Topic04_ 學習目標

- import 內建模組
- def myFunction()
- print() 和 return
- import 自構模組

自己寫個小工具

- 程式設計師們叫它 "function" (函式)

- 之前已經看過的小工具：
  - str(whateverinput) => return "whateverinput"
  - int(digitInput) => return ...-1, 0, 1, ...
- 回想一下課程裡看到的程式碼，每次都要開檔，讀檔再關檔，好煩哦，所以可以用 def 自己定義一個小工具，

```
def file2Text(textFileName):
    theFILE = open(textFileName)
    theTEXT = theFILE.read().decode("utf-8")
    theFILE.close()
    return theTEXT
```

使用外部工具
-import (引進) 現成的工具

- import myTools

  sampleText = myTools.file2TEXT("sample.txt")
- prompt 對 return 的反應
- editor 對 return 的反應
- 對引入的工具取個暱稱：

  import myTools as mt

  sampleText = mt.file2TEXT("sample.txt")

  print sampleText

自己再寫個小工具

- 有的程式語言稱之為「方法」 (method)

- 用 def 自己定義一個小工具，可以把字串中的數字抓出來：

```
def numberExtractor(inputSTR):
    #version 01
    resultList = [ ]
    for character in inputSTR:
        if character.isdigit():
            resultList.append(character)
        else:
            pass
        return resultList
```

- 遇上中文數字時怎麼辦？

再談記憶體

- 其實不講「記憶體」也可以啦！

- 資料的儲存，除了用方括號 [ ] 表示的列表外，另外有兩個很類似的容器。一個是由圓括號 ( ) 構成的 Tuple ，另一個則是由花括號 { } 構成的 Dictionary 。
- ( ) Tuple 的用法和 [ ] 列表很像，但它的內容不增不減，也無法更動。這種「一次寫死」的特性，使得它比較節省一點記憶體，在大量運算時的速度也會快一點。
- { } Dictionary 的用法則剛好反過來。它和 [ ] 列表完全不一樣，內容隨時可以增減，所以在大量運算時的速度是最慢的。但它像字典一樣，對人類來說是「最容易讀」的一種資料儲存方式。

Tuple (元組)

- 不能更動內容的列表

- ( ) Tuple 和 [ ] List 的比較：
  - "one" in ["one", "two", "three"]
  - "one" in ("one", "two", "three")
  - numList = ["one", "two", "three"]
  - numList.append("four")
  - len(numList)
  - numTuple = ("one", "two", "three")
  - numTuple.append("four")
  - len(numTuple)
  - numTuple = ("one", "two", "three", "four")
  - print numList[2]
  - print numTuple[3]

升級自己的小工具
- 抓出中文數字

- 用 def 自己定義一個小工具，可以把字串中的阿拉伯數字和中文數字抓出來：

```
def numberExtractor(inputSTR):
    #version 02
    cNumberTuple = (u"一", u"二", u"三", u"四", u"五", u"六", u"七", u"八",
    u"九", u"零")
    resultList = [ ]
    for character in inputSTR:
        if character.isdigit():
            resultList.append(character)
        elif character in cNumberTuple:
            resultList.append(character)
        else:
            pass
        return resultList
```

Dictionary { 字典 }

- 依名字（key）取得內容（value)

- ( ) Tuple 、 [ ] List 和 {} Dictionary 的比較：
  - numList = ["one", "two", "three"]
  - numList.append("four")
  - numTuple = ("one", "two", "three")
  - numTuple = ("one", "two", "three", "four")
  - print numList[2]
  - print numTuple[3]
  - numDict = {"one": 1, "two": 2, "three": 3}
  - numDict["four"] = 4
  - numDict[2]
  - numDict["two"]

More about Dictionary { 字典 }

- Dictionary 的「名字」 (key) 與「內容」 (value) 一定是成雙成對地出現。
- PeterProfileDict = {"name": "Peter",
  "cellNo: "09xx000xxx",
  "birth" : "1980.xx.00",
  "edu" :[" 小學 ", " 國中 "]
  }
- PeterProfileDict.keys()
- PeterProfileDict.values()
- PeterProfileDict.has_key("job")
- len(PeterProfileDict)
- "edu" in PeterProfileDict

再升級自己的小工具
- 附上原始資料

- 用 def 自己定義一個小工具，可以把字串中的阿拉伯數字和中文數字抓出來之外，還顯示原始輸入資料：

```
def numberExtractor(inputSTR):
    #version 03
    cNumberTuple = (u"一", u"二", u"三", u"四", u"五", u"六", u"七", u"八", u"九", u"零")
    resultList = [ ]
    resultDict = {}
    for character in inputSTR:
        if character.isdigit():
            resultList.append(character)
        elif character in cNumberTuple:
            resultList.append(character)
        else:
            Pass
        resultDict["input"] = inputSTR
        resultDict["output"] = resultLIST
        return resultDict
```

測試自己的小工具
- debug debug debug

- 假設我有兩個字串長這樣：

  KennyBeerSTR = "Kenny had 16 glasses of beer last night."

  VivianBeerSTR = "Vivian had 12 glasses of beer last night."
- 誰昨天喝了的啤酒比較多？

  KennyBeerNumber = numberExtractor(KennyBeerSTR)

  VivianBeerNumber = numberExtractor(VivianBeerSTR)

  print KennyBeerNumber

  print VivianBeerNumber
- 這樣有辦法比大小嗎？

升級自己的小工具
- 實作

- Version 03 的 numberExtractor() 回傳的答案是裝在列表裡，而列表裡面都是數字的「字串」！
- 希望能達到的目標：
  - 列表裡的數字字串能「串」成「一個」字串。
  - 串在一起的字串能「轉型」成數字。
  - 回傳 (return) 這個轉好型態的數字。

課後練習 2-1 (完成者，本次作業分數 75 起計)

- 請建立一個名為 "homework.py" 的 python 程式。
- 在 homework.py 中加入適當的程式碼，讓程式執行以後可以達成以下要求：
  - 在 python prmpt 中被 import
  - 內含一個叫 mandarinStrOpener(xFILE) 的函式。其功能為讀入一個純文字檔案做作參數，直接回傳該純文字檔案內文之 utf-8 編碼字串，而且字串中不含 "\n" 字元。

課後練習 2-2(完成者，本次作業分數 80)

- 請設計一個工具 (function) ，能把 0 ~ 99999 之間任一個阿拉伯數字轉成「中文大寫」。
  - 例如：
    - ChineseNumber(15) =>return " 十五 "
    - ChineseNumber(15001) =>return " 一萬五千零一 "

本節完成目標

- import 內建模組
- def myFunction()
- print() 和 return
- import 自構模組