



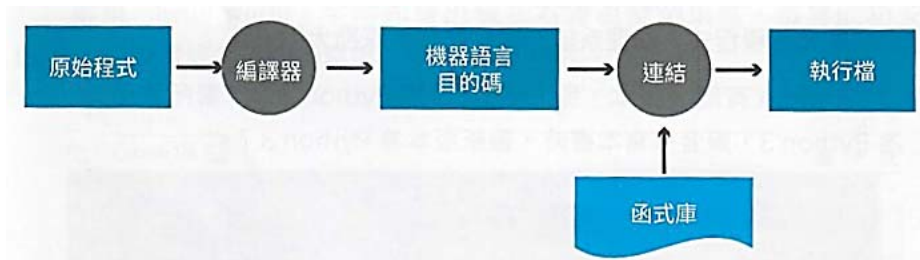
## 2. Python程式語言

- 2.1 認識Python語言
- 2.2 變數、運算子與資料型態
- 2.3 流程控制
- 2.4 檔案操作

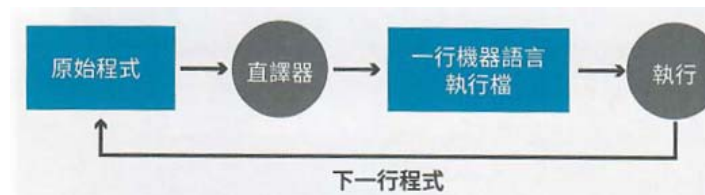
## 2-1認識Python語言

### 1. Python 是直譯式語言

C/C++程式碼需透過編譯器(Compiler)轉為機器語言目的碼，再連結函式庫，最後才會生成執行檔。



Python使用直譯器(Interpreter)來執行，直譯器不會輸出執行檔，而是一行一行轉換成機器語言執行。



直譯器是一行一行轉換執行，所以效率比傳統編譯語言C或C++來的低。

## 2-1認識Python語言

### 2. Python 是動態型態程式語言

「動態型態」意指程式碼內變數不需要宣告資料型態就可以使用。

Python直譯器會根據變數值自動判斷它的資料型態來執行，直譯器不會輸出執行檔，而是一行一行轉換成機器語言執行。

```
1 a = 1                # a為整數
2 b = 1.0              # b為浮點數
3 c = 'hello world'    # c為字串
```

## 2-1 認識Python語言

### 3. Python 是強型態程式語言

Python不須事先宣告變數，但Python為強型態語言，意思是並不會自動進行資料型態轉換

```
1 a = 1
2 b = 1.0
3 c = 'hello world'
4
5 d = c + b
```

# c和b型態不同，不能相加

正確做法應加上`str()`轉換為字串

```
1 a = 1
2 b = 1.0
3 c = 'hello world'
4
5 d = c + str(b)
6 print(d)
```

執行結果

hello world1.0

## 2-1認識Python語言

### 4.Python 使用縮排表示區塊

Python不使用{}來表示區塊(block)，而是使用縮排來表示，  
同一縮排視為同等級區塊，同等級區塊若有不同縮排，則視為錯誤。

```
1 a = 1
2 b = 1.0
3 c = 'hello world'
4
5
6
```

#### 說明

第三行首出現橘色三角形警示，因為多了一個空格，若仍加以執行，則會出現錯誤。

```
5 while a < 1:
6     print (a)
7     a = a + 1
8
```

#### 說明

次級縮排通常預留 2 格或 4 格空格，依使用者習慣而異，只要區塊內縮排一致即可。



## 2.2 變數、運算子 與資料型態

## 2-2變數、運算子與資料型態

### 1.變數

Python變數不須事先宣告，只要指定變數值即可馬上使用，但變數一定要先給初始值

```
1 name = 'Jack'
2 age = 20
3 grade = 87
4 h = 170.0
5 w = 70.0
```

接下來可使用`print()`指令，將值印出來(非字串要加上`str()`)

```
7 print('姓名 = ' + name)
8 print('年齡 = ' + str(age))
9 print('成績 = ' + str(grade))
10 print('身高 = ' + str(h))
11 print('體重 = ' + str(w))
```

執行結果

```
姓名 = Jack
年齡 = 20
成績 = 87
身高 = 170.0
體重 = 70.0
```

## 2-2變數、運算子與資料型態

### 1.變數

接著要**存檔**。要在桌面建立一個空資料夾來儲存檔案，請先打開Windows檔案總管：

#### 步驟1

選擇「開始/所有程式/附屬應用程式/Windows檔案總管」

##### 說明

1. 點擊「桌面」。
2. 空白處按滑鼠右鍵。
3. 點擊「新增 (W)」。
4. 點擊「資料夾 (F)」，會出現資料夾輸入名稱畫面。



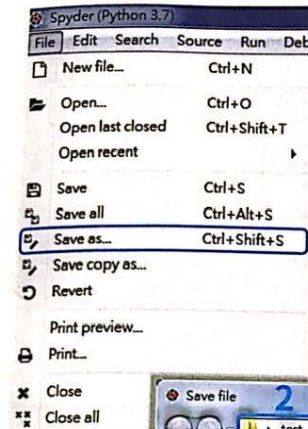


## 2-2變數、運算子與資料型態

### 1.變數

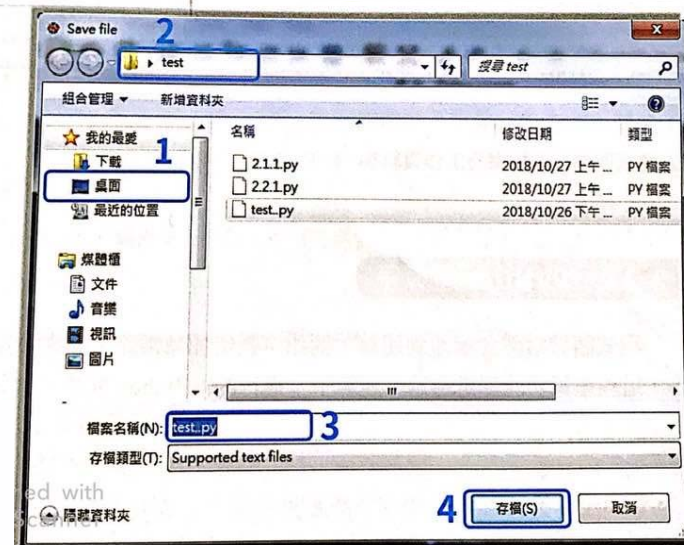
### 步驟2

再回到spyder，點擊  
「file」/「Save as」



#### 說明

1. 點擊「桌面」。
2. 選擇「test」資料夾。
3. 輸入檔名「test.py」。
4. 點擊「存檔 (S)」。

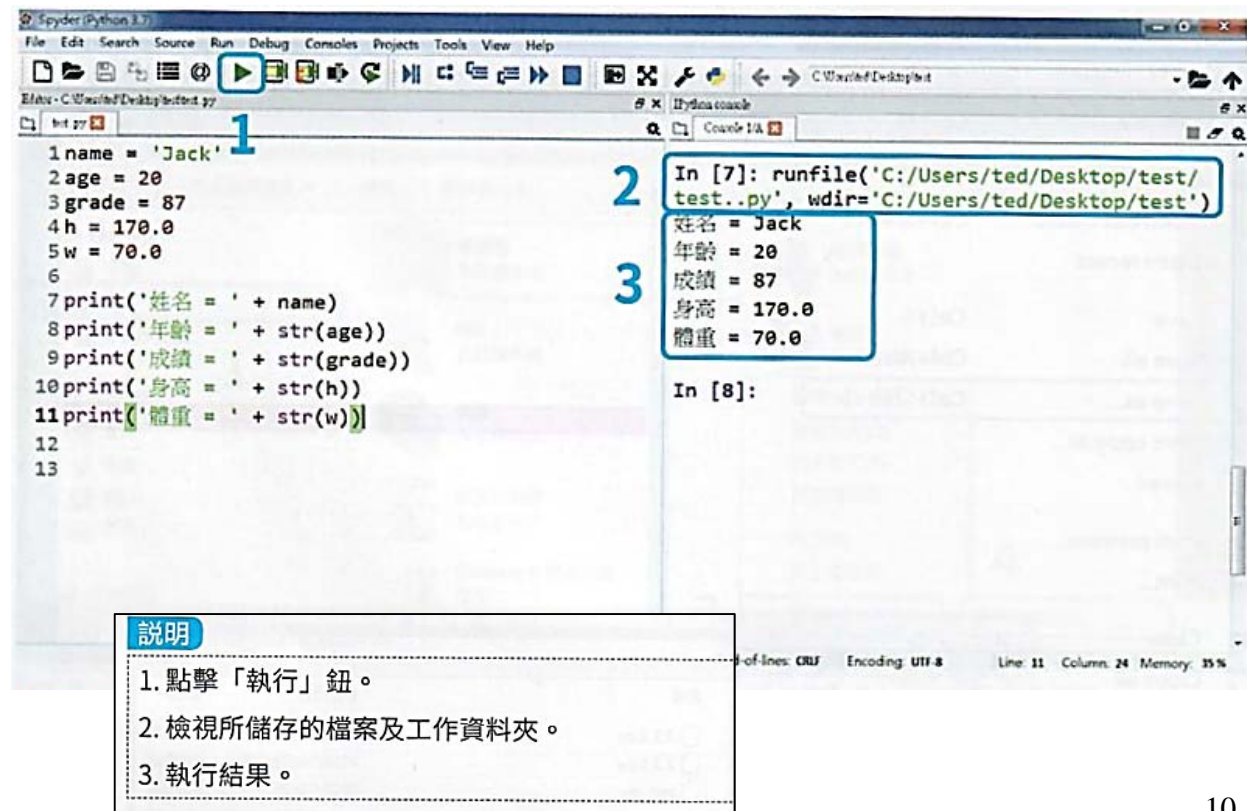


## 2-2變數、運算子與資料型態

### 1.變數

### 步驟3

在spyder中執行程式並確定檔案所儲存位置。



The screenshot shows the Spyder Python IDE interface. The editor window displays a Python script with the following code:

```
1 name = 'Jack'
2 age = 20
3 grade = 87
4 h = 170.0
5 w = 70.0
6
7 print('姓名 = ' + name)
8 print('年齡 = ' + str(age))
9 print('成績 = ' + str(grade))
10 print('身高 = ' + str(h))
11 print('體重 = ' + str(w))
12
13
```

The console window shows the execution of the script, with the following output:

```
In [7]: runfile('C:/Users/ted/Desktop/test/test..py', wdir='C:/Users/ted/Desktop/test')
姓名 = Jack
年齡 = 20
成績 = 87
身高 = 170.0
體重 = 70.0

In [8]:
```

The status bar at the bottom indicates the file encoding is UTF-8 and the current line is 11, column 24, with 35% memory usage.


**說明**

1. 點擊「執行」鈕。
2. 檢視所儲存的檔案及工作資料夾。
3. 執行結果。

## 2-2變數、運算子與資料型態

### 2.Python 運算子

不同運算子優先順序不同，運算子優先順序如下：



運算子	說明
()	括號
**	指數
~	位元: not
+,-	正負號
*,/,//,%	算數:乘、除、整數除法、餘數
+,-	算數:加、減
<<,>>	位元:左移、右移
&	位元:and



運算子	說明
^	位元: xor
	位元: or
in , not in	隸屬成員、非隸屬成員，
is, is not	同一參考，非同一參考
<,<=,>,>=,!=,==	小於，小於等於，大於，大於等於，不等於，等於
not	邏輯:否
and	邏輯:且
or	邏輯:或

## 2-2變數、運算子與資料型態

### 2.Python 運算子

如果運算式包含多個相同優先級別運算子，則順序由左往右運算，以下列運算式為例：（「#」為註解之意，「#」後面的內容不會執行）

**步驟 1** 「-」與「+」優先級別相同，所以先處理左邊「9 - 5」

9 - 5 + 2 # 6

**步驟 2** 接下來繼續處理「4 + 2」，最後得出結果「6」

4 + 2 # 6

## 2-2變數、運算子與資料型態

### 2.Python 運算子

「相同優先級別，順序由左至右」有一例外情況：「多重指定」。「多重指定」執行順序是「由右至左」，以下列運算式為例：

`a = b = c = 50`

執行順序：

**步驟 1** 三個「=」優先級別皆相同，稱為「多重指定」，執行順序由右至左，所以先執行「`c = 50`」

**步驟 2** 接著執行「`b = c`」

**步驟 3** 接著執行「`a = b`」。最後結果是 `a`、`b`、`c` 三個變數值皆為 50

`a = b = c = 50`

`a = b = c`

`a = b`

## 2-2變數、運算子與資料型態

### 3.基本資料型態

Python 資料型態分為「基本型態」「容器型態」

- **基本型態**：整數、浮點數、布林、字串
- **容器型態**：清單、字典、集合、元組

## 2-2變數、運算子與資料型態

### 3.基本資料型態

#### 一. 整數(Integer)

- 整數資料型態沒有小數點

```
1 a = 1
2 b = 100
3 c = 1000000
4 d = -273
```

```
1 a = 10
2 print(type(a)) # <class 'int'>
3 print(a)       # 10
4 print(a + 1)   # 11
5 print(a - 1)   # 9
6 print(a * 2)   # 20
7 print(a / 2)   # 5.0 # 一般除法，結果是浮點數
8 print(a // 2)  # 5    # 整數除法
9 print(a % 2)   # 0
10 print(a ** 2) # 100  # a的2次方
11 a = a + 1
12 print(a)      # 11
13 a += 1
14 print(a)      # 12
15 a = a * 2
16 print(a)      # 24
17 a *= 2
18 print(a)      # 48
```

## 2-2變數、運算子與資料型態

### 3.基本資料型態

#### 二.浮點數(Float)

- 浮點數資料型態是整數加上小數，精確度可達小數點後15位：

```
1 a = 1.0
2 b = 99.99
3 c = 3.1415926
```

- Python浮點數可以執行相關運算範例：

```
1 a = 3.14
2 print(type(a)) # <class 'float'>
3 print(a) # 3.14
4 print(a - 1) # 2.14
5 print(a * 2) # 6.28
6 print(a / 2) # 1.57
7 print(a // 2) # 1.0 # 整數除法
```



## 2-2變數、運算子與資料型態

### 3.基本資料型態

#### 三.布林(Boolean)

- 布林資料型態有兩種，分別為True 和 False：

```
1 a = True
2 b = False
```

- 除了直接使用True/False，以下數值也是為False：
  - 0或0.0
  - ()空元組
  - []空清單
  - {}空字典
  - None

## 2-2變數、運算子與資料型態

### 3.基本資料型態

### 三.布林(Boolean)

- 常使用的邏輯運算子(not, and, or)和關係運算子(==, !=, >, <)的運算結果也是為布林值

```
1 x = True
2 y = False
3 print(type(x))    # <class 'bool'>
4 print(x and y)    # False
5 print(x or y)     # True
6 print(not x)      # False
```

```
1 x = 3
2 y = 4
3 print(x == y)     # False
4 print(x != y)     # True
5 print(x > y)      # False
6 print(x >= y)     # False
7 print(x < y)      # True
8 print(x <= y)     # True
```

## 2-2變數、運算子與資料型態

### 3.基本資料型態

### 四.字串(String)

- 字串是以單引號「'」或雙引號「"」括起來的序列字元

```
1 a = 'hello'
2 b = "world"
3 c = 'Python programming'
```

- 上述三個都是「字串」(string) 資料型態，  
python沒有「字元」(char)資料型態，若引號內字串只有一個字母時即為字元。

```
1 a = 'h'
2 b = "w"
```

## 2-2變數、運算子與資料型態

### 3.基本資料型態

### 四.字串(String)

- 其他字串處理功能：顯示字串、求得字串長度、串接兩個字串、格式化字串

```
1 a = 'hello'
2 b = "Python"
3 print(a + b)          # helloPython
4 print(len(a))         # 5
5
6 c = a + ' ' + b
7 print(c)              # hello Python
8
9 d = '%s %s %f' % (a, b, 3.7)
10 print(d)             # hello Python 3.700000
11
12 e = '%s %s %0.1f' % (a, b, 3.7)
13 print(e)             # hello Python 3.7
14
15 f = '%s %s %.1f' % (a, b, 3.7)
16 print(f)            # hello Python 3.7
```

顯示字串

求得字串長度

串接兩個字串

格式化字串

#### 說明

第 3 行：字串相加（串接）。

第 4 行：len() 為字串長度。

第 9 行：格式化字串，「%s」表示字串，「%f」表示浮點數，「%d」表示整數，相互對應如下：

%s: a

%s: b

%f: 3.7。

第 12 行：格式化字串，%0.1f 表示浮點數取小數下一位。

第 15 行：同第 12 行。

## 2-2變數、運算子與資料型態

### 3.基本資料型態

### 四.字串(String)

```
1 s = 'python'
2 print(s.capitalize())      # Python
3 print(s.upper())           # PYTHON
4 print(s.rjust(10))         #      python
5 print(s.center(10))        #    python
6 print(min(s))              # h
7 print(max(s))              # y
8 print(s.replace('p','c'))  # cython
9 print(s.count('p'))        # 1
10
11 s2 = ' hello python 3.7 '
12 print(s2.strip())          # hello python 3.7
```

#### 說明

第 2 行：capitalize() 首字大寫。  
第 3 行：upper() 全部大寫。  
第 4 行：rjust(10) 向右對齊，全長長度 10。  
第 5 行：center(10) 置中對齊，全長長度 10。  
第 6 行：min() 最小字母。  
第 7 行：max() 最大字母。  
第 8 行：replace() 「p」用「c」取代。  
第 9 行：count() 計算「p」出現次數。  
第 12 行：strip() 刪除前後空格。



## 2-3 流程控制

## 2-3 流程控制

「流程控制」可配合不同條件來執行不同區塊，也能重複執行區塊內的程式碼。主要分成三種：

- **循序控制**：逐行執行、沒有交叉，也沒有繞圈圈，從第一行執行到最後一行
- **條件控制**：就像選擇題，可分為**單選**、**二選一**、**多選一**，依「if」條件子句的運算結果
- **迴圈控制**：重複執行某一個程式區塊，迴圈通常包含一個**迴圈終止條件**，避免無窮迴圈

## 2-3流程控制

### 1.循序控制

逐行執行、沒有交叉，也沒有繞圈圈，從第一行執行到最後一行

```
1 name = 'Jack'
2 age = 20
3 grade = 87
4 h = 170.0
5 w = 70.0

1 x = 3
2 y = 4
3 print(x == y)    # False
4 print(x != y)    # True
5 print(x > y)      # False
6 print(x >= y)     # False
7 print(x < y)      # True
8 print(x <= y)     # True
```



## 2-3 流程控制

### 2. 條件控制

使用「if」條件運算式，配合程式區塊建立決策。

- 單選(if)
- 二選一(if/else)
- 多選一(if/elif/else)

```
1 2
a = int(input('請輸入今天氣溫幾度 : '))
if a > 30: 3
    print('打開冷氣')
    print('或是打開窗戶')
4
print('今天氣溫 ' + str(a) + ' 度')
```

➤ 執行此程式，輸入「20」並按「Enter」

➤ 執行結果

```
請輸入今天氣溫幾度 : 20
今天氣溫 20 度
```

➤ 執行此程式，輸入「36」並按「Enter」

➤ 執行結果

```
請輸入今天氣溫幾度 : 36
打開冷氣
或是打開窗戶
今天氣溫 36 度
```

## 2-3 流程控制

### 2. 條件控制

- 單選(if)
- 二選一(if/else)
- 多選一(if/elif/else)

進一步活用邏輯運算式，當氣溫在30~40度之間，才執行區塊。

```
1
2
3 a = int(input('請輸入今天氣溫幾度 : '))
4 if a > 30 and a < 40:
5     print('打開冷氣')
6     print('或是打開窗戶')
7
8 print('今天氣溫 ' + str(a) + ' 度')
9
```

- 執行此程式，輸入「36」並按「Enter」
- 執行結果

```
請輸入今天氣溫幾度 : 36
打開冷氣
或是打開窗戶
今天氣溫 36 度
```

## 2-3 流程控制

### 2. 條件控制

「if/else 二選一條件」

如果條件運算式為「真(True)」就執行「if」程式區塊，否則執行「else」程式區塊。

- 單選(if)
- **二選一(if/else)**
- 多選一(if/elif/else)

```
1 a = int(input('請輸入今天氣溫幾度 : '))
2 if a > 20:
3     print('騎腳踏車上學')
4 else:
5     print('搭公車上學')
```

- 執行此程式，輸入「36」並按「Enter」
- 執行結果

```
請輸入今天氣溫幾度 : 22
騎腳踏車上學
```

- 執行此程式，輸入「10」並按「Enter」
- 執行結果

```
請輸入今天氣溫幾度 : 10
搭公車上學
```

## 2-3 流程控制

「if/elif/else多選一條件」是「if/else」擴充，多加了「elif」條件判斷。

### 2. 條件控制

- 單選(if)
- 二選一(if/else)
- 多選一(if/elif/else)

```
1 a = int(input('猜拳：1 剪刀 2 石頭 3 布 :'))
2 if a == 1 :
3     print('你出了：剪刀')
4 elif a == 2:
5     print('你出了：石頭')
6 else:
7     print('你出了：布')
```

➤ 執行結果

```
猜拳：1 剪刀 2 石頭 3 布 :1
你出了：剪刀
```

```
猜拳：1 剪刀 2 石頭 3 布 :2
你出了：石頭
```

```
猜拳：1 剪刀 2 石頭 3 布 :3
你出了：布
```

## 2-3 流程控制

### 2. 條件控制

多重「elif」

- 單選(if)
- 二選一(if/else)
- 多選一(if/elif/else)

```
1 a = int(input('請問要點幾號餐 (1~6) : '))
2 if a == 1 :
3     print('你點1號餐：大麥克')
4 elif a == 2:
5     print('你點2號餐：雙層吉事牛肉堡')
6 elif a == 3:
7     print('你點3號餐：8盎司牛肉堡')
8 elif a == 4:
9     print('你點4號餐：麥香堡')
10 elif a == 5:
11     print('你點5號餐：麥克雞塊')
12 else:
13     print('你點6號餐：火腿芝士堡')
```

➤ 執行結果

```
請問要點幾號餐 (1~6) : 1
你點1號餐：大麥克

請問要點幾號餐 (1~6) : 5
你點5號餐：麥克雞塊
```

## 2-3 流程控制

### 3. 迴圈控制

通常使用在已經知道所需重複執行次數場合，擁有計數器變數去追蹤已重複次數，計數器可以遞增或遞減，一直執行到迴圈結束條件成立為止。

- 計數迴圈(**for**)
- 條件迴圈(**while**)

```
1 sum = 0
2 for i in range(10):
3     sum = sum + i
4     print('累計：' + str(sum))
```

➤ 執行結果

```
累計：0
累計：1
累計：3
累計：6
累計：10
累計：15
累計：21
累計：28
累計：36
累計：45
```

## 2-3 流程控制

### 3. 迴圈控制

- 計數迴圈(**for**)
- 條件迴圈(**while**)

#### 說明

range 參數說明：range(start, end, step)

1. start：計數自 start 開始。預設由 0 開始，例如 range(5) 相當於 range(0, 5)。

range( ) 函式	實際範圍
range(5)	0, 1, 2, 3, 4
range(10)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
range(101)	0 ~ 100

2. end：計數直到 end 結束，但不包括 end。例如：range(0, 5) 是 [0, 1, 2, 3, 4]，不包含 5。

range( ) 函式	實際範圍
range(1, 5)	1, 2, 3, 4
range(2, 10)	2, 3, 4, 5, 6, 7, 8, 9
range(51, 101)	51 ~ 100

3. step：為間隔，預設為 1。例如：range(0, 5) 相當於 range(0, 5, 1)。

range( ) 函式	實際範圍
range(1, 10, 2)	1, 3, 5, 7, 9
range(1, 11, 2)	1, 3, 5, 7, 9
range(0, -10, -2)	0, -2, -4, -6, -8



## 2-3 流程控制

### 3. 迴圈控制

- 計數迴圈(**for**)
- 條件迴圈(**while**)

```
1 sum = 0
2 for i in range(1,11):
3     sum = sum + i
4     print('累計：' + str(sum))
```

# 1累加到10 (不包含11)

➤ 執行結果

```
累計：1
累計：3
累計：6
累計：10
累計：15
累計：21
累計：28
累計：36
累計：45
累計：55
```



## 2-3流程控制

### 3.迴圈控制

- 計數迴圈(**for**)
- 條件迴圈(**while**)

➤ 執行結果

```
1 sum = 0
2 for i in range(1,11,2):
3     sum = sum + i
4     print('累計：' + str(sum))
```

```
累計：1
累計：4
累計：9
累計：16
累計：25
```

# 1 + 3 + 5 + 7 + 9 (間隔2，不包含11)

## 2-3 流程控制

### 3. 迴圈控制

需要在程式區塊自己處理計數器變數的變化。

- 計數迴圈(for)
- 條件迴圈(while)

```
1 sum = 0
2 i = 1
3
4 while i <= 10:
5     sum = sum + i
6     print('累計: ' + str(sum))
7     i = i + 1
```

➤ 執行結果

```
累計：1
累計：3
累計：6
累計：10
累計：15
累計：21
累計：28
累計：36
累計：45
累計：55
```



## 2-4 檔案操作

## 2-4檔案操作

### 步驟1

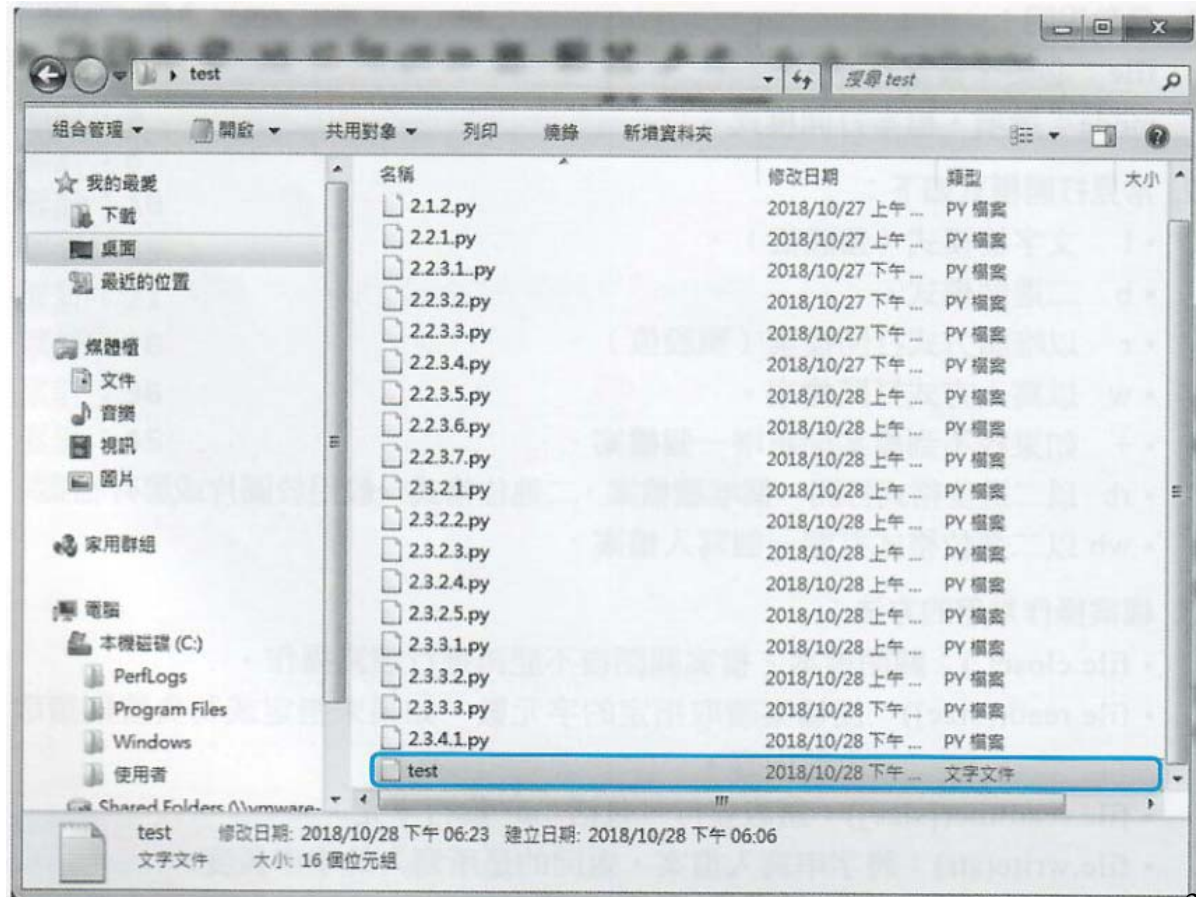
寫“我的第一個檔案！”到名稱為「test.txt」的檔案

```
1 f = open("test.txt", "w+")  
2 print ("檔案名:", f.name)  
3 f.write("我的第一個檔案!")  
4 f.close()
```

## 2-4檔案操作

### 步驟2

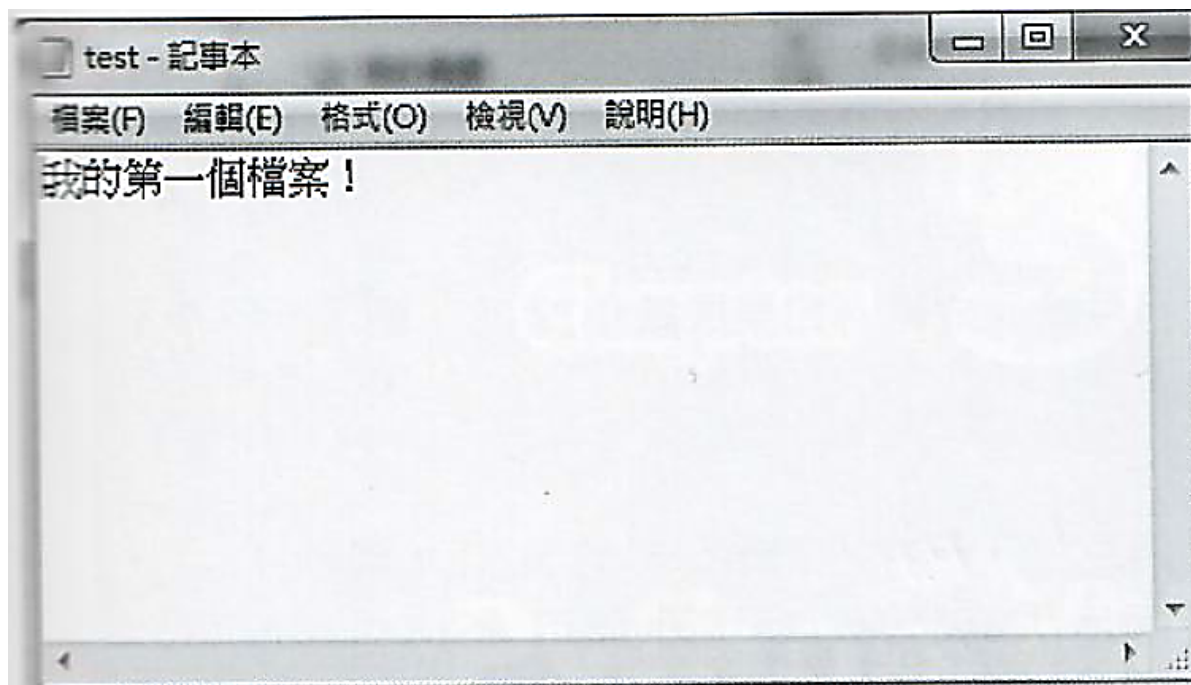
打開檔案總管，尋找  
「test.txt」檔案



## 2-4檔案操作

### 步驟3

雙擊確認「test.txt」  
檔案內容



#### 說明

使用 `open()` 打開檔案時，是為了確保無論是否出錯都能正確地關閉檔案，通常會使用 `with open()`：

## 2-4檔案操作

### 步驟4

新建另一Python程式檔案，  
讀取「test.txt」檔案，  
以下為with open()範例

```
1 with open('test.txt', 'r') as f:  
2     print (f.read())
```

#### 說明

第1行：以唯讀方式開啟文字檔「test.txt」。

第2行：將「test.txt」內容印出來。

```
In [6]: runfile('C:/Users/ted/Desktop/test/  
2.4.2.py', wdir='C:/Users/ted/Desktop/test')
```

我的第一個檔案！

## 2-4檔案操作

### open():方法

- 使用`open()`方法後，最後一定要使用`close()`方法關閉檔案
- 有兩個參數：
  1. 檔案名(file)：必要的，檔案路徑
  2. 開啟模式(mode)：Optional，檔案打開模式

```
1 f = open("test.txt", "w+")
2 print ("檔案名:", f.name)
3 f.write("我的第一個檔案!")
4 f.close()
```

常見打開模式	
t	文字檔模式(預設值)
b	二進制模式
r	以唯獨的方式打開檔案(預設值)
w	以寫入的方式打開檔案
+	如果找不到檔案則新增一個檔案
rb	以二進位格式打開一個唯獨檔案，二進位格式一般用於圖片或影片檔案
wb	以二進位格式打開一個寫入檔案



## 2-4檔案操作

### open():方法

- 檔案操作常用的方法

```
1 with open('test.txt', 'r') as f:  
2     print (f.read())
```

常見打開模式	
file.close()	關閉檔案。檔案關閉後不能再進行讀寫操作
file.read([size])	從檔案讀取指定的字元數，如果未指定或為負值則讀取全部內容
file.readline([size])	讀取整行，包括“\n”換行字元
file.write(str)	將字串寫入檔案，返回的是所寫入的字串長度
File.name	顯示檔案名稱

## 實作練習1:

### 1) 有一串列如下：

```
dept = ['工工系', '運管系', '管科系', '資財系', '資管所',  
        '科管所', '經管所', 'GMBA', 'EMBA']
```

當中有「系」的項目接上「(所)」，有「MBA」的項目接上「學程」，請列出用含編號方式列出這些系所：

1 : 工工系(所)

2 : 運管系(所)

...

9 : EMBA學程

### 2) 請設計一支程式，請使用者輸入一字串，則執行下列：

- 若輸入字元是大寫字母，改成小寫字母輸出。
- 若輸入字元是小寫字母，改成大寫字母輸出。
- 若輸入是其他字元，直接輸出。

### 3) 請使用者輸入五行字串，將資料存到「test.txt」檔案中。