

# 程式設計概論 Programming 101

## 一 程式變數與第一個python內建函式

---

授課老師：邱淑怡

DATE: 9/26/2023

# 大綱

---

1. Python 程式碼撰寫風格
2. 資料型別
3. 程式的變數
4. 變數如何運用

# Python 程式碼撰寫風格

---

程式(program)：由一行一行的敘述(statement)所組成

敘述：由關鍵字、特殊字元或識別字所組成

- 關鍵字(keyword): Python所定義特定意義或用途，ex: if, for,...
- 特殊字元(special character): 小括號、單(雙)引號、井字(#)符號
- 識別字(identifier): 程式人員自行定義的新字做為變數、函式的名稱
  - 字母大小寫意義不一樣
  - 利用空白 讓程式容易閱讀；但須避免無意義的空白
  - 變數名稱建議是有意義的名稱

敘述可包含函式(function)、流程控制(flow control)、類別(class)，可執行的單元

# Python 程式碼撰寫風格(cont.)

---

## 縮排

- 每個縮排層級使用 4 個空白，或用[tab]鍵，**但不能混合空白和[tab]鍵**
- Python 使用縮排來劃分程式的執行區塊，**程式不能隨意縮排**

註釋(comment): # 標示單行註釋；''' 或'''' 標示多行註釋

- 建議程式中多些註釋說明

## 建議**一行一個敘述**

- `print('Python'); print("Python code")`



不建議使用

# Python變數型別(data type)

---

## 數值型別(numeric type)

- int, float, bool(布林值)

## 文字序列型別(text sequence type)

- string

## 序列型別(sequence type)

- list, tuple

## 集合型別(set type)

- set

## 對映型別(mapping type)

- dictionary

# 基本資料型態(data type)

## □ 數值(numeric)

- int: 表示整數(integer)
  - print(3+6)
- float: 表示浮點數
  - 如: 3.14159265359
- bool: 表示布林(Boolean): True (T) / False (F)
  - 1>2

不同的資料型態  
不同的操作及使用方式

注意大小寫

## □ str: 字串(string)由一連串字元組成

- 單引號、雙引號、三個單引號、三個雙引號
- 單引號和雙引號不能混合使用，如何分別使用?

# 變數 (variable)命名

---

- 變數參照可改變的值，這個值儲存在記憶體
- 變數命名
  1. 以英文字母或底線(\_)開頭，之後可以接字母、數字或底線
  2. Python 的關鍵字(keyword)及內建函數(built-in function)不能當變數名稱
  3. **字母大小寫有差別(case-sensitive)**
  4. 變數名稱可以為中文 (Python 3)，不建議使用
  5. 建議變數名稱需有意義的名稱

變數的第一個字元  
不能是數字

# 變數 (variable)

---

變數: 一般而言，會利用變數來儲存數值、字串或其他資料型態的資料

a=10

b=3.56

c="python for everyone"

「=」就是指派(assign)

- 將等號右邊的數值、字串或其他資料給等號左邊的變數
- 右邊必須是已確定的資料

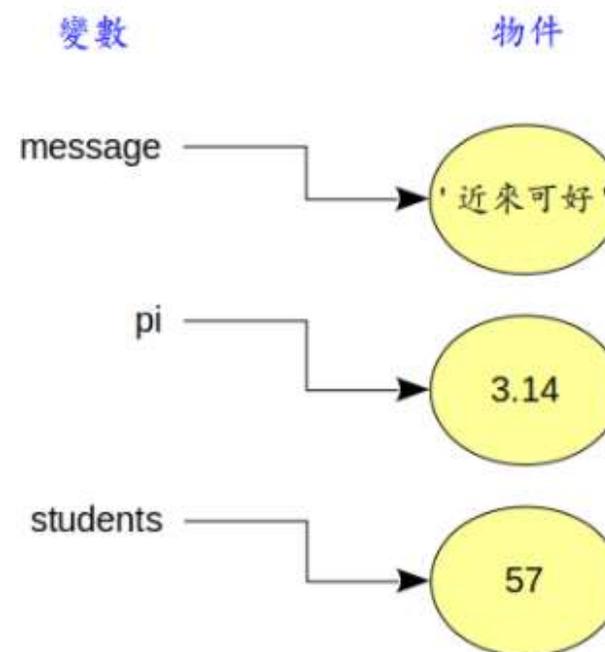
# 變數(variable)

變數：代表一個值的名稱

指派指令 (Assignment instruction) 可以建立新的變數，並設定其值

例子

- `message = '近來可好'`
- `pi = 3.14`
- `students = 57`

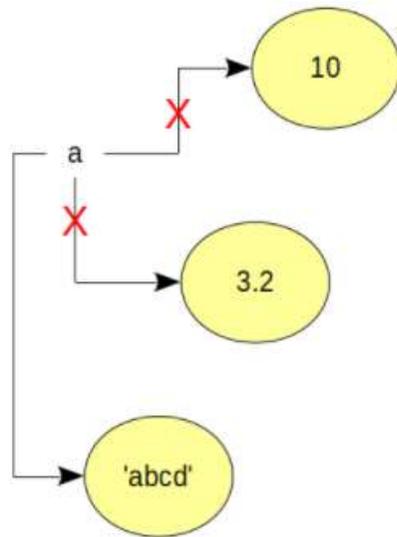


# Python變數特性

- 利用指派指令直接創造新變數
- 動態資料型別(dynamic typing)，只要指派指令表示式的資料型態改變，變數的資料型態就改變了

- Ex:

- a=10
- a=3.2
- a='abcd'
- print(a)



## Python特殊用法: 指派鏈 (chained assignments)

- X=Y=Z=100
- a, b, c=100, 3.14159, "Hello"
- print(X,Y,Z,a,b,c)

# Python錯誤變數的示範

---

7eleven='7-11'

Q&A='Q&A'

Q A =10

if='if'

# Python函式(function)

---

□ Function(函式): 提供很多有用的功能

◆ Python內建函式

◆ Python模組下的函式：內建模組或外部模組

◆ 自行定義的函式

□ print()

print(項目1[,項目2,..., sep=分隔字元, end=結束字元])

\* 預設值: sep=' ',end='\n'

◦ print(100,"Python",60,sep='###',end='.')

# print() function

---

1. 格式化(%)
2. 格式化(format)
3. f-string

# print() 格式化(%)

語法:

print(項目 % (參數列))

\* %s:字串, %d:整數, %f:浮點數, 預設靠右對齊

\* %3d 代表3位數字, %3s 代表3個文字字串

\* %-3d 或 %-3s 代表文字數字靠左對齊

\* %.2f 代表取2位小數, 四捨五入

\* %6.2f 代表6位數字(3位整數,1位小數點,2位小數)

參數	意義
%d	以整數資料型態輸出
%s	以字串資料型態輸出
%f	以浮點數資料型態輸出
%%	在字串中顯示 %

# print() 格式化(format)

---

語法:

```
print(字串.format(參數列))
```

- \* {}：參數位置，多個{}預設要以後方參數順序為準
- \* {i}：指定插入參數的索引值，如{0}代表第1個
- \* {i:格式化指定}：參考%的格式化指定

```
print("考{1}分成績是{0:8.3 }".format("John",78))
```

# print(): f-string

---

```
text = 'python'
```

```
print(f'Hello, {text}')
```

```
x = 10
```

```
y = 27
```

```
print(f'x + y = {x + y}')
```