

# 模組(Module)使用

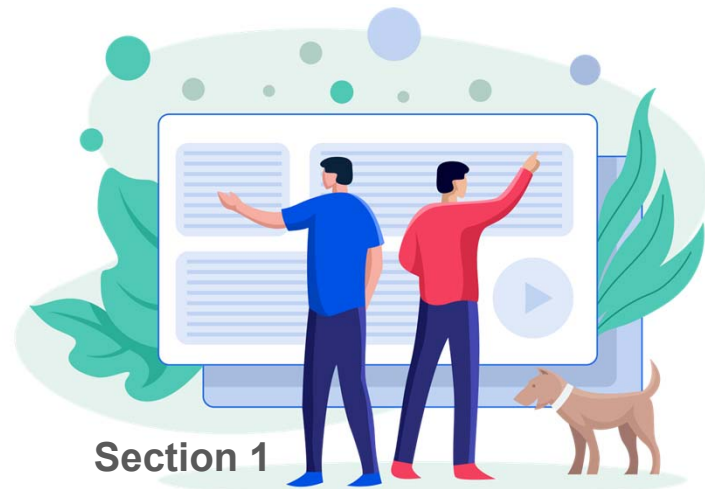
# 學習目標

- 介紹標準程式庫(Standard Libraries)
- sys 標準程式庫模組
- os 標準程式庫模組



# 常用標準程式庫(Standard Libraries)

## ■ 介紹標準程式庫



# 模組位置

- 當匯入模組時，Python 直譯器會依以下順序搜尋模組
  - 目前目錄
  - “PYTHONPATH” Shell 變數設定的路徑
  - 預設路徑
- 模組搜尋路徑存放於系統 “sys” 模組內的 “path” 變數
  - ```
>>> import sys  
>>> sys.path
```

# Python標準程式庫

- Python 的標準程式庫相當廣泛，提供了各式各樣的工具
- 標準程式庫包含用 C 撰寫的內建模組，可以存取各種系統功能，例如檔案 I/O。也有用 Python 撰寫的模組，提供程式所面臨的各種狀況的標準解決方案
- Python 核心(Core) 部分有很多內建的資料型態，例如數值(Numbers) 與串列(Lists)，可以直接使用而不需要匯入模組
- 這些資料型態都定義在標準程式庫，還有很多內建元件都定義在標準程式庫內

# 內建常數(Constants) -1

## ■標準程式庫內有定義內建常數(Constants)

▣True: 布林真值 · False: 布林假值

■a = True

b = False

if a is True:

    print('a is true.')

else:

    print('a is false')

if b:

    print('b is true.')

else:

    print('b is false.')

# 內建常數(Constants) -2

■ None: 內建常數，用來表達空值，類似其他語言的 null 關鍵字

## ■ 範例

```
□ lst = ['apple', None, 77, '', 123]
  for item in lst:
    if item is not None:
      print(item)
```

## ■ 結果

```
□ apple
  77

  123
```

# 內建函數(Functions)

| Built-in Functions |               |             |              |              |            |                |
|--------------------|---------------|-------------|--------------|--------------|------------|----------------|
| abs()              | chr()         | exec()      | hex()        | map()        | print()    | staticmethod() |
| all()              | classmethod() | filter()    | id()         | max()        | property() | str()          |
| any()              | compile()     | float()     | input()      | memoryview() | range()    | sum()          |
| ascii()            | complex()     | format()    | int()        | min()        | repr()     | super()        |
| bin()              | delattr()     | frozenset() | isinstance() | next()       | reversed() | tuple()        |
| bool()             | dict()        | getattr()   | issubclass() | object()     | round()    | type()         |
| breakpoint()       | dir()         | globals()   | iter()       | oct()        | set()      | vars()         |
| bytearray()        | divmod()      | hasattr()   | len()        | open()       | setattr()  | zip()          |
| bytes()            | enumerate()   | hash()      | list()       | ord()        | slice()    | __import__()   |
| callable()         | eval()        | help()      | locals()     | pow()        | sorted()   |                |



# sys 模組 -1

- sys 模組允許接收執行時環境資訊，也可以修改環境資訊
- 必須匯入 sys 模組
  - `import sys`
- sys 模組一個最常用的功能是處理程式收到的命令列引數(Command-line arguments)
  - 使用 `sys.argv` 串列(List) 來完成
  - `sys.argv` 串列的第一個引數一定是模組/程式名稱，其餘項目分別表示收到的引數值

## sys 模組 -2

### ■ 範例 (argv.py)

### ■ import sys

# C-style

```
for i in range(len(sys.argv)):
```

```
    print("sys.argv[" + str(i) + "] : " + sys.argv[i])
```

```
print()
```

# Python-style

```
for arg in sys.argv:
```

```
    print(arg)
```

# sys 模組 -3

## ■ `sys.exit([status])` 方法(Method) 可以正常的(Gracefully) 結束程式

- 會觸發 `SystemExit` 例外(Exception)，如果沒有攔截，就會結束程式
- 可以傳送一個引數當作結束碼(Exit code)，0 代表成功，非 0 代表有錯誤

## ■ `sys.platform`

- 表示所在的平台識別字，例如 'linux'、'win32'、'darwin'

## ■ `sys.version`

- 表示 Python 直譯器的版本，以及編譯版本(Build number) 與使用的編譯器(Compiler used) 等相關資訊

# os 模組 -1

- os 模組提供一個與作業系統相關功能的跨平台介面(Portable way)
- os 模組所有的函數如果發生錯誤，會觸發 OSError 的例外(Exception)
- 必須匯入 os 模組
  - `import os`
- `os.rename(src, dst)`
  - 將 src 的檔案或目錄更名為 dst
- `os.remove(filename)`
  - 刪除 filename 檔案

# os 模組 -2

- os 模組也提供了一系列目錄相關的功能

- `os.listdir(dirname)`

  - ▣ 列出 `dirname` 目錄內的檔案與目錄

- `os.getcwd()`

  - ▣ 傳回目前所在的目錄名稱(Current Work Directory)

- `os.chdir(dirname)`

  - ▣ 切換目錄(Change Directory) 到 `dirname`

- `os.rmdir(dirname)`

  - ▣ 刪除 `dirname` 的空目錄

# 常用標準程式庫(Demo)

## ■ 如何使用常用標準程式庫



# 本章重點精華回顧

## ■ 常用標準程式庫(Standard Libraries)



# Lab: 模組(Module)使用

## ■ Lab01: 使用常用標準程式庫(Standard Libraries)



# Lab01: 使用常用標準程式庫(Standard Libraries)

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “argv.py” 程式，了解sys模組的應用
3. 開啟命令列(cmd)，切換到相關目錄，做以下測試  
python argv.py aaa bbb ccc
4. 關閉argv.py程式視窗
5. 使用 “File/Open...” 開啟 “listdir.py” 程式，了解sys與os模組的應用
6. 開啟命令列(cmd)，切換到相關目錄，做以下測試  
python listdir ..
7. 關閉listdir.py程式視窗