

# 模組(Module)設計與使用

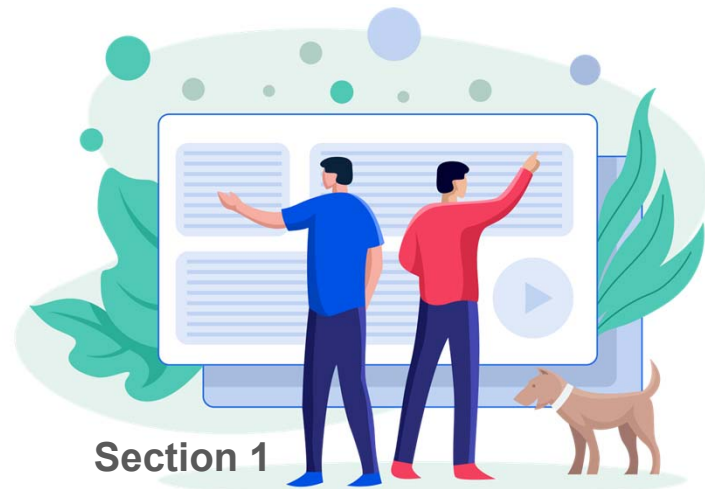
# 學習目標

- 定義模組(Module)
- 使用模組(Module)
- `__name__` 內建變數的用法



# 模組(Module)的設計與使用

- 瞭解模組的設計
- 瞭解模組的用法



# 模組(Modules)

- 模組(Modules) 可以組織 Python 程式碼，將相關程式碼集合在一個模組可以讓程式更容易了解與使用
- 模組視為一個 Python 物件，可以給模組一個名稱，讓程式來使用該模組的內容
- 簡言之，模組是一個檔案，內含 Python 程式碼。模組可以定義函數(Functions)、類別(Classes) 以及變數，也可以包含執行碼(Runnable codes)

# import 敘述

- Python 程式可以匯入另一支 Python 程式、匯入 Python 標準程式庫模組、或是匯入其他 Python 套件提供的模組
- 語法
  - `import module1[, module2[,... moduleN]`
- 當 Python 直譯器遇到 import 敘述時，它會依 Python 模組搜尋路徑找到模組，然後匯入
- 每個模組只會匯入一次，不管它在程式內被 import 了幾次。如此可以避免不必要的錯誤發生

# 定義模組(Modules)

## ■ 範例

■ # bmi module

```
def BMI(w,h):
```

```
    """
```

Calculate BMI

Arguments:

w:weight, h:height

```
    """
```

```
    return w/(h/100)**2
```

# 定義與匯入模組

- Python 模組名稱與檔名相同，例如模組名稱為 “support” ，則檔名為 “support.py”
- support.py
  - `def show_name(arg):`  
    `print ('Hello :', arg)`
- `# Import module support`
  - `import support`  
    `support.show_name('Rose')`
- 可以使用 “as” 關鍵字來簡化匯入的模組名稱
  - `import support as sp`

# from ... import 敘述

- Python 的 from 敘述可以匯入模組內特定的屬性到目前程式的命名空間 (Namespace)

- 語法

  - `from modname import name1[, name2[, ... nameN]]`

- `# Import show_name in support module`

  - `from support import show_name`  
`show_name('Rose')`

- `from ... import *` 敘述

  - 將模組內所有的屬性直接匯入到目前程式的命名空間。這是一個方便的措施，但是必須小心使用，以避免同名的屬性被覆蓋掉



# dir() 內建函數

■ dir()內建函數(Function) 傳回模組內定義的所有名稱，以排序過的串列(List) 表達

■ >>> import math

>>> dir(math)

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos',  
'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',  
'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial',  
'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose',  
'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p',  
'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh',  
'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

# 特殊Python系統變數

■ `__name__` 代表模組名稱，內容有以下兩種可能

- `__main__`：最上層腳本環境

- 一般是代表直接執行該腳本(script)

- 模組名稱

- 代表該腳本是被匯入的(import)

■ 當直接執行 Python 程式時，`__name__` 會被自動設定為 `'__main__'`，如此程式可以依此判斷是直接執行還是被其他程式匯入

- ```
if __name__ == '__main__':  
    # execute only if run as a script  
    main()
```

■ `__file__` 代表模組的檔名

# \_\_name\_\_ 系統變數 -1

■ bmi.py

■ # bmi module

```
def BMI(w,h):  
    return w/(h/100)**2
```

# MAIN

```
if __name__ == '__main__':  
    height=float(input('Height(cm): '))  
    weight=float(input('Weight(kg): '))  
    print('BMI is ',BMI(weight,height))
```

## \_\_name\_\_ 系統變數 -2

- Caller

- `from bmi import BMI`

```
# call defined function
```

```
result = BMI(67,175)
```

```
print('175cm,67kg,BMI={:.4f}'.format(result))
```

# 模組的設計與使用(Demo)

- 如何設計模組
- 如何使用模組



# 本章重點精華回顧

- 模組(Module) 的設計與使用
- `__name__` 內建變數的用法



# Lab: 模組(Module)設計與使用

## ■ Lab01: 設計與使用模組(Module)

# Lab01: 設計與使用模組(Module)

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “support.py” 程式，了解模組的設計格式
3. 使用 “File/Open...” 開啟 “ex16\_1.py” 程式，了解模組的使用方法
4. 關閉support.py與ex16\_1.py程式視窗
5. 使用 “File/Open...” 開啟 “bmi.py” 程式，了解模組的設計格式
6. 使用 “File/Open...” 開啟 “ex16\_2.py” 程式，了解模組的使用方法
7. 關閉bmi.py與ex16\_2.py程式視窗