



巨匠直播教學

APCS Python語法基礎班

模組與套件

www.pcschoolonline.com.tw

本堂教學重點

1. 引用模組
2. 套件宣告
3. 常用標準庫模組

課程內容

1. 模組與套件

1-1. 模組引用

1-2. 常用模組

1-3. 自訂模組

1-4. 套件宣告與使用

1-5. 匯入模組注意事項

2. 常用標準庫模組

2-1. sys模組

2-2. os模組

2-3. shutil模組

2-4. date/time模組

課程內容

1. 模組與套件

- 1-1. 模組引用
- 1-2. 常用模組
- 1-3. 自訂模組
- 1-4. 套件宣告與使用
- 1-5. 匯入模組注意事項

2. 常用標準庫模組

- 2-1. sys模組
- 2-2. os模組
- 2-3. shutil模組
- 2-4. date/time模組

模組 Module

◆ 模組 Module

- ◆ 重複利用現成的模組來擴充功能，減少不必要的重複開發
 - 標準程式庫模組
 - 第三方開發的模組
 - 過去已開發的模組
- ◆ 將專案中不同功能的實作集中放在獨立的程式中
 - 便於程式的管理維護及重複使用

模組 Module

◆ 常用模組

◆ 標準程式庫模組

◆ 第三方函式庫

- NumPy 和 SciPy：統計及科學運算
- Matplotlib：Python 的繪圖函式庫
- Pandas：高階數據操作函式庫
- Flask 和 Django：網站架設網頁框架
- Pygame：遊戲開發

Python 模組使用

◆ Python 模組使用

- ◆ 每個Python原始碼檔案都可以是一個模組
- ◆ 使用非內建的類別或函式，需宣告匯入模組後才可使用
- ◆ NameSpace(命名空間)：讓函式名稱不會產生衝突

匯入模組

◆ 匯入整個模組

◆ import 模組

◆ import 模組 as 別名名稱

- 將匯入模組取名

◆ 使用匯入模組中函式：

模組名稱.函式名稱 / 模組名稱.變數名稱

別名名稱.函式名稱 / 別名名稱.變數名稱

匯入模組

◆ 匯入模組部分內容

◆ from 模組 import 模組內函式或變數

- 匯入模組中某一個函式或變數

◆ 使用模組中匯入的內容

- 直接使用函式名稱或常數呼叫，不必在加上模組名稱

函式名稱

變數名稱

匯入常用模組

◆ 以math模組為例

◆ math是標準程式庫提供，不需安裝

◆ 匯入模組

- import math

◆ 驗證模組包含的內容

- dir(模組名稱)

◆ 查詢函式使用方法

- help(模組名稱.函式名稱)

math

常數: pi, e, nan, inf

函式:

ceil(x)

floor(x)

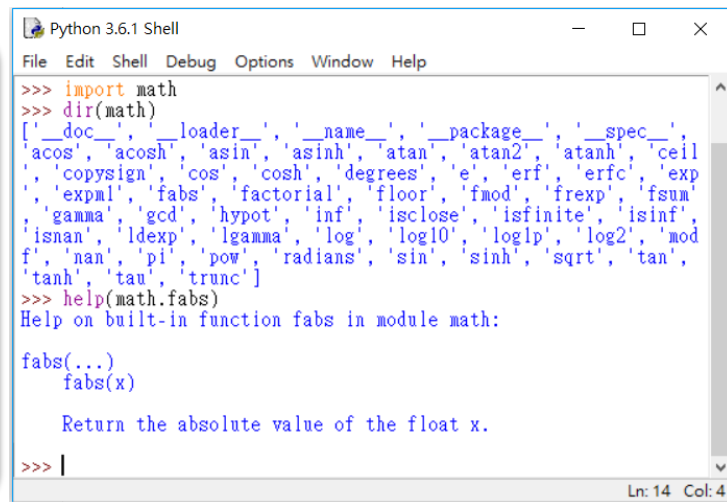
fabs(x)

fmod(x, y)

factorial(x)

gcd(a, b)

.....



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
>>> 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', 'fsun',
'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf',
'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'loglp', 'log2', 'mod',
'nan', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'tau', 'trunc']
>>> help(math.fabs)
Help on built-in function fabs in module math:

fabs(...)
    fabs(x)

    Return the absolute value of the float x.

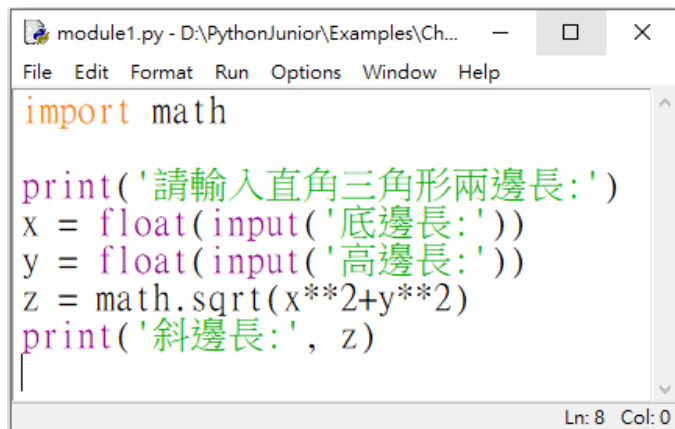
>>> |
```

常用模組：math模組

◆ math 模組

- ◆ 提供進階數學運算的函式
- ◆ 常數: pi, e, nan, inf
- ◆ 函式:
 - `ceil(x)`：天花板，傳回比X大的最小整數
 - `floor(x)`：地板，傳回比X小的最大整數
 - `fabs(x)`：絕對值
 - `gcd(a, b)`：a與b的最大公因數
 - `sqrt(x)`：平方根
 - `sin(x)`、`cos(x)`、`tan(x)`：三角函數

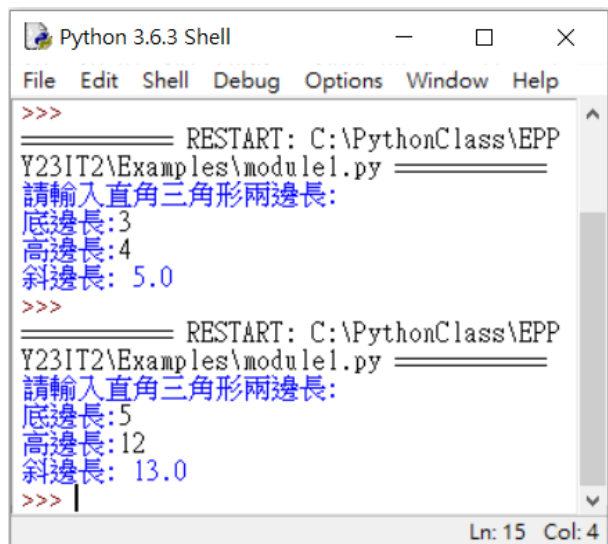
程式範例



```
import math

print('請輸入直角三角形兩邊長:')
x = float(input('底邊長:'))
y = float(input('高邊長:'))
z = math.sqrt(x**2+y**2)
print('斜邊長:', z)
```

Ln: 8 Col: 0



```
>>>
===== RESTART: C:\PythonClass\EPP
Y23IT2\Examples\module1.py =====
請輸入直角三角形兩邊長:
底邊長:3
高邊長:4
斜邊長: 5.0
>>>
===== RESTART: C:\PythonClass\EPP
Y23IT2\Examples\module1.py =====
請輸入直角三角形兩邊長:
底邊長:5
高邊長:12
斜邊長: 13.0
>>> |
```

Ln: 15 Col: 4

常用模組：random模組

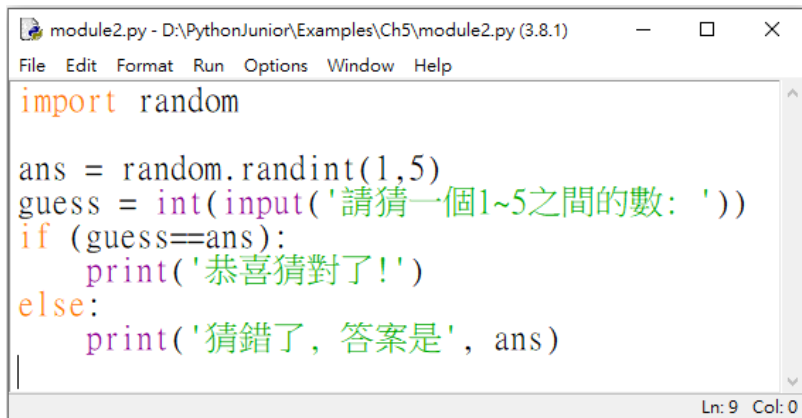
◆ random 模組

◆ 產生隨機亂數或選擇的工具

◆ 常用函式:

- `random.randint(start, stop)`：產生 `start` 到 `stop` 間隨機整數。
- `random.random()`：產生 0.0 到 1.0 間隨機浮點數。
- `random.choice(seq/str)`：由序列`seq`/字串`str`中隨機抽選一個元素/字元
- `random.randrange(start, stop[, step])`：由範圍中隨機選取一個整數。
 - ▶ 效果等同`random.choice(range(start, stop[, step]))`
- `random.sample(population, k)`：由總體`population`中隨機取樣`k`個元素

程式範例

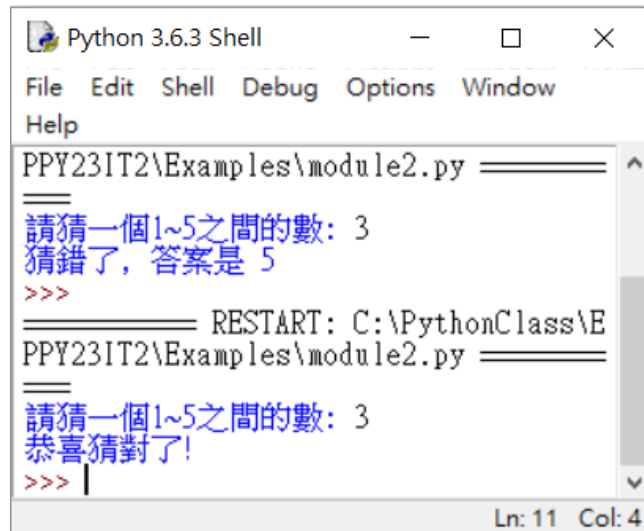


A screenshot of a Python IDE window titled "module2.py - D:\PythonJunior\Examples\Ch5\module2.py (3.8.1)". The window contains the following Python code:

```
import random

ans = random.randint(1,5)
guess = int(input('請猜一個1~5之間的數: '))
if (guess==ans):
    print('恭喜猜對了!')
else:
    print('猜錯了, 答案是', ans)
```

The status bar at the bottom right indicates "Ln: 9 Col: 0".



A screenshot of a Python 3.6.3 Shell window titled "Python 3.6.3 Shell". The window shows the execution of the program from the previous slide. The output is as follows:

```
PPY23IT2\Examples\module2.py ==
==
請猜一個1~5之間的數: 3
猜錯了, 答案是 5
>>>
===== RESTART: C:\PythonClass\E
PPY23IT2\Examples\module2.py =====
==
請猜一個1~5之間的數: 3
恭喜猜對了!
>>> |
```

The status bar at the bottom right indicates "Ln: 11 Col: 4".

程式範例

```
module3.py - D:\Python\Junior\Examples\Ch5\module3.py (3.8.1)
File Edit Format Run Options Window Help
import random

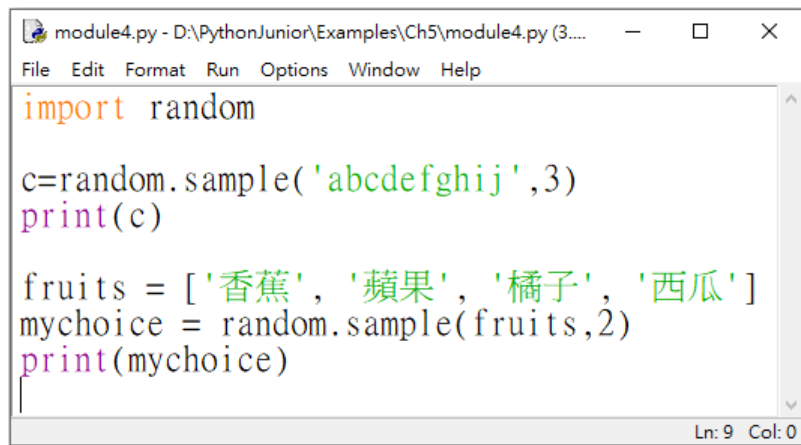
fruits = ['香蕉', '蘋果', '橘子', '西瓜']
mychoice = random.choice(fruits)
print(fruits)
yourchoice = input('請選擇上列水果: ')
if(mychoice==yourchoice):
    print('你的選擇和我相同!')
else:
    print('你的選擇和我不同!我選'+mychoice)
```

Ln: 11 Col: 0

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: C:/PythonClass/EPP
Y23IT2/Examples/module3.py =====
['香蕉', '蘋果', '橘子', '西瓜']
請選擇上列水果: 蘋果
你的選擇和我不同!我選橘子
>>>
===== RESTART: C:/PythonClass/EPP
Y23IT2/Examples/module3.py =====
['香蕉', '蘋果', '橘子', '西瓜']
請選擇上列水果: 蘋果
你的選擇和我相同!
>>>
```

Ln: 18 Col: 4

程式範例

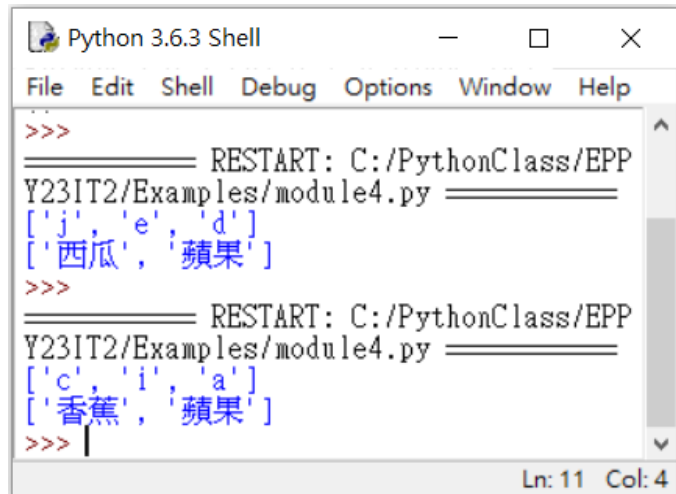


```
import random

c=random.sample('abcdefghij',3)
print(c)

fruits = ['香蕉', '蘋果', '橘子', '西瓜']
mychoice = random.sample(fruits,2)
print(mychoice)
```

Ln: 9 Col: 0



```
>>>
===== RESTART: C:/PythonClass/EPP
Y23IT2/Examples/module4.py =====
['j', 'e', 'd']
['西瓜', '蘋果']
>>>
===== RESTART: C:/PythonClass/EPP
Y23IT2/Examples/module4.py =====
['c', 'i', 'a']
['香蕉', '蘋果']
>>>
```

Ln: 11 Col: 4

自訂模組

◆ 自訂模組

◆ 一個 .py 檔案就是一個模組 (module)

- 與一般Python主程式檔沒有什麼區別
- 檔案名稱即為預設模組名稱
- 可包含變數資料、函式操作或類別定義

匯入自訂模組

◆ 匯入自訂模組，使用自訂模組的函式或變數資料

◇ 匯入模組

- import 模組
- import 模組 as 別名名稱
- from 模組 import 模組內函式或變數

◇ 呼叫函式或讀取變數資料

- 模組名稱.函式名稱() / 別名名稱.函式名稱() / 函式名稱()
- 模組名稱.變數名稱 / 別名名稱.變數名稱 / 變數名稱

程式範例

```
calculator.py - D:\PythonJunior\Examples\Ch5\calcul...
File Edit Format Run Options Window Help

pi = 3.14159

def calcArea(r):
    return pi*r*r

def toFahrenheit(tempC):
    return 9/5*tempC + 32

def calcBmi(height, weight):
    return weight/(height/100)**2

def getMessage(bmi):
    if bmi <= 18.5:
        message = '您的體重過輕'
    elif 25 > bmi > 18.5:
        message = '您的體重正常'
    elif bmi >= 25 and bmi < 30:
        message = '您的體重過重'
    elif bmi > 30:
        message = '您的體重肥胖'
    return message
```

Ln: 22 Col: 0

```
import1.py - D:\PythonJunior\Examples\Ch5\import1.py (3.8.1)
File Edit Format Run Options Window Help

import calculator

r = int(input('請輸入圓半徑(公分):'))
area = calculator.calcArea(r)

print('半徑為%d公分的圓,面積為%.2f平方公分' %(r,area))
```

Ln: 7 Col: 0

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

===== RESTART: D:\PythonJunior\Examples\Ch5\import1.py =====
請輸入圓半徑(公分):10
半徑為10公分的圓,面積為314.16平方公分
>>> |
```

Ln: 7 Col: 4

程式範例

```
calculator.py - D:\PythonJunior\Examples\Ch5\calcul...
File Edit Format Run Options Window Help
pi = 3.14159

def calcArea(r):
    return pi*r*r

def toFahrenheit(tempC):
    return 9/5*tempC + 32

def calcBmi(height, weight):
    return weight/(height/100)**2

def getMessage(bmi):
    if bmi <= 18.5:
        message = '您的體重過輕'
    elif 25 > bmi > 18.5 :
        message = '您的體重正常'
    elif bmi >= 25 and bmi < 30:
        message = '您的體重過重'
    elif bmi > 30:
        message = '您的體重肥胖'
    return message
```

Ln: 22 Col: 0

```
import2.py - D:\PythonJunior\Examples\Ch5\import2.py (3.8.1)
File Edit Format Run Options Window Help
import calculator as temp

tc = float(input("請輸入攝氏溫度:"))
tf = temp.toFahrenheit(tc)

print("攝氏%.2f度等於華氏%.2f度" %(tc,tf))
```

Ln: 7 Col: 0

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: D:\PythonJunior\Examples\Ch5\import2.py =====
請輸入攝氏溫度:50
攝氏50.00度等於華氏122.00度
>>> |
```

Ln: 5 Col: 4

程式範例

```
calculator.py - D:\PythonJunior\Examples\Ch5\calcul...
File Edit Format Run Options Window Help
pi = 3.14159

def calcArea(r):
    return pi*r*r

def toFahrenheit(tempC):
    return 9/5*tempC + 32

def calcBmi(height, weight):
    return weight/(height/100)**2

def getMessage(bmi):
    if bmi <= 18.5:
        message = '您的體重過輕'
    elif 25 > bmi > 18.5:
        message = '您的體重正常'
    elif bmi >= 25 and bmi < 30:
        message = '您的體重過重'
    elif bmi > 30:
        message = '您的體重肥胖'
    return message
```

Ln: 22 Col: 0

```
import3.py - D:\PythonJunior\Examples\Ch5\import3.py (3.8.1)
File Edit Format Run Options Window Help
from calculator import calcBmi, getMessage

h = int(input('身高:'))
w = int(input('體重:'))
bmi = calcBmi(h,w)
message = getMessage(bmi)

print('BMI值為%.2f, 診斷結果%s' %(bmi,message))
```

Ln: 9 Col: 0

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: D:\PythonJunior\Examples\Ch5\import3.py =====
身高:178
體重:80
BMI值為25.25, 診斷結果您的體重過重
>>> |
```

Ln: 10 Col: 4

Python 套件

◆ Python 提供套件的功能

◆ 將多個模組組織起來

- 同目錄結構

```
import sound.effects.echo
sound.effects.echo.filter(delay=0.7, atten=4)
```

◆ 便於多模組的維護管理

- 可有效避免命名衝突

```
import sound.effects.echo as echo
echo.filter(delay=0.7, atten=4)
```

◆ 使用套件時，用 . 表達路徑結構關係

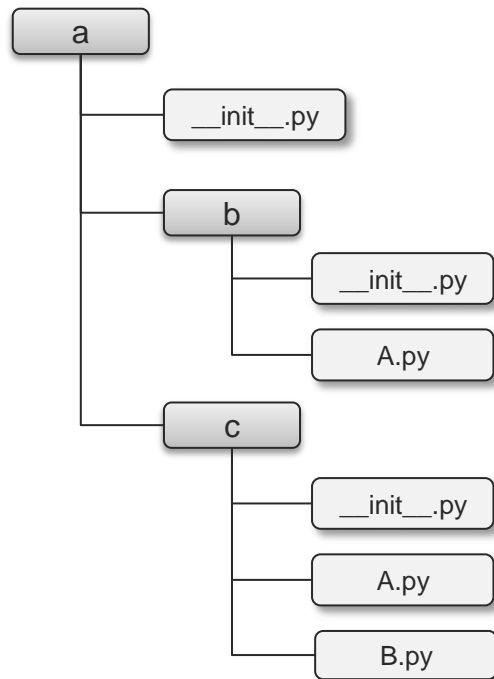
◆ 匯入模組時加上套件路徑資訊

```
sound/
  __init__.py
  formats/
    __init__.py
    wavread.py
    wavwrite.py
    aiffread.py
    aiffwrite.py
    aread.py
    auwrite.py
    ...
  effects/
    __init__.py
    echo.py filter()
    surround.py
    reverse.py
    ...
  filters/
    __init__.py
    equalizer.py
    vocoder.py
    karaoke.py
    ...
```

建立套件

◆ 建立套件

- ◆ 建立資料夾結構，將模組檔案放在資料夾中
- ◆ 目錄中包含一個__init__.py 檔案
 - __init__.py 可以是一個空檔案
 - 或撰寫套件初始化所需的程式
 - ▶ __all__()



匯入/使用套件

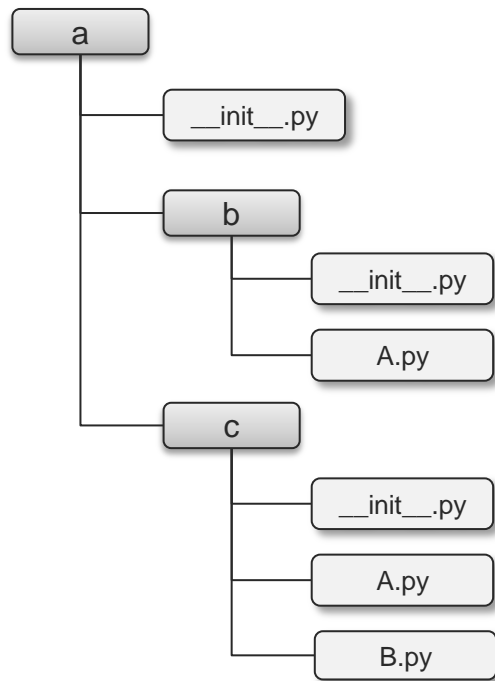
◆ 匯入套件

- ◆ import 套件.模組
- ◆ import 套件.模組 as 別名名稱
- ◆ from 套件 import 模組
- ◆ from 套件.模組 import 函式或變數

◆ 使用套件

- ◆ 套件.模組.函式或變數
- ◆ 別名名稱.函式或變數
- ◆ 模組.函式或變數
- ◆ 函式或變數

程式範例



```
__init__.py - C:\PythonClass...  
File Edit Format Run Options Window Help  
Ln: 1 Col: 0
```

```
A.py - C:\PythonClass\...\A.py...  
File Edit Format Run Options Window Help  
name = 'a.b.A.py'  
def show():  
    print('I am', name)  
Ln: 4 Col: 0
```

```
A.py - C:\PythonClass\...\A.py ...  
File Edit Format Run Options Window Help  
name = 'a.c.A.py'  
def show():  
    print('I am', name)  
Ln: 4 Col: 0
```

```
B.py - C:\PythonClass\...\B.py ...  
File Edit Format Run Options Window Help  
name = 'a.c.B.py'  
def show():  
    print('I am', name)  
Ln: 4 Col: 0
```

```
package1.py - C:\PythonClass/...  
File Edit Format Run Options Window Help  
import a.b.A  
from a.c import A  
import a.c.B as B  
  
print(a.b.A.name)  
print(A.name)  
print(B.name)  
print()  
  
a.b.A.show()  
A.show()  
B.show()  
Ln: 13 Col: 0
```

```
Python 3.6.3 Shell  
File Edit Shell Debug Options Window Help  
Class/package1.py  
a.b.A.py  
a.c.A.py  
a.c.B.py  
  
I am a.b.A.py  
I am a.c.A.py  
I am a.c.B.py  
>>> |  
Ln: 12 Col: 4
```

Q：下列關於Python模組敘述何者正確？

- a) Python 模組儲存在副檔名.mod的檔案中
- b) Python 可由程式副檔名判斷內容是模組還是Script主流程
- c) Python內建的標準函式庫，透過模組管理各種不同的函式
- d) 將程式碼放在不同的模組中，可以讓程式碼架構更有條理

Q：下列關於模組引用敘述何者錯誤？

- a) 使用import car 引入car模組後，可直接呼叫模組中的函式start()
- b) from car import start 這個語法表示從car模組中引入start函式，此時主流程中可直接呼叫start()
- c) 模組名稱太長時，可以使用as替模組取一個較簡單的別名
- d) from car import * 表示引入car模組中所有函式

Q：下列關於套件(Package)敘述何者錯誤？

- a) 套件是模組概念的延伸，把各種相關模組彙整成一整組工具
- b) 套件提供不同的命名空間(namespace)，可避免名稱衝突
- c) 套件所在資料夾中擁有 `__init__.py` 檔案
- d) `__init__.py` 檔案中可以撰寫套件初始化的程式碼
- e) `include` 語法用來引用套件，`import` 語法用來引用模組

Q：下列程式碼片段，何者可計算`my_list`中元素的總和？

```
import math
```

```
my_list = [.3, .3, .3, .1, .1]
```

- a) `math.floor(my_list)`
- b) `math.ceil(my_list)`
- c) `math.fsum(my_list)`
- d) `math.factorial(my_list)`

練習：樂透開獎

- ◆ 隨機產生七個在1-49之間的號碼
- ◆ 從中取出一個特別號
- ◆ 由小到大列印大樂透號碼
- ◆ 列印特別號

練習：猜數字遊戲

- ◆ 隨機產生一個1-100之間的答案 `answer`
- ◆ 玩家猜一個數字 `guess`
 - ◇ 數字猜中了遊戲結束
 - ◇ `guess > answer`，顯示數字太大，下一次猜測上限為`guess`
 - ◇ `guess < answer`，顯示數字太小，下一次猜測下限為`guess`
- ◆ 玩家有五次猜的機會，五次猜不中公布答案

課程內容

1. 模組與套件

- 1-1. 模組引用
- 1-2. 常用模組
- 1-3. 自訂模組
- 1-4. 套件宣告與使用
- 1-5. 匯入模組注意事項

2. 常用標準庫模組

- 2-1. sys模組
- 2-2. os模組
- 2-3. shutil模組
- 2-4. date/time模組

標準程式庫常用模組

◆ Python標準函式庫

- ◆ Python 提供完整強大的標準函式庫，能輕鬆完成很多常見任務
- ◆ math 模組 / random 模組
- ◆ sys 模組：Python系統環境資訊
- ◆ os 模組：作業系統相關的操作函式
- ◆ shutil 模組：日常檔案和目錄管理任務
- ◆ date/time/datetime 模組：日期與時間

標準程式庫常用模組：sys 模組

◆ sys 模組

◇ Python系統環境資訊屬性

- sys.modules：匯入模組資訊
- sys.path：模組搜尋路徑
- sys.argv：命令列參數
- sys.stdout：標準輸出
- sys.stderr：錯誤輸出
- sys.stdin：標準輸入

程式範例

```
stdLib1.py - D:\PythonJunior\Examples\Ch5\stdLib1.py (3.8.1)
File Edit Format Run Options Window Help

import sys

print("匯入模組資訊: ", sys.modules)

print("模組搜尋路徑: ", sys.path)

print("命令列參數: ", sys.argv)

sys.stdout.write("標準輸出: Hello World!\n")
sys.stderr.write("錯誤輸出: Hello World!\n")

sys.stdout.write("請輸入姓名:")
print("Hello", sys.stdin.readline())

Ln: 14 Col: 0
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: D:\PythonJunior\Examples\Ch5\stdLib1.py =====
匯入模組資訊: Squeezed text (181 lines).
模組搜尋路徑: ['D:\\PythonJunior\\Examples\\Ch5', 'C:\\Python38\\python38.zip', 'C:\\Python38\\DLLs', 'C:\\Python38\\lib', 'C:\\Python38', 'C:\\Python38\\lib\\site-packages', 'C:\\Python38\\lib\\site-packages\\win32', 'C:\\Python38\\lib\\site-packages\\win32\\lib', 'C:\\Python38\\lib\\site-packages\\Pythonwin']
命令列參數: ['D:\\PythonJunior\\Examples\\Ch5\\stdLib1.py']
標準輸出: Hello World!
錯誤輸出: Hello World!
請輸入姓名:Sean
Hello Sean

>>> |

Ln: 11 Col: 4
```

標準程式庫常用模組：os 模組

◆ os 模組

◇ 作業系統相關的操作函式

- `os.name`：系統名稱，'posix'/'nt'/'ce'/'java'
- `os.getcwd()`：顯示當前目錄
- `os.chdir(path)`：切換目錄路徑
- `os.listdir(path='.')`：列出指定目錄內容
- `os.system()`：執行作業系統命令

程式範例

```
stdLib2.py - D:\PythonJunior\Examples\Ch5\stdLib2.py (3.8.1)
File Edit Format Run Options Window Help

import os

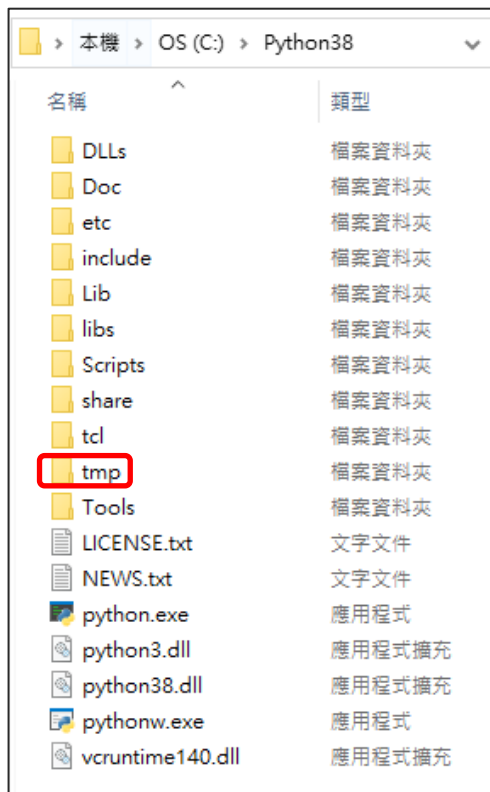
print(os.name) # 顯示系統名稱 'posix'/'nt'/'ce'/'java'
print(os.getcwd()) # 顯示當前目錄
os.chdir(r"C:\Python38") # 切換路徑
print(os.getcwd()) # 顯示當前目錄
print(os.listdir(r"C:\Python38")) # 列出目錄內容
os.system('mkdir tmp') # 執行系統命令

Ln: 9 Col: 0
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

\stdLib2.py =====
nt
D:\PythonJunior\Examples\Ch5
C:\Python38
['DLLs', 'Doc', 'etc', 'include', 'Lib', 'lib
s', 'LICENSE.txt', 'NEWS.txt', 'python.exe',
'python3.dll', 'python38.dll', 'pythonw.exe',
'Scripts', 'share', 'tcl', 'Tools', 'vcruntim
e140.dll']
>>> |

Ln: 17 Col: 4
```



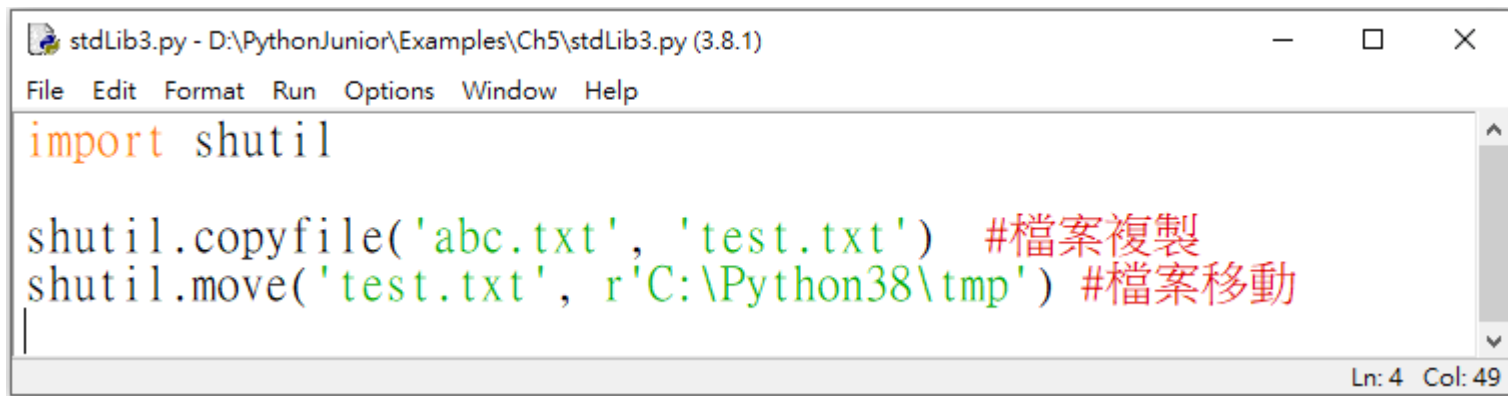
標準程式庫常用模組：shutil 模組

◆ shutil 模組

◇ 檔案和目錄管理任務

- `shutil.move(src, dst, copy_function=copy2)`：檔案移動或改名
- `shutil.copy(src, dst, *, follow_symlinks=True)`：檔案或目錄複製
- `shutil.copyfile(src, dst, *, follow_symlinks=True)`：檔案複製
- `shutil.copytree(src, dst, symlinks=False, ignore=None, copy_func=copy2, ignore_dangling_symlinks=False)`：複製目錄及其目錄內檔案
- `shutil.rmtree(path, ignore_err=False, onerr=None)`：刪除目錄及其內檔案

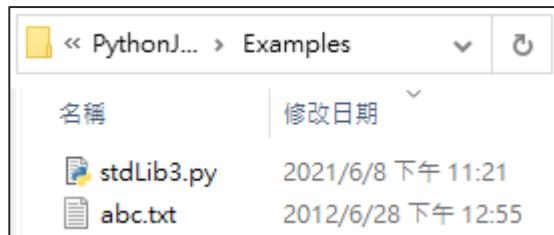
程式範例



```
stdLib3.py - D:\PythonJunior\Examples\Ch5\stdLib3.py (3.8.1)
File Edit Format Run Options Window Help
import shutil

shutil.copyfile('abc.txt', 'test.txt') #檔案複製
shutil.move('test.txt', r'C:\Python38\tmp') #檔案移動

Ln: 4 Col: 49
```



PythonJ... > Examples	
名稱	修改日期
stdLib3.py	2021/6/8 下午 11:21
abc.txt	2012/6/28 下午 12:55



OS (C:) > Python38 > tmp	
名稱	修改日期
test.txt	2021/6/8 下午 11:24

標準程式庫常用模組：time 模組

◆ time 模組

◆ 系統時間、計時相關的內容

◆ `time.time()`：傳回系統時間數值，1970/1/1 00:00:00 (UTC)起算的毫秒數

◆ `time.ctime()`：傳回系統時間字串表示，Thu Nov 16 16:36:06 2017

◆ `time.localtime([t])`：傳回時間結構物件(年,月,日,時,分,秒,星期,天數,夏令時間)

◆ `time.sleep(secs)`：程式暫停指定的秒數

標準程式庫常用模組：datetime 模組

- ◆ datetime 模組，內含date / time / datetime / timedelta 類別
- ◆ date 類別代表日期相關的內容
 - ◇ date(year, month, day)：建立指定日期物件
 - ◇ date.today()：傳回目前系統日期時間物件
 - ◇ date.strftime(format)：設定系統日期顯示字串
 - %y：年(2位), %Y：年(4位), %m：月(數字), %b：月(文字), %d：日, %w：星期
 - ◇ date.isoformat() / str(date) 取得日期isoformat格式 yyyy-mm-dd
 - ◇ date.isoweekday() 取得1(星期一)~7(星期日)

標準程式庫常用模組：datetime 模組

◆ time類別代表時間相關的內容

- ◆ `time(hour,minute,second,millisecond)`：建立指定時間物件
- ◆ `time()`：建立00:00:00時間物件
- ◆ `time.strftime(format)`：設定系統時間顯示字串

◆ datetime類別代表日期加時間的內容

- ◆ `datetime(year, month, day, hour, minute, second, millisecond)`：建立指定日期加時間物件

◆ timedelta類別代表時間差

- ◆ `date`相減會變成`timedelta`
- ◆ 以`days`為計算單位

程式範例

```
stdLib4-1.py - D:\PythonJunior\Examples\Ch5\stdLib4-1.py (3.8.1)
File Edit Format Run Options Window Help

import time

print(time.time())
time.sleep(2)
print(time.ctime())
time.sleep(2)
lt = time.localtime()
print(lt)
print(lt.tm_year, lt.tm_mon, lt.tm_mday)

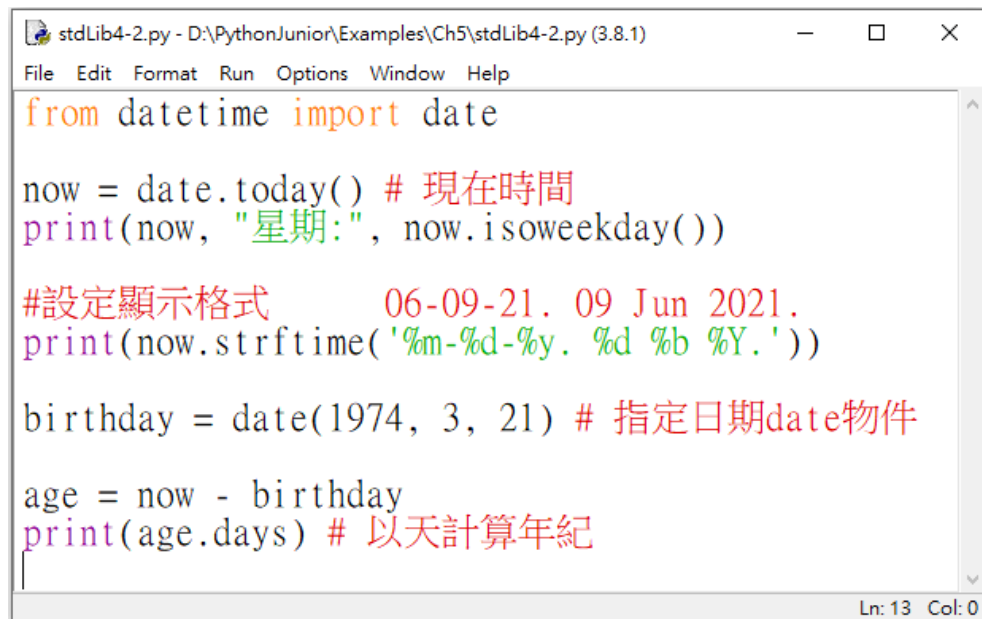
Ln: 10 Col: 0
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

= RESTART: D:\PythonJunior\Examples\Ch5\stdLib4-1.py
1623169545.6287854
Wed Jun  9 00:25:47 2021
time.struct_time(tm_year=2021, tm_mon=6, tm_mday=9, tm_
_hour=0, tm_min=25, tm_sec=49, tm_wday=2, tm_yday=160,
tm_isdst=0)
2021 6 9
>>> |

Ln: 25 Col: 4
```

程式範例



```
stdLib4-2.py - D:\PythonJunior\Examples\Ch5\stdLib4-2.py (3.8.1)
File Edit Format Run Options Window Help
from datetime import date

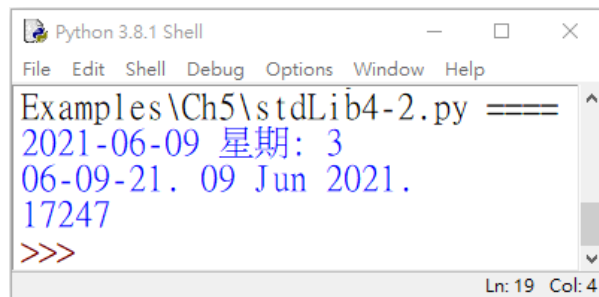
now = date.today() # 現在時間
print(now, "星期:", now.isoweekday())

#設定顯示格式      06-09-21. 09 Jun 2021.
print(now.strftime('%m-%d-%y. %d %b %Y.'))

birthday = date(1974, 3, 21) # 指定日期date物件

age = now - birthday
print(age.days) # 以天計算年紀
```

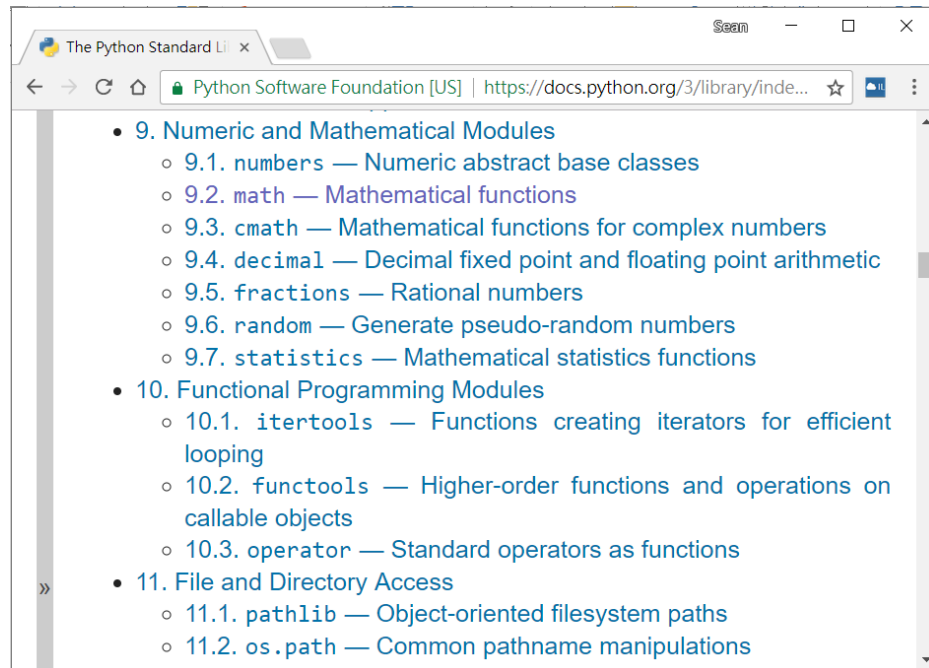
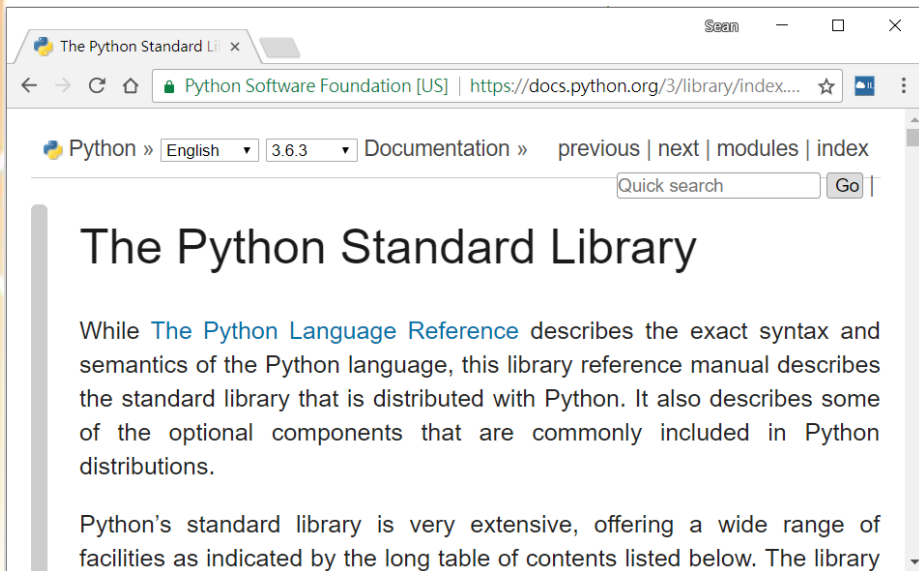
Ln: 13 Col: 0



```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Examples\Ch5\stdLib4-2.py ===
2021-06-09 星期: 3
06-09-21. 09 Jun 2021.
17247
>>>
```

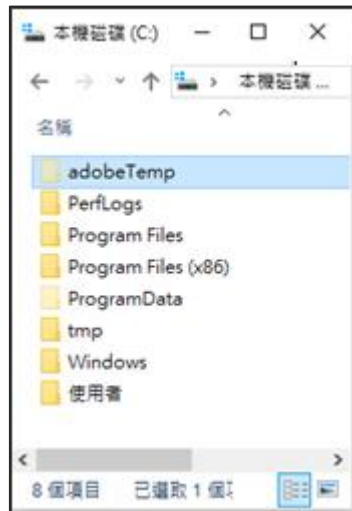
Ln: 19 Col: 4

Python 標準函式庫文件



Q：下列哪個選項可建立C:\myFolder\test資料夾？

- a) `os.chdir(r'C:\myFolder\test')`
- b) `os.mkdir(r'C:\myFolder\test')`
- c) `print(os.listdir(r'C:\myFolder\test'))`
- d) `os.makedirs(r'C:\myFolder\test')`



Q：下列程式碼片段，哪個選項可顯示兩個時間相隔秒數？

```
import datetime

start = datetime.date(2018, 1, 15)
end = datetime.date(2018, 1, 20)

diff = end - start
```

- a) `print(int(datetime.timedelta(diff)))`
- b) `print(diff.total_seconds())`
- c) `print(diff/60)`
- d) `print(diff)`

練習：日期倒數計算器

- ◆ 寫一個程式計算今天到今年重大活動日期還有幾天
 - ◇ 生日、開學、中秋、明星演唱會、聖誕節、學測等



巨匠直播教學

www.pcschoolonline.com.tw