

Python初探

Roy Chen

什麼是Python？

Python 自西元 1989 年 Guido van Rossum 開始開發，也由於 Python 簡單、易學的特性，使許多大學紛紛採取作為入門的第一種程式語言

Python 是一種強大的物件導向程式語言 (object-oriented programming language)，具有非常廣泛的功能，例如用作網路、遊戲、伺服器、科學計算、軟體發展、3D 繪圖等等，幾乎各方面應用皆可使用 Python，相關資源非常多。

Python的特色

- 容易撰寫：Python 具有許多物件導向的特性。
- 功能強大：從簡單的數學運算、字串處理、網際網路協定連線、網際網路資料處理、各種壓縮格式，以及 POSIX 與主要作業系統的支援功能等等。
- 跨平台：各種主要的作業系統都支援 Python。
- 容易擴充：Python 算是執行效率不錯的直譯式語言。

Python的執行

- 由於 Python 是直譯 (interpret) 式的程式語言，因此無須編譯 (compile)

Console運作方式：在命令列下打python
即可啟動

```
Last login: Thu Oct 15 11:41:09 on ttys000
RoydeMacBook-Pro:~ roy$ python
Python 2.7.9 (default, Aug 23 2015, 12:15:55)
[GCC 4.2.1 Compatible Apple LLVM 6.1.0 (clang-602.0.53)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

執行檔運作方式：檔名為demo1.py，在
命令列下打python demo1.py即可執行

```
RoydeMacBook-Pro:py_example roy$ python Function1.py
1
RoydeMacBook-Pro:py_example roy$ █
```

使用Python

- 基礎型別
- 數值與運算子
- 變數與物件
- 邏輯決策
- 函式
- 物件與類別

基礎型別

Python提供的型別

- Number(數值) = { Integer, Float, Imaginary number }
- String(物件)
- Group(物件) = { List, Tuple, Dictionary }

什麼是imaginary number

複數的符號表示相當的簡單，通常寫作： $a+bi$ ，這裡的 a 跟 b 是實數，而 i 是虛單位，最重要的性質就是 $i^2=-1$ ，因此我們可以知道，所謂的實數，其實就是虛部為零的複數，所以在Python要表示複數就可以使用：`complex(x, y)`來表示。 x 代表實部的數字， y 代表虛部

Example：

```
>>> complex(2,3)  
(2+3j)
```

數值與運算子

- Python廣泛地將數字以Number(數值)類別包裝起來，而Number下面只有三種型態 {Integers, Floats, Imaginary number}
- 基本運算子 $+$, $-$, $*$, $/$, $\%$ 皆能使用於Python內

Example :

```
>>> 1+2  
3
```


變數與物件

- 不同型態的物件如何運作
- 什麼是Reference

什麼是List

List為串列的資料結構，裡面的成員可以是任何的物件，數字、字串、其他串列，使用者可以自行新增，刪除，插入此串列

Example：

```
>>> l=["a",["b","c"],1]  
>>> print l  
['a', ['b', 'c'], 1]
```

List的相關操作

- **len**: 回傳List的長度

```
>>>l=["a",["b","c"],1]
>>>len(l)
3
```
- **slicing** : 產生子串列

```
>>>l=["a",["b","c"],1]
>>> s1 = l[1:2]
>>> print(s1)
[['b', 'c']]
```
- **append** : 在List尾端加入一個新元素

```
>>> l.append("h")
>>> print(l)
['a', ['b', 'c'], 1, 'h']
```
- **pop** : 回傳最後一個元素，並將該元素從List中移除掉

```
>>> l.pop()
'h'
>>> print(l)
['a', ['b', 'c'], 1]
```
- **remove** : 移除某個元素

```
>>>l.remove("a")
[['b","c"],1]
```

什麼是Tuple

- Tuple也是串列的一種，裡面的成員可以是任何的物件(Object)，數字(Number)、字串(String)、其他串列(List)或Tuple
- List與Tuple非常類似，唯一的差別在於: 串列可以修改，而Tuple不能修改。

Example :

```
>>> l=("a","b","c")
```

```
>>> l[0]
```

```
a
```

什麼是Dictionaries

Dictionaries類似List和Tuple，也是用於群組資料的容器，而但為Key而不是一般的數字，透過索引可以找出對應的值

Example:

```
>>>fruit = {"apple":1, "orange":2}
```

```
>>>fruit["apple"]
```

```
1
```

邏輯決策

- If...else : Condition
- While : Repetition
- For : Repetition
- Error handling

If...else Statement

If 陳述 (statement) 後空一格，到冒號：之間為條件 (condition)

Example:

```
if  $a==b$ :  
    print("a equal b")  
else :  
    print("a not equal b")
```

If...elif...else Statement

Example:

if a==b:

print(“a equal b”)

elif a>b:

print(“a large than b”)

else:

print(“a small equal than b”)

While Statement

While陳述(statement) 後面加條件式(condition)，如果條件式為 True，則執行Block內的函式，否則就離開此陳述

while *condition* :

Block

Example:

count = 10

i=0

while *i*<*count*:

 print *i*

i=*i*+1

For Statement

For 迴圈只需要取得元素的控制變數 (variable) ，其餘條件 (condition) 測試與調整控制變數的部份，迴圈中會自動完成。

for *a* in *val* :

Block (with variable *a*)

Example :

```
list = ["a","b","c","d"]
```

```
for a in list:
```

```
    print a
```

其他陳述

- **continue**：直接進入下一個迴圈
- **break**：跳出迴圈
- **return**：函數 (function) 中若使用 **return**，函數會直接回傳數值 (value)，也隨之終止函數執行。
- **pass**：什麼都不做的statement

例外處理

凡是可能會產生例外的程式碼，Python 利用 **try-except** 陳述 (try-except statement) 讓程式設計師自行處理例外。 **try-except** 為關鍵字 (keyword) 之一，專門用來例外處理 (exception handling) 的

try

可能會發生錯誤的區塊

catch:

錯誤發生時的處理區塊

else:

沒有錯誤發生時的處理區塊

finally:

無論有沒有錯誤發生都會執行的區塊

函式

函數 (function) 物件 (object) 可以執行一些工作，或是進行計算。
Python 中定義函數使用關鍵字 (keyword) **def**，其後空一格接函數的識別字 (identifier) 名稱加小括弧，然後冒號

```
def identifier(val):  
    block
```

Example :

```
def test(a):  
    print(a)
```

函式範例

```
def comparison( $a$ ,  $b$ ):  
    if  $a > b$ :  
        print(“a large than b”)  
    elif  $a < b$ :  
        print(“a small than b”)  
    else:  
        print(“a equal b”)
```

物件與類別

物件與類別的關係就如人與人類的關係，物件代表的是個體，每個都不一樣，每個物件都有三個元件來區分其個體性

- Identity：唯一的代碼
- Type：物件的型態
- Value：內容

何謂類別(Class)

- 在物件導向語言的結構
- Class內的元素
 1. 值(Field)
 2. 方法(Method)
- 建構子(Constructor)為Class被new出來所要執行的函式

Example:

class Car:

Field

***color** = ""*

Constructor

*def **__init__**(self):*

color = "red"

Method

*def **turnRight**(self, rad):*

print("turn right")

def turnLeft(self, rad):

print("turn left")

模組與套件

- 模組就是已經寫好的 Python 程式檔案，我們在需要的時候使用 **import** 到我們自己的程式中就可以使用相關定義，同樣的，使用標準模組庫中的模組也要 **import**，例如下例 **import** 標準模組庫中的 `random` 模組
- Python 已經內建大多數常用的功能，更多的功能都放在標準模組庫 (standard library) 裡

套件與模組的Import

- 一個 .py 檔案就是一個模組 (module)，可以有多個類別
- 如何Import類別：

class_car.py

```
class Car:
    color = ""
    def __init__(self):
        color = "red"
    def turnRight(self, rad):
        print("turn right")
    def turnLeft(self, rad):
        print("turn left")
```

import_class.py

```
from class_car import Car

car = Car()
car.turnRight()
```



在OpenWrt下安裝python 的 Websocket套件

步驟

- 登入OpenWrt
- mkdir plugging
- cd plugging
- `curl https://bootstrap.pypa.io/ez_setup.py -k -o - | python` //安裝setuptools
- `wget --no-check-certificate https://pypi.python.org/packages/source/s/six/six-1.10.0.tar.gz` //下載six套件
- `tar zxvf six-1.10.0.tar.gz` //解壓縮six
- `cd six-1.10.0`
- `python setup.py install` //安裝six
- `wget --no-check-certificate https://pypi.python.org/packages/source/w/websocket-client/websocket_client-0.32.0.tar.gz` //下載websocket套件
- `tar zxvf websocket_client-0.32.0.tar.gz` //解壓縮websocket
- `python setup.py install` //安裝websocket套件