



01：變數與輸出入

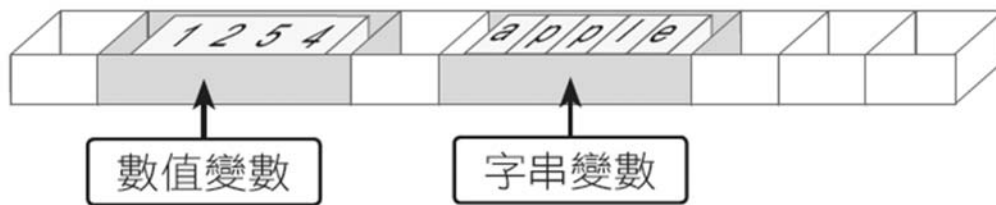
2018.1.30

主題

- 變數
 - 變數命名規則
 - 數值、布林與字串資料型態
- 輸出、輸入
 - print, input
- 資料型態轉換

變數(variable)

- 當變數被使用時，系統會配置一塊記憶體空間給此變數
 - 變數名稱 → 此塊記憶體的代號
 - 資料型態 → 此塊記憶體的大小



3

指定值

- Python 變數不需宣告就可使用，語法為：
 - 變數名稱 = 變數值
 - e.g., `score = 92` # Python的變數
 - e.g., `int score = 92` // C語言使用變數前要先宣告

4

● 指定值

- 使用變數時不必指定資料型態，Python 根據變數值設定資料型態
 - 例如上述變數 `score` 的資料型態是整數(integer)
 - e.g., `fruit = “香蕉”`
 - # `fruit`的資料型態為字串(string)

5

● 其他指定法

- 如果多個變數有相同變數值，可以一起指定變數值
 - 例如變數 `a`、`b`、`c` 的值皆為 20：
 - `a = b = c = 20`
- 也可以在同一列指定多個變數，變數之間以「`,`」分隔
 - 例如變數 `age` 的值為 18，`name` 的值為「魯夫」：
 - `age, name = 18, “魯夫”`

6

變數的命名

- 變數命名必須遵守的規則，否則在執行時會產生錯誤
- Python 變數的命名規則為：
 - 變數名稱的第一個字母必須是大小寫字母、_、中文
 - 1fruit → ☹️
 - 只能由大小寫字母、數字、_、中文組成變數名稱
 - fruit_price → 😊
 - fruit-price → ☹️

7

變數的命名

- 變數命名必須遵守的規則，否則在執行時會產生錯誤
- Python 變數的命名規則為：
 - 英文字母大小寫視為不同變數名稱
 - Fruit, fruit
 - 變數名稱不能與 Python 內建的保留字(reserved word)相同
 - 也稱為關鍵字(keyword)

8

● 保留字

Keywords in Python programming language

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

9

● 中文變數名稱

- 雖然 Python 3.x 的變數名稱支援中文，但不推

10

練習一下

- 這些變數名稱合法嗎？

7eleven



John&Mary



Angela baby



print



11

命名慣例規範 — PEP 8

- 避免使用的命名：
 - 除了計數器及迭代器之外，不要用單一字母為變數命名
 - 變數名稱前後不要加上兩個雙底線，如 `__fruit`
- 命名通則：
 - 一般變數用小寫英文單字，且單字間用底線符號相連
 - e.g., `stu_name`, `course_info`

12

資料型態

- 數值(numeric)
 - 整數(int)、浮點數 (float)
 - e.g., 23, 12.21, 15.0
- 布林 (Boolean)
 - True, False
 - 1 or 3 → True
 - 0 and 1 → False

13

資料型態

- 字串(string)
 - Python 字串資料型態(str) 是以一對雙引號「"」或單引號「'」包起來，例如：
 - str1 = “這是字串”
 - str2 = ‘這也是字串’
 - 如果字串要包含引號本身，可以這樣用：
 - str3 = ‘胡適說過“要怎麼收穫，先那麼栽”’

14

跳脫字元

- 字串含有特殊字元（如Tab、換行等），要使用跳脫字元
 - 以「\」開頭，後面接一定格式的字元來代表特定意義

跳脫字元	意義	跳脫字元	意義
\'	單引號「'」	\"	雙引號「"」
\\	反斜線「\」	\n	換行
\r	游標移到列首	\f	換頁
\b	後退鍵(BackSpace)	\v	垂直定位
\o	以八進為表示字元	\t	Tab鍵
\x	以十六進為表示字元	\a	響鈴

15

主題

- 變數
 - 變數命名規則
 - 數值、布林與字串資料型態
- 輸出、輸入
 - print, input
- 資料型態轉換

16

輸出到螢幕

- 利用 `print` 輸出到螢幕的語法有三種：
 - 簡易法
 - `print("TQC+ Python 3認證考試")`
 - 格式化輸出
 - `print("%3.2f" % num)`
 - 交給 Python 處理
 - `print("{} 好棒棒".format(name))`

17

輸出到螢幕 (I)

- `print` 命令能將指定項目輸出到螢幕，語法為：
 - `print(項目1[, 項目2, ..., sep = 分隔字元, end = 結束字元])`
 - 項目1, 項目2, ...：可以一次輸出多個項目，以逗號分隔
 - `sep`：分隔項目的字元，預設值為一個空白
 - `end`：輸出完後自動加入的字元，預設值為換行字元(`\n`)

```
3 print("多吃水果")
4 print(100, "多吃水果", 60, end="")
5 print(90, "多吃水果", 30, sep="-")
```

18

輸出到螢幕 (II)

- print 命令支援參數格式化功能，語法為：
 - 「%s」→ 字串
 - 「%d」→ 整數
 - 「%f」→ 浮點數
 - print(項目 % (參數列))

```
3 name = "喬巴"
4 score = 95
5
6 print("%s 的成績為 %d" %(name, score))
```

19

格式化輸出 (II)

- 格式化輸出可控制輸出位置，讓輸出的資料整齊排列，例如：
 - %5d, %5s：輸出 5 個字元
 - 小於 5 位數 → 在數字左方填入空白字元
 - 大於 5 位數 → 全部輸出
 - %8.3f：輸出 8 個字元（含小數點），小數固定輸出 3 位數
 - 若整數小於 5 位數 → 在數字左方填入空白字元
 - 若小數小於 3 位數 → 在數字右方填入「0」字元

20

● 格式化輸出 (II)

- 這個範例的輸出是甚麼？

```
2 pi = 3.14
3
4 print("圓周率 = %3.5f" % pi)
5 print("圓周率 = %1.1f" % pi)
6
7 num = 99
8
9 print("%3d" % num)
10 print("%-3d" % num)
```

21

● 輸出到螢幕 (III)

- 也可使用字串的 `format` 方法來做格式化，以一對大括號「`{}`」表示參數的位置，語法為：
 - `print(字串.format(參數列))`

```
2 name = "羅賓"
3 score = 99
4
5 print("{} 的成績為 {}".format(name, score))
```

22

輸入資料

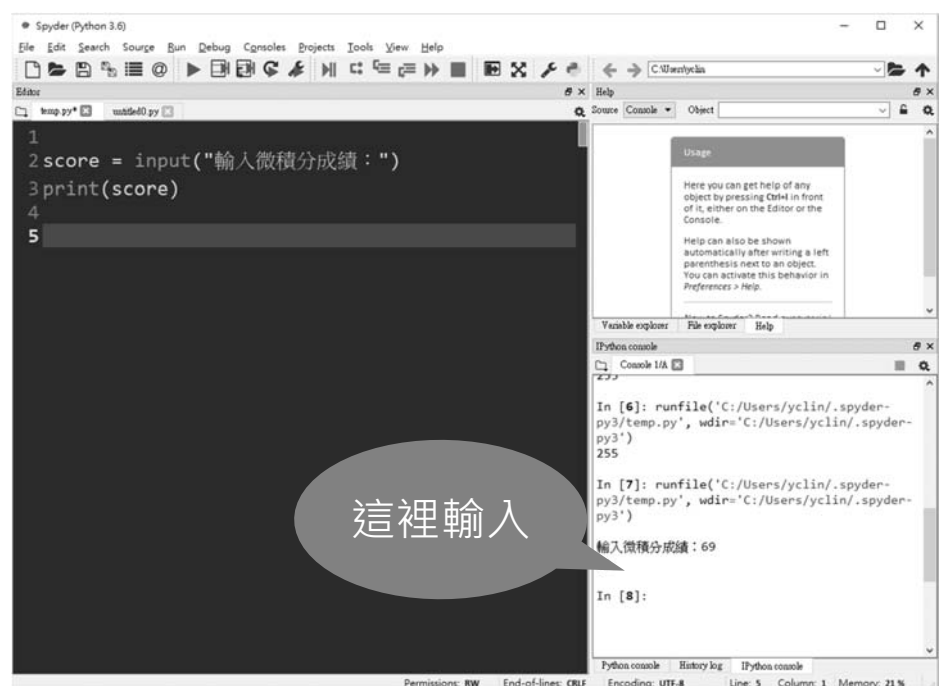
- print 命令是輸出資料，input 命令與 print 命令相反，是讓使用者由「標準輸入」裝置輸入資料。
 - 變數 = input([提示字串])

```
2 score = input("輸入微積分成績：")
3 print(score)
```

23

輸入資料

- 注意在 Spyder 編輯器的輸入方式



24

主題

- 變數
 - 變數命名規則
 - 數值、布林與字串資料型態
- 輸出、輸入
 - print, input
- 資料型態轉換

25

資料型態

- 試試看這段程式碼

```
2 score = input("輸入微積分成績：")
3 print(score)
4
5 print(score + 10)
```

26

● 如何修正？

```
2 score = int(input("輸入微積分成績："))
3 print(score)
4
5 print(score + 10)
```

27

● eval 命令

- 語法：eval (表示式)

```
2 score = eval(input("輸入微積分成績："))
3 print(score + 10)
4
5 x = 1
6 print(eval('x+1'))
```

28

查看資料型態

- `type` 命令會取得變數的資料型態，當不確定某些項目的資料型態，可用 `type` 命令確認，語法為：
 - `type(變數名稱)`

```
6 x = 100
7 y = 5.5
8 z = x - y
9
10 print(type(z))
```

29

資料型態轉換

- 如果系統無法自動進行資料型態轉換，就需以資料型態轉換命令強制轉換。
- Python 強制資料型態轉換命令有：
 - `int()` → 轉換為整數型態
 - `float()` → 轉換為浮點數型態
 - `bool()` → 轉換為布林型態
 - `str()` → 轉換為字串型態

30

● 試試看

```
2 # 如何修正下列錯誤
3
4 string = "67"
5 num = 23 + string
6
7 print("得到的數字為" + num)
```

31

● PY3-0002

一、圓形面積計算

2. 設計說明：

- (1) 請撰寫一程式，輸入一圓的半徑，並加以計算此圓之面積和周長，最後請印出此圓的半徑（Radius）、周長（Perimeter）和面積（Area）。

* 提示：輸出浮點數到小數點後第二位。

32

TO BE CONTINUED...