## 模組module

python模組的組成有可能只是一個py檔，也有可能是一個由資料夾所組成的package

### import module path

python try to fetch module from sys.path :

>>>import sys

>>> sys.path

['', '/usr/lib/python3.4', '/usr/lib/python3.4/plat-x86\_64-linux-gnu', '/usr/lib/python3.4/lib-dynload', '/usr/local/lib/python3.4/dist-packages', '/usr/lib/python3/dist-packages']

how to add new path to sys.path:

1. 設定env PYTHONPATH的方式來新增尋找module的路徑
2. 把路徑放在一個.pth檔裏面:

In test.pth: /home/<username>

然後把這個test.pth檔加進'/usr/local/lib/python3.x/dist-packages'裏面

## JSON

python json 模組有四个方法：dumps和loads、dump和load。其中，dumps和loads是在内存中转换（python对象和json字符串之间的转换），而dump和load则是对应于文件的处理。

json.loads()

json.load()

dict obj

json.dump()

.json file

string obj

json.dumps())

## repr與str

python中每一個對象都會去implement \_\_str\_\_還有\_\_repr\_\_兩個方法，這兩個方法都會回傳一個代表這個對象的字串，而如果調用了print function，其輸出是\_\_str\_\_返回的字串，若是直接輸入變數，輸出則為\_\_repr\_\_返回的字串

## Class

### 傳遞dict內容至物件

using **setattr**

class Dict2Obj(object):

    """

    Turns a dictionary into a class

    """

    #----------------------------------------------------------------------

    def \_\_init\_\_(self, dictionary):

        """Constructor"""

        for key in dictionary:

            setattr(self, key, dictionary[key])

    #----------------------------------------------------------------------

    def \_\_repr\_\_(self):

        """"""

        attrs = str([x for x in self.\_\_dict\_\_])

        return "<dict2obj: %s="">" % attrs

#----------------------------------------------------------------------

if \_\_name\_\_ == "\_\_main\_\_":

    ball\_dict = {"color":"blue",

                 "size":"8 inches",

                 "material":"rubber"}

    ball = Dict2Obj(ball\_dict)

## setattr 以及 getattr 的用法

setattr() 可以『設定』 class 內定義變數的值，getattr() 則是可以『取得』class 內定義變數的值

setattr(object, name, value)

getattr(object, name[, default]) //default is optional , when obj.name does not exist, it return default , or it throw AttributeError