

Lecture7

前情提要

- 兩大類 loop
- for -> 通常跑固定次數
 - 很常搭配 range()
 - range(10) 和 range(0, 10) 是相同的
- while -> 想成一直執行直到條件不成立的 if
 - while 比較容易出現 infinite loop

繼續 loop

- break -> 跳出 loop，通常搭配 if, elif, else 使用
- continue -> 直接開始下一輪 loop，無視後面的程式碼，一樣搭配 if, elif, else
- 注意！break 和 continue 都只能用在 loop 之中，不然會被認定是錯誤！

Loop 中常見的特別寫法

- 看得出來這些可以轉換成我們比較熟悉的形式呢？

1. $y += 2$ $y = y + 2$

2. $-=$

3. $/=$

4. $*=$

要開始進入困難的地方囉

Function

- 新名詞？
- 事實上，你已經見過他很多次了
- `print()`, `len()`, ... 這些都是 function
- 只是他們都是內建的 (built in functions)
- 那如果我們想自己寫 function 來達成想要的功能？

自定義 function

function name

argument(s): 0

~ 多個

```
def my_func(x):  
    for i in range(10):  
        x += i  
    return x
```

回傳值: 非必要

Global Variable VS Local Variable

global variable

← `y = 0`

local
variable

```
def my_func(x):  
    for i in range(10):  
        x += i  
    return x
```

function
裡面

Global Variable VS Local Variable (cont'd)

- function 可以想像是一道隔離牆，裡外是不同的世界，外面進得來，裡面出不去
- Global Variable
 - 在 function 外面，function 裡外皆可用
 - 小提醒：不建議直接在 function 中使用！為什麼？
- Local Variable
 - 在 function 裡面，出去就沒人認得了！
 - 如果 local 跟 global variable 撞名 -> 只認 local

Default Argument(s)

- 就好像進入遊戲畫面的預設難度一樣
- 只要沒有更動，就會輸出 default 的值

default



```
def my_func(x, a = 100):  
    for i in range(10):  
        x += i  
    print(a)  
    return x
```

有點特別的東西

長度沒有限制的 list

```
def sumOfList(*list_a):  
    listSum = 0  
    for item in list_a:  
        print(item)  
        listSum += item  
    return listSum
```

```
listSum = sumOfList(1, 2, 3, 4, 5)  
print("sum = " + str(listSum))
```

小練習

- 輸入國英數三科
- 印出等第
- 回傳三科平均