

Week 02 - 2023-0929

設置開發環境

- ☒ 安裝 Python
- ☒ 安裝 Spyder
- ☒ 確認可以執行

複習上週內容

變數的概念

左方為變數，= 的概念為賦予，右邊為給定的值

```
num = 100      # 變數 num 被賦予為 100 的數值
word = 'hello' # 變數 word 被賦予為 'hello' 的字串
area = 5 * 2    # 變數 area 被賦予為 5 * 2 的運算式
```

邏輯判斷練習

命題：我們今天身上只有五十元，如果超過五十元我們買不了，只要等於五十元以及小於五十元，我們可以買。

```
apple_price = 50

if apple_price > 50:
    print("Do not buy")
elif apple_price < 50:
    print("buy less")
elif apple_price == 50:
    print("buy equal")
```

Python 變數有趣補充 - 多重賦值 (multiple assignment)

變數做兩數內容交換時我可以用以下的方式進行實作，相較於其他語言還需要透過暫存 (temp) 的角色來實作，Python 這點可以充分發會語言特性。

```
a = 100
b = 200

a, b = b, a

# 其他語言方法
a = 100
```

```
b = 200
temp = a    # 需要有個 temp 的角色來幫助
a = b
b = temp
```

資料型態

- **int** (整數)：用於表示整數數值，例如：**1**、**-5**、**100**。
- **str** (字串)：用於表示文字或字元串，必須放在單引號 (' ') 或雙引號 (" ") 之間，例如：**"Hello, World!"**。
- **float** (浮點數)：用於表示帶有小數點的數值，例如：**3.14**、**-0.5**、**2.0**。
- **bool** (布林值)：用於表示布林邏輯值，僅有兩個可能的值：**True** (真) 和 **False** (假)，用於條件判斷和邏輯運算。
- **list** (串列)：用於表示有序的、可變動的數據集合，可以包含不同資料型態的元素，並使用中括號 [] 來表示，例如：**[1, 2, 3]**。
- **dict** (字典)：用於表示鍵 - 值對 (key-value pair) 的數據集合，每個鍵對應一個值，使用大括號 { } 來表示，例如：**{"name": "John", "age": 30}**。
- **tuple** (元組)：用於表示有序的、不可變動的數據集合，類似於串列，但使用小括號 () 來表示，例如：**(1, 2, 3)**。

這些不同的資料型態在Python中用於儲存和處理不同類型的資訊，了解它們對於寫程式非常重要，因為您需要根據您的需求選擇適當的資料型態來儲存和操作資料。

運算

在 Python 中，可以使用以下常見運算子進行基本數值運算：

- **+**：加法運算，用於將兩個數字相加。
- **-**：減法運算，用於將一個數字減去另一個數字。
- *****：乘法運算，用於將兩個數字相乘。
- **/**：除法運算，用於將一個數字除以另一個數字，結果是浮點數。
- **//**：整數除法運算，用於將一個數字除以另一個數字，結果是整數。
- **%**：取餘數運算，用於獲取一個數字除以另一個數字後的餘數。
- ******：指數運算，用於將一個數字的幕次方。

```
a = 5
b = 2

c = a + b
print(f'加法結果：{c}') # 7

d = a - b
print(f'減法結果：{d}') # 3

e = a * b
print(f'乘法結果：{e}') # 10

f = a / b
print(f'除法結果：{f}') # 2.5
```

```
g = a // b
print(f'整數除法結果：{g}') # 2

h = a % b
print(f'取餘數結果：{h}') # 1

i = a ** b
print(f'指數運算結果：{i}') # 25
```

邏輯判斷

我們有了基礎的變數概念，當我們今天要做比較的時候，我們會用到邏輯判斷子，有以下常用的內容：

- `==`: 等於
- `!=`: 不等於
- `>`: 大於
- `<`: 小於
- `>=`: 大於等於
- `<=`: 小於等於

if ... else...

```
if 條件:
    # 條件成立時執行
else:
    # 條件不成立時執行
```

elif

```
if 條件:
    # 條件成立時執行
elif 條件:
    # 條件成立時執行
else:
    # 條件不成立時執行
```

處理小數點

format

我們如果要透過四捨五入的方式取到小數點後第二位我們可以透過 `format` 的方式 `print` 出來以取得我們想要的數值

```
probability = 0.123456789

print(f'{probability}')          # 0.123456789
print(f'{probability:.2f}')      # 0.12
print('%.2f' % probability)      # 0.12
```

round()

我們如果要透過四捨五入的方式取到小數點後第二位我們可以透過 `round()` 的方式來取得我們想要的數值。

```
probability = 0.123456789
print(round(probability, 2))    # 0.12
```

`round(數值, 小數點後第幾位)`

透過型別轉換

另外我們也可以透過先前學習到的型別轉換先轉換成 `str`，再透過對 `str` 的處理轉換成 `float` 因而得到我們想要的數值。

```
probability = 0.123456789
probability_str = str(probability)
probability_str = probability_str[:4]
probability = float(probability_str)
print(probability)            # 0.12
```

透過運算的方式截斷

將一個浮點數 `probability` 乘以 100，然後再將結果轉換為整數。接著，它會再將該整數轉換回浮點數，並保留兩位小數。這樣的處理方式通常稱為「四捨五入到兩位小數」。

這樣的處理方式可能有助於限制浮點數的小數部分，以確保它只保留兩位小數，而不進行四捨五入

```
probability = 0.123456789
print(int(probability * 100) / 100)
```

程式輸入

我們今天程式的內容一旦是需要使用者提供的，我們便要將內容回傳給程式，因此透過 `input()` 的互動方式來讓程式作動，以下方案例而言，程式想要獲取我們的名字資訊，因此透過 `input()` 來取得使用者資

訊，並且透過「儲存變數」的概念將使用者輸入的內容保留，最後透過 `print()` 的方式顯示在畫面上。

```
name = input('請輸入你的名字：')

print(f'你好，{name}')
```

指定輸入的類型

一般而言，如果我們單用 `input()` 做輸入的動作，我們得到的任何內容都是 `str` 無論是輸入 `123`, `abc`, `0.123456789` 對 Python 而言都是字串 (`str`) 的形式，因此我們如果要變成其他類型我們就要在 `input()` 前加上我們想要的型別名稱，通常我們會用到 `int()`, `float()`, `bool()` 的類型，甚至嚴謹一點加上 `str()` 也是沒問題的哦！

```
int_num = int(input())      # int
float_num = float(input())  # float
str_num = str(input())      # str
bool_num = bool(input())    # True or False
```

同一行需要多個輸入

今天，如果我們需要同時獲取多個輸入值，我們可以使用 `map()` 函數，這允許我們以一行的方式獲取多個輸入值。為了在這些輸入值之間進行區分，我們可以使用 `split()` 函數的幫助。`split()` 預設使用空白作為分隔符，將輸入字符串分割成多個部分。以下是一個示例，我們使用 `map()` 來獲取使用者輸入的兩個數字，然後使用 `int()` 將這些輸入轉換為整數類型，最後使用 `print()` 將計算結果顯示在螢幕上：

```
a, b = map(int, input('請輸入兩個數字：').split())
c = a + b
print(f'計算結果：{c}')
```

`map()` 用法

`map()` 是一個內建函式，可以將一個函式套用到一個或多個可迭代物件上，並回傳一個 `map` 物件，`map()` 的用法如下：

```
map(function, iterable, ...)
```

- `function`: 變數類別
- `iterable`: 輸入內容

split() 用法

`split()` 是一個內建函式，可以將一個字串分割成多個部分，並回傳一個 `list` 物件，`split()` 的用法如下：

```
split(sep=None, maxsplit=-1)
```

- `sep`: 以空格做分割
- `maxsplit`: 想要分割的數量

```
a = 'Hello World'
print(a.split())          # ['Hello', 'World']
print(a.split('o'))       # ['Hell', ' W', 'rld']
print(a.split('o', 1))    # ['Hell', ' World']

b = 'Hello, World'
print(b.split())          # ['Hello,', 'World']
print(b.split(','))       # ['Hello', ' World']
print(b.split(',', 1))    # ['Hello', ' World']
```

 [第一週課程](#) |  [第三週課程](#)

License

All of these teaching materials are owned by [Hugo ChunHo Lin](#).

These materials are intended for tutoring purposes. They are open-source to foster a more vibrant Python learning community. We warmly welcome fellow enthusiasts interested in Python to use them. If you use a substantial portion of the source code, please include a link back to this repository.