number of programs without copying its definition into each program. Modules are imported from other modules using the **import** keyword and the file name without an extension. The first time a module is loaded into a running Python script, it is initialized by executing the code in the module once.

   a. Import your module

      i. Define a function in moudle my_module.py

```
Def hello_world(name):
    print("Hello,World!Mynameis%s"%name)
```

      ii. In a nother py file, import and use your module

```
Import my_module
my_module.hello_world("wei")
```

   b. Import buildin module,

      There are lot of build modules: https://docs.python.org/3/library/

```
Import datetime
print(datetime.date.today())
```