

Python速成班 - 01

Python 緣起

- Python在1989年被發明
- 發明者是一位名為 Guido van Rossum 的荷蘭工程師
 - 因為假期時太無聊，所以開發出了這款程式語言
- 他希望Python能符合他的幾個理想：
 - **能夠跟系統進行互動**
 - **操作簡單**
 - **更貼近人類語言，而非電腦才讀得懂的語言**
- 程式語言有高階、低階之分
 - 越高階的語言，人類越容易讀懂；越低階則越適用於電腦
 - 最低階的語言即為**機器語言**，由許多0和1所組成



若在Python檔案中寫下 `import this`，則會印出Python的設計哲學：

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Python的用途

- 資料科學
- 機器學習
- 網頁後端
- 爬蟲、自動化測試
- 快速開發
 - 提供一個簡單直觀的語言，讓使用者專注在功能而非語法上

變數類型

- Python是一個**動態型別**的語言
 - 與之相對的是**靜態型別**的語言，對於**型態的規則**比較嚴謹
 - 在靜態型別語言中，不能使用整數變數來儲存字串，但在動態型別的語言是可以這樣做的

- 常見的靜態型別語言有 C , C++ 和 Java ；常見的其他動態型別語言有 JavaScript
- Python的基本變數類型僅四種：
 - **int 整數**
 - **float 浮點數**
 - **boolean 布林值**
 - **string 字串**

```
a = 1 # 整數
b = 1.0 # 浮點數
c = True # 布林值
d = "Hello World" # 字串
```

數學運算

```
1 + 1 # 加法
1 - 1 # 減法
1 * 1 # 乘法
1 / 1 # 除法；會得到浮點數
1 // 1 # 取商數
1 % 1 # 取餘數
1 ** 1 # 次方
```

- + , - , * 等運算符號稱作**運算子 (Operator)**
- 運算子旁邊的數字稱作**運算元 (Operand)**

比較運算

```
1 > 1 # 大於
1 >= 1 # 大於等於
1 == 1 # 等於
1 != 1 # 不等於
```

邏輯運算

- True 代表「真」，False 代表「假」
 - and：需兩者皆為True，結果才可以為True

A	B	A and B
0	0	0
1	0	0
0	1	0
1	1	1

- or：其中一個為True，結果就會是True

A	B	A or B
0	0	0
1	0	1
0	1	1
1	1	1

- not：若接收到True，結果就變成False；接收到False則變成True

A	not A
0	1
1	0

```
# 邏輯運算 且、或、非
True and True # 且
True or False # 或
not True # 非
```

條件判斷

- `if ... elif ... else`
- 若滿足一件事情，則....，若滿足另一件事情，則...，若都沒有，則...

```
if 條件A:
    print("條件A成立")
elif 條件B:
    print("條件B成立")
else:
    print("條件A, B皆不成立")
```

- 只要條件成立，就會執行**縮排內**的程式碼；條件不成立則跳過
 - **條件可以輸入數字**；輸入0則代表False，0以外的數字代表True



以下兩種程式碼的判斷結果會一樣

```
if A:
    if B:
        if C:
            D
```

```
if A and B and C:
    D
```

迴圈 Loop

```
while 條件
    print("條件成立")
```

- 可迭代物件：意指可以被迴圈取得的東西

```
for i in range(10):  
    print(i) #第一遍 0 第二遍 1...以此類推至第十遍 9
```

```
# 輸入  
input() # 只讀取一行  
input("請輸入年齡") # 讀取一行 (以"請輸入年齡"作為提示)  
  
# 輸出  
print() # 印出一行並自動換行
```

結合變數使用

```
name = "John"  
print("Hello", name)
```

- 結果為 `Hello John`
- Python 的 operator 會去看左右邊東西的型態，來決定該有什麼動作
 - 若是**數字**，則將兩者相加；若是**文字**，則將文字接在一起

```
name = "John"  
print("Hello " + name) # `+`用在字串上時可以連接字串
```

- 可用input函式接收使用者**輸入**的內容

```
name = input("請輸入名字")  
print("Hello " + name) # 會印出你輸入的名字
```

安裝Python

網址: <https://www.python.org/>

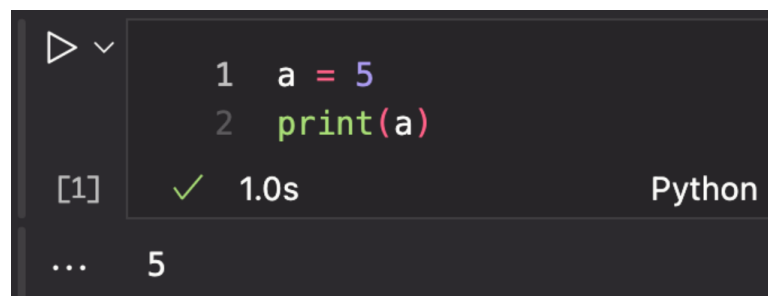
版本: 3.11.2

安裝VSCode

- VSCode是一個具有許多功能的編輯器
- 右下角會提供一些套件，開放給使用者免費安裝

網址: code.visualstudio.com

- 安裝完兩者後，在VSCode裡面新增一個.ipynb的檔案
 - 寫一行 `a = 5` 的程式碼並執行，再把它刪掉
 - 這時寫 `print(a)` 並執行，會發現：即使已經刪掉定義 `a = 5` 的程式碼，結果依然會顯示5
 - 這是jupyter notebook中的暫存行為，即使 `a = 5` 的程式碼已經刪除，`a`的值依然會保留在記憶體中
 - Ex：先執行以下程式，結果會是5



```
1 a = 5
2 print(a)
```

[1] ✓ 1.0s Python

... 5

- 再來把 `a = 5` 註解掉，並且重新執行
- 會發現執行結果依然會是5



```
1 # a = 5
2 print(a)
```

[2] ✓ 0.0s Python

... 5

- 除非 restart kernel，否則程式會記得你寫過什麼
- 新增一個.py檔案，前面取自己想取的名字。Ex：test.py
 - 寫好後輸入 `python (檔案名稱).py`，即可執行

詳細的 I/O 與延伸知識

- 轉換型態：先定義 aaa 為一個字串，再把它轉為整數
 - 使用 `int(字串)` 函式來讓字串轉變成整數型態
 - 再使用 `type(變數)` 函式確認型態轉變前後的結果

```
aaa = "9527"
print(type(aaa)) #<type 'str'>
aaa = int("9527")
print(type(aaa)) #<type 'int'>
```

```
# 把字串轉成數字
a = int("123")
# 把輸入的字串轉成數字
b = int(input("請輸入年齡"))
```

- 特別的輸入方式

```
import sys
a = int(input())
b = int(input())
print(f"兩個數字相加結果是：{a+b}")
```

```
import sys
a = int(sys.stdin.readline())
b = int(sys.stdin.readline())
print(f"兩個數字相加結果是：{a+b}")
```


• 格式化輸出

```
print(f"Hello {name}") # 會印出 Hello John
# 也可以做數字的格式化
print(f"Hello {name}, 你今年 {age} 歲")
```

```
print("Hello", name, "你今年", age, "歲")
# >>> Hello John 你今年 20 歲
```

• 用不同的分隔符號

- 將字串之間從空格改成用【|】進行連結

```
print("Hello", name, "你今年", age, "歲", sep="|")
# >>> Hello|John|你今年|20|歲
```

• 不同的結尾（預設是換行）

- 將字串結尾改成驚嘆號

```
print("Hello", name, "你今年", age, "歲", end="!")
# >>> Hello John 你今年 20 歲!
```

• 組合技

- 會使中間出現分隔符號，並且可自訂義符號款式

```
print(1,2,3)
print(1,2,3,sep=";")
print(1,2,3,sep=";",end="!") # 這裡預設的換行變成!了
print(1,2,3)
```

result :

```
1 2 3
1;2;3
1;2;3!1 2 3
```

- 有趣的玩具 - 倒數計時

```
import time
n = int(input("請輸入倒數秒數："))
for i in range(n, 0, -1):
    print(f"\r剩餘 {n-i} 秒", end="")
    time.sleep(1)
print("\r時間到!")
```

什麼是 Online Judge ?

- 顧名思義：線上評審
- 可以在線上判斷你的程式碼是否正確的網站
- 通常會有一些題目，你可以在網站繳交自己的程式碼，並且由網站判斷是否正確

需要注意的事：

因為是由電腦判斷結果是否正確，任何多印出來的字都會被判斷錯誤，包含但不限於：

- 空白
- 換行
- 輸入提示字（例如：請輸入年齡）等等



請務必參照題目要求，並且只印出題目要求的東西

示範網站：oj.ebg.tw