

# 物件導向軟體設計 期中筆記

Created by Weber, YC Huang 2019-11-04

## 1. Git

名詞	英文	功能
數據庫	Repository	紀錄檔案或目錄狀態的地方
提交	Commit	將變更與新增的檔案/目錄儲存到數據庫中，提交後數據庫所產生上次提交與這次的差異，稱為 Revision
工作目錄	Working tree	保存目前正在處理檔案之目錄
索引	Index	位於工作目錄與數據庫之間，是為了向數據庫提交做準備之暫存區域。Git 執行提交時候，不是直接將工作目錄的狀態儲存至資料庫，而是將索引的狀態儲存到數據庫。

## 2. Basic Python

2.1 程式設計錯誤類型：

- 語法錯誤 (syntax error)
- 執行期間錯誤 (runtime error)
- 邏輯錯誤 (logic error)

2.2 變數：取代儲存於記憶體的位置

- 英文大小寫視為不一樣之變數
- 僅能由英文大小寫、數字、中文、\_ 組為變數名稱
- 變數名稱第一個字元必須是大小寫英文、\_、中文

2.3 註解：單行 # 多行 '''、'''

2.4 type()：取得資料型態，資料型態轉換 型態()，如 int()

2.5 運算子：+, -, \*, /, %(取餘數), //(取商), \*\*(次方)

2.6 input & print：

- 注意 input() 預設為字串型態，若要轉換型態為整數要加上 int()，同理浮點數
- print(項目1,項目2,..., sep = 分隔字元, end = 結束字元)  
sep 預設為 " ", end 預設為 '\n'

2.7 參數化格式：

- %d : 整數資料型態輸出
- %f : 浮點數資料型態輸出
- %s : 字串資料型態輸出
- %e or %E : 科學記號資料型態輸出

## 2.8 程式結構：

- Sequence structure :
- Selection structure : 例如，邏輯判斷式
- Iteration structure : 例如，迴圈

## 2.9 迴圈函數

- for 迴圈常用於已知迭代次數，while 常用於迭代次數不明
- range(i) : 執行i次
- range(i,j,k) : i 起始，j 終止，k 間隔
- break : 強迫迴圈結束
- continue : 強迫跳回迴圈起頭

# 3. Advanced Python

## 3.1 List : list 內元素資料型態可相同也可不相同

- index : 索引值從 0 開始 (cf. R語言)，索引值也可以是負值，-1 代表最後一個元素
- 改變串列元素 list[i] = 'new\_element'
- 迴圈讀取：
  - for item in list :
  - for i in range(len(list))

功能	回傳值
list.index('元素')	返回該元素索引(index)值
list.count('元素')	返回該元素累計次數
list.append('元素')	新增該元素於 list 末
list.insert(i, '元素')	於list i 位置插入該元素，若索引值 i 超過元素長度則等同於 append()
list.remove('元素')	刪除該元素
list.pop(i)	刪除該索引值所對應的值，若 pop() 無填寫索引值，則默認移除最後一個值
del list[i : j : k]	刪除 range i~j，間隔為 k 之值
list.sort()	由小到大排序
list.reverse()	反轉 list 原來之序列
sorted(list, reverse=True)	排序，結果並不會影響原來 list，reverse=True 則降冪，反之升冪

## 3.2 tuple : 使用方式與 list 相同，但「無法更動元素值」。可以透過 list() , tuple() 自由變更資料型態

## 3.3 dictionary :

- dictionary 建立方式：
  - dicts = {key1:val1, key2:val2, ...}, string, int, float 皆可以為 key
  - dicts = dict([[key1,val1], [key2,val2], ...])
  - dicts = dict(key1=val1, key2=val2, ...)
- dictionary 取值：
  - dicts[key]
 

dictionary 使用 key 當作取值索引，key 必須是唯一。當 dictionary key 有重複時，dicts[key] 前面的 key 對應 value 會被覆蓋，僅回傳最後一個的 value；當 dictionary 取值的 key 不存在時，回傳為所有 key value 隨意排序。此法若 key 不存在即發生錯誤
  - dicts.get(key, default = )
 

此法若 key 不存在也不會發生錯誤，若有設定預設值 default，若 key 不存在返回預設值，若無設定預設值，返回 None
- dictionary 維護：
  - dicts[key] = value：修改與新增相同，若該 key 存在即為修改，不存在則新增
  - dictionary 刪除：
    - del dicts[key]：刪除特定 key
    - dicts.clear()：刪除所有元素
    - del dicts：刪除 dictionary，刪除後該 dictionary 不存在
- dictionary 其他功能：

功能	回傳值
dicts.copy()	複製字典
dicts.len()	返回元素個數
dicts.items()	取得 key - value 組合，資料型態為 dict_items
dicts.keys()	取得 key 組合，資料型態為 dict_keys
dicts.values()	取得 value 組合，資料型態為 dict_values
key in dicts	檢查 key 是否存在 dicts，返回布林值 True, False
dicts.setdefault()	與 dicts.get() 相似，差別在於沒有該 key 時，也無預設值，可能改變該字典內容(cf. dicts.get()不會改變字典內容)

dictionary 無排序規則，以上取 values、取 keys，若要進一步處理，均須透過 list() 轉為 list，無法用索引方式 dicts[i] 取單一元素

dicts.items() 透過 list() 轉換為 2 維list

## 4. Function

### 4.1 自訂函式：

- 若函式有多個回傳值，必須使用相同數量之變數來儲存，之間以 ， 相隔
- 函式要在開始執行前建立好
- 設定預設值之參數必須放在參數最後，如 def progression(start, stop, step = 1)：
- 全域變數 vs 區域變數 (迴圈也屬於區域)，函式內相同名稱之全域與區域變數，區域優先使用
- lambda：

lambda parameter1, parameter2, ... : 運算 A if 關係運算 else 運算 B

- 如 : f = lambda n1, n2 : n1 if n1 > n2 else n2 (回傳大的值)
- 沒有函式名稱，只能一行，不能夠使用迴圈與指定運算子，不需要使用 return
- 常搭配 filter() 與 map() 一起用，filter() 用來篩選資料中需要的特定資料；map() 將給定資料輸入函式返回新值，不會減少資料數量

#### 4.2 import :

- import package
- import package as pack
- import package as p
- from package import \*
- from package import p
- 常用套件：
  - import random as r

功能	回傳值
r.choice(str)	從字串中隨機取得字元
r.randint(i, j)	由 i 到 j 中，隨機取一整數
r.sample(str, n)	由字串中隨機取n個字元
r.shuffle(list)	串列位置洗牌
r.uniform(i, j)	由 i 到 j 中，隨機取一浮點數

- import time as t

功能	回傳值
t.clock()	取得程式執行時間
t.sleep(n)	程式停止執行n秒
t.time()	取得目前時間數值
t.ctime()	取得時間字串
t.localtime()	取得時間元組資訊，與 tm 一起應用