

# 基礎程式設計 - Python

Po-Chieh Yu  
[pcyu@saturn.yzu.edu.tw](mailto:pcyu@saturn.yzu.edu.tw)

# Review

- 資料類型
- 變數命名，註解
- 運算子
- 數學函式
- 讀取螢幕輸入的資訊

# Review

- 請撰寫一個程式，讓使用者輸入民國年份並轉換成西元年份印出。

- 請撰寫一個程式，讓使用者輸入民國年份並轉換成西元年份印出。
- `republic_input = input("Please enter in Republic Era:")`
- `republic_input = int(republic_input)`
- `ce_output = republic_input + 1911`
- `print(ce_output)`

- 請撰寫一個程式，讓使用者輸入一個數字並計算其log值且印出。(提示：可用 `math.log10` 函式)
- 請撰寫一個程式，讓使用者輸入兩個數字x與y，並計算 $x^y$ 值且印出。

- 請撰寫一個程式，讓使用者輸入一個數字並計算其log值且印出。(提示：可用 `math.log10` 函式)
  - `import math`
  - `num = input("Please enter your number:")`
  - `num = float(num)`
  - `num_log = math.log10(num)`
  - `print(num_log)`

- 請撰寫一個程式，讓使用者輸入兩個數字 $x$ 與 $y$ ，並計算 $xy$ 值且印出。

- 請撰寫一個程式，讓使用者輸入兩個數字 $x$ 與 $y$ ，並計算 $x^y$ 值且印出。
  - `import math`
  - `num1 = input("Please enter your number x:")`
  - `num2 = input("Please enter your number y:")`
  - `num1 = float(num1)`
  - `num2 = float(num2)`
  - `answer = num1**num2`
  - `print(answer)`



# 複習題

- 請撰寫一個程式，讓使用者輸入一個角度並計算其sin值且輸出。(提示: 利用數學函數sin()，sin()裡需為弧度，角度轉成弧度公式為  $\text{radians} = \text{degrees} * \text{PI} / 180$ )

# 複習題

- 請撰寫一個程式，讓使用者輸入一個角度並計算其sin值且輸出。  
(提示: 利用數學函數sin()，sin()裡需為弧度，角度轉成弧度公式為  $\text{radians} = \text{degrees} * \text{PI} / 180$ )
- `import math`
- `input_deg = input("Please enter in degree:")`
- `input_deg = float(input_deg)` #先把輸入的內容轉成浮點數
- `input_radian = input_deg*(math.pi)/180` #把input\_deg轉成弧度
- `output_sin = math.sin(input_radian)` #再使用sin函式
- `print(output_sin)`

# 字串 (string)

- 處理的頻率比數字還高
- 可以使用單引號或雙引號來定義字串
- `A="This is a string."`
- 空字串: `A=""`
- 可以將其他資料形態轉成字串: `str()`
- Python字串是不可變的

# 字串運算

- 字串結合: +
- A="Hello!"
- B="Python!"
- C=A+B
- print(C)

# 字串運算

- 字串複製: \*
- A="HaHa"\*3
- print(A)

# 字串運算

- 記得變數互換嗎? `x,y=y,x`
- 試試看字串是否成立?
- `x='This is an apple'`
- `y='Hello! Python!'`
- `print(x)`
- `print(y)`
- `x,y = y,x`
- `print(x)`
- `print(y)`

# 字串運算

- 擷取字元: []
- 利用字串名稱後面的方括號指定位移值
- 第一個字元由0開始，代表最左邊的字元
- 也可以用-1開始，代表最右邊的字元
- A='This is my first program!'
- print(A[0])

# Python字串是不可變的

- `name="PC Yu"`
- `name[0]="A"`
- 會有錯誤，要利用字串函式



# 字串運算

- `my_string='example'`
- 利用[]擷取一段字串
- 擷取整個序列: `my_string[:]`
- 指定開始的位移值到最後: `[start:]` (ex: `my_string[10:]`)
- 從開始到指定的位移值減1: `[:end]` (ex: `my_string[:20]`)
- 指定開始的位移值到指定的位移值減1: `[start:end]`  
(ex: `my_string[3:10]`)
- 指定開始的位移值到指定的位移值減1，並且間隔n個字元:  
`[start:end:n]` (ex: `my_string[3:6:2]`)

# 練習一下

- `my_string="The formation and evolution of supermassive black holes"`

# 字串函式

- my\_string="The formation and evolution of supermassive black holes"
- **len()** : 取得字串的長度
- string\_length = **len**(my\_string)
- print(string\_length)
- **split()**: 分割字串 (字串專用函式)
- 練習空白字元分割
- split\_string = my\_string.**split**(' ')
- print(split\_string)

- **join(): 結合字串 (字串專用函式)**
- 練習將分割後的字串用 '-' 結合
- join\_string = '-'.**join**(split\_string)
- print(join\_string)
- **replace(): 取代字串 (字串專用函式)**
- 練習用 '&' 取代 '-'
- replace\_string = join\_string.**replace**('-', '&')
- print(replace\_string)

# Play with your string!

- `my_string="The formation and evolution of supermassive black holes"`
- 先得知字串的長度: `len(my_string)`
- 傳回前6個字元: `my_string[0:6]`
- 他的開頭是不是"The": `my_string.startswith('The')`
- 結尾是不是"The": `my_string.endswith('The')`
- 找出"and"第一次出現的位移值: `my_string.find('and')`
- 最後一次出現的位移值: `my_string.rfind('and')`
- "The"出現幾次? "the"出現幾次? `my_string.count('the')`

# 大小寫轉換

- `my_string="the formation and evolution of supermassive black holes"`
- 把第一個字母改成大寫:
- `my_string = my_string.title()`
- `print(my_string)`
- 把所有字元改成大寫: `my_string.upper()`
- 把所有字元改成小寫: `my_string.lower()`
- 大小寫對調: `my_string.swapcase()`

# Play with your string!

- 先得知字串的長度
- 傳回前6個字元
- 他的開頭是不是"The": `my_string.startswith()`
- 結尾是不是"The": `my_string.endswith()`
- 找出"and"第一次出現的位移值: `my_string.find()`
- 最後一次出現的位移值: `my_string.rfind()`
- "The"出現幾次? "the"出現幾次? `my_string.count()`

# Play with your string!

- `my_string="The formation and evolution of supermassive black holes"`
- 先得知字串的長度: `len(my_string)`
- 傳回前6個字元: `my_string[0:6]`
- 他的開頭是不是"The": `my_string.startswith('The')`
- 結尾是不是"The": `my_string.endswith('The')`
- 找出"and"第一次出現的位移值: `my_string.find('and')`
- 最後一次出現的位移值: `my_string.rfind('and')`
- "The"出現幾次? "the"出現幾次? `my_string.count('the')`



# 練習題

- 將字串 `my_string='I like python!'` (1)都改成大寫並存到另一個變數輸出;(2)將新變數的字串以空白字元分割並輸出;(3)將分割的字串用'%'連接起來並輸出;(4)將連接後的字串，用'\$'取代'%'並輸出

# 練習題

- 將字串 `my_string='I like python!'` (1)都改成大寫並存到另一個變數輸出;(2)將新變數的字串以空白字元分割並輸出; (3)將分割的字串用'%'連接起來並輸出; (4)將連接後的字串，用'\$'取代 '%' 並輸出
- `u_string = my_string.upper()`
- `print(u_string)`
- `u_string = u_string.split(' ')`
- `print(u_string)`
- `u_string = '%'.join(u_string)`
- `print(u_string)`
- `u_string = u_string.replace('%','$')`
- `print(u_string)`