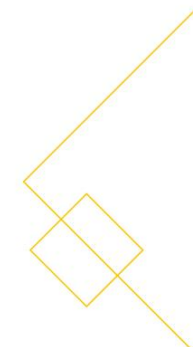




Ch7模組、套件 與獨立程式




7-1 模組

- 模組就是一個Python 檔案，每一個Python 檔案被視為一個模組，可以在程式中匯入其他Python 模組，模組就可以不斷地被其他程式再利用。
- 到此已經介紹程式的關鍵字(**if**、**for**、**while**...)、變數與運算子，可以想成單字

- 多個關鍵字、變數與運算子組合成一行程式，可以想成句子
- 多行程式可以組合成函式，可以想成段落
- 多個函式可以組合成模組，可以想成是一篇文章，以下介紹模組的實作與匯入模組。

7-1-1 實作模組

- Python 的模組就是一個檔案，實作一個模組，可以隨機回傳「剪刀」、「石頭」、「布」三個其中一個。

| 行號 | 範例 ( : ch7\guess.py) |
|----|--|
| 1 | <code>import random</code> |
| 2 | <code>status = ['剪刀', '石頭', '布']</code> |
| 3 | <code>def figure_guess():</code> |
| 4 | <code> return random.choice(status)</code> |

程式解說

- 第1 行：匯入模組random。
- 第2 行：定義變數status為串列，串列的內容有「剪刀」、「石頭」、「布」。
- 第3 到4 行：自訂函式figure_guess，使用random 模組的函式choice，隨機從變數status 中挑選一個元素出來，經由return 回傳選出來的元素。


7-1-2 匯入模組

- 有了模組後，其他程式就可以將模組匯入，使用「import」進行匯入，匯入的方法如下。

- 使用「**import 模組名稱**」匯入整個模組，該模組(檔案)要在系統所指定的路徑內，系統會依序找尋所指定的路徑下是否有該模組(檔案)
- 可以將模組與匯入該模組的程式放在同一個資料夾下，這樣一定能夠匯入，接著使用「**模組名稱. 函式名稱()**」執行匯入模組的函式

- 匯入整個模組
 - 例如：新增一個檔案在資料夾ch7 下，取名為 **ch7-1-2-1a-mod.py**
 - 在資料夾ch7 下有檔案 **guess.py**，程式中匯入模組guess，使用「`guess.figure_guess()`」呼叫模組guess 的函式figure_guess。

7-1-2 匯入模組

| 行號 | 範例 ( : ch7\ch7-1-2-1a-mod.py) | 執行結果 |
|----|---|------|
| 1 | <code>import guess</code> | 石頭 |
| 2 | <code>computer = guess.figure_guess()</code> | |
| 3 | <code>print(computer)</code> | |


- 第1 行：匯入模組guess。
- 第2 行：呼叫模組guess 的函式 `figure_guess`，隨機從「剪刀」、「石頭」與「布」中挑選一個回傳到變數computer。
- 第3 行：使用函式 `print` 顯示變數computer 的內容到螢幕上。

7-1-2 匯入模組

- 只匯入模組中想要的函式
 - 使用「**from 模組名稱 import 函式名稱**」匯入模組中的特定函式，該模組(檔案)要在系統所指定的路徑內
 - 系統會依序找尋所指定的路徑下是否有該模組(檔案)，可以將模組與匯入該模組的程式放在同一個資料夾下，這樣一定能夠匯入，接著使用「**函式名稱()**」執行匯入模組的函式

- 只匯入模組中想要的函式
 - 例如：新增一個檔案在資料夾ch7 下，取名為ch7-1-2-1b-mod.py，在資料夾ch7 下有檔案guess.py，程式中匯入模組guess，使用「figure_guess()」呼叫模組guess 的函式figure_guess。

7-1-2 匯入模組

| 行號 | 範例 ( : ch7\ch7-1-2-1b-mod.py) | 執行結果 |
|----|---|------|
| 1 | from guess import figure_guess | 剪刀 |
| 2 | computer = figure_guess() | |
| 3 | print(computer) | |

程式解說


- 第1 行：匯入模組guess的函式figure_guess。
- 第2行：呼叫函式figure_guess，隨機從「剪刀」、「石頭」與「布」中挑選一個回傳到變數computer。
- 第3 行：使用函式print 顯示變數computer的內容到螢幕上。

7-1-2 匯入模組

- 使用別名匯入模組
 - 使用「**import 模組 as 別名**」匯入模組並命名為「別名」，該模組(檔案)要在系統所指定的路徑內，系統會依序找尋所指定的路徑下是否有該模組(檔案)，可以將模組與匯入該模組的程式放在同一個資料夾下，這樣一定能夠匯入，接著使用「別名. 函式名稱()」執行匯入模組的函式

- 使用別名匯入模組
 - 例如：新增一個檔案在資料夾ch7 下，取名為 **ch7-1-2-1c-mod.py**
 - 在資料夾ch7 下有檔案 **guess.py**，程式中匯入模組guess，重新命名為gs，使用「gs.figure_guess()」呼叫模組gs 的函式figure_guess。

7-1-2 匯入模組

| 行號 | 範例 ( : ch7\ch7-1-2-1c-mod.py) | 執行結果 |
|----|---|------|
| 1 | <code>import guess as gs</code> | 剪刀 |
| 2 | <code>computer = gs.figure_guess()</code> | |
| 3 | <code>print(computer)</code> | |


程式解說

- 第1 行：匯入模組`guess`，重新命名為`gs`。
- 第2 行：呼叫模組`gs` 的函式`figure_guess`，隨機從「剪刀」、「石頭」與「布」中挑選一個回傳到變數`computer`。
- 第3 行：印出變數`computer`到螢幕上。

7-1-3 匯入模組的路徑

- 若想要知道Python 匯入模組的資料夾路徑與順序，需先匯入模組sys，讀取sys.path的每一個元素就可以知道，可以發現第一個找尋模組是否存在的資料夾就在執行程式的資料夾下，若找到就不會到下一個資料夾去找尋。

7-1-3 匯入模組的路徑

| 行號 | 範例 ( : ch7\ch7-1-3-mod.py) | 程式說明 |
|----|--|---|
| 1 | <code>import sys</code> | 第 1 行：匯入模組 <code>sys</code> 。 |
| 2 | <code>for path in sys.path:</code> | 第 2 到 3 行：使用 <code>for</code> 迴圈依序讀取模組 <code>sys</code> 的屬性 <code>path</code> 到變數 <code>path</code> ，印出變數 <code>path</code> 到螢幕上。 |
| 3 | <code>print(path)</code> | |

執行結果

```
K:\mybook\python\ch7  
K:\mybook\python  
C:\Users\user\AppData\Local\Programs\Python\Python36-32\python36.zip  
C:\Users\user\AppData\Local\Programs\Python\Python36-32\DLLs  
C:\Users\user\AppData\Local\Programs\Python\Python36-32\lib  
C:\Users\user\AppData\Local\Programs\Python\Python36-32  
C:\Users\user\AppData\Local\Programs\Python\Python36-32\lib\site-packages
```

7-2 套件

- 多個模組(檔案)放在同一個資料夾下，在該資料夾下新增一個檔案，檔案名稱為「`__init__.py`」，該資料夾就形成套件。

7-2-1 實作套件

- 實作一個套件game，新增dice.py 可以產生擲骰子的點數，與新增poker.py 可以產生撲克牌的花色與點數，將這兩個檔案放在資料夾game 下，在資料夾game 下新增一個檔案「__init__.py」
- 檔案「__init__.py」的內容可以是空的，資料夾與檔案的關係如下。

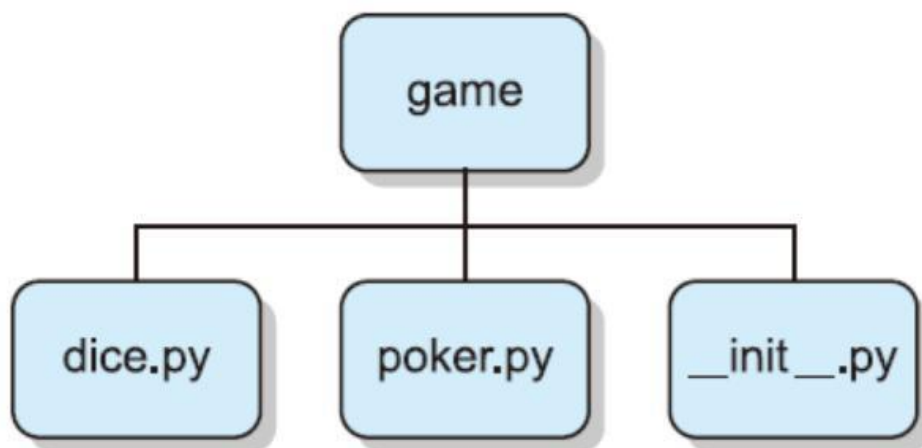



圖 7-1 套件 game 的資料夾關係

7-2-1 實作套件

- 新增dice.py，程式碼如下


| 行號 | 範例 ( : ch7\game\dice.py) |
|----|--|
| 1 | from random import choice |
| 2 | def dice(): |
| 3 | return choice(range(1,7,1)) |

程式解說

- 第1 行：匯入模組random 的choice函式。
- 第2 到3 行：定義dice 函式，隨機回傳數值1 到6 的其中一個數字。

7-2-1 實作套件

- 新增poker.py，程式碼如下：

| 行號 | 範例 ( : ch7\game\poker.py) |
|----|---|
| 1 | from random import choice |
| 2 | def poker(): |
| 3 | a = ['C', 'H', 'D', 'S'] |
| 4 | b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 'T', 'J', 'Q', 'K'] |
| 5 | return choice(a)+str(choice(b)) |

程式解說

- 第1 行：匯入模組random 的函式choice。
- 第2 到5 行：定義poker 函式，串列a 表示花色與串列b 表示點數，隨機挑選串列a 與串列b 中一個元素，回傳挑選出來的花色與點數串接起來的結果。

7-2-2 套件的使用


- 使用「**from 套件名稱 import 模組名稱1, 模組名稱2**」匯入套件中的特定模組，該套件要在系統所指定的路徑內，系統會依序找尋所指定的路徑下是否有該套件
- 可以將套件資料夾與匯入該套件的程式放在同一個資料夾下，這樣一定能夠匯入，接著使用「**模組名稱. 函式名稱()**」執行匯入套件的模組中函式。

7-2-2 套件的使用

- 例如：新增一個Python 檔案在資料夾ch7下，取名為7-2-2-pkg.py

- 前一節在資料夾ch7 下有game 資料夾，game 資料夾下有檔案dice.py 與poker.py
- 程式中匯入套件game 的模組dice 與poker，使用「dice.dice()」呼叫模組dice 的函式dice，使用「poker.poker()」呼叫模組poker 的函式poker。

7-2-2 套件的使用

| 行號 | 範例 ( : ch7\7-2-2-pkg.py) | 執行結果 |
|----|--|------|
| 1 | <code>from game import dice, poker</code> | 2 |
| 2 | <code>for i in range(2):</code> | SQ |
| 3 | <code> print(dice.dice())</code> | 5 |
| 4 | <code> print(poker.poker())</code> | H1 |

程式解說


- 第1 行：匯入套件game 的模組dice與poker。
- 第2到4行：使用for迴圈執行兩次，呼叫模組dice的函式dice隨機產生骰子點數，印出骰子點數到螢幕上，與呼叫模組poker 的函式poker 隨機產生撲克牌花色與點數，印出撲克牌花色與點數到螢幕上。

7-3 腳本程式

- 腳本程式可以當成模組被匯入，也可以成為獨立執行的腳本程式，使用「`if __name__ == '__main__':`」
- 串接在此判斷條件後的程式碼，在腳本程式被當成模組被匯入時，不會被執行，只有在腳本程式獨立執行時才會執行。

7-3-1 實作腳本程式

- 製作一個「剪刀、石頭、布」的小遊戲，電腦出拳為隨機出拳，使用者由介面輸入「剪刀」、「石頭」與「布」
- 比較電腦出拳與使用者出拳決定勝負的結果，使用「`if __name__ == '__main__':`」製作出獨立執行的腳本程式，當執行該腳本程式，就可以玩「剪刀、石頭、布」的遊戲。

| 行號 | 範例 ( : ch7\guess2.py) |
|----|---|
| 1 | from guess import figure_guess |
| 2 | def run(): |
| 3 | computer = figure_guess() |
| 4 | my = input(' 請輸入「剪刀」、「石頭」或「布」？ ') |
| 5 | print(' 電腦出 ', computer) |
| 6 | if my == ' 剪刀 ': |
| 7 | if computer == ' 剪刀 ': |
| 8 | print(' 平手 ') |
| 9 | elif computer == ' 石頭 ': |
| 10 | print(' 電腦獲勝 ') |
| 11 | else: |
| 12 | print(' 玩家獲勝 ') |

```
13     elif my == '石頭':  
14         if computer == '剪刀':  
15             print(' 玩家獲勝 ')  
16         elif computer == '石頭':  
17             print(' 平手 ')  
18         else:  
19             print(' 電腦獲勝 ')  
20     else:  
21         if computer == '剪刀':  
22             print(' 電腦獲勝 ')  
23         elif computer == '石頭':  
24             print(' 玩家獲勝 ')  
25         else:  
26             print(' 平手 ')  
27 if __name__ == '__main__':  
28     for i in range(10):  
29         run()  
30 else:  
31     print(' 我不是獨立執行的 python 程式 ')
```

執行結果

顯示最後兩次的執行結果

```
請輸入「剪刀」、「石頭」或「布」？剪刀  
電腦出 布  
玩家獲勝
```

```
請輸入「剪刀」、「石頭」或「布」？剪刀  
電腦出 剪刀  
平手
```

程式解說

- 第1 行：匯入模組guess的函式figure_guess。
- 第2 到26 行：定義函式run。
- 第3行：呼叫函式figure_guess，回傳從「剪刀」、「石頭」與「布」中挑選一個回傳到變數computer。

- 第4 行：在螢幕上顯示「請輸入「剪刀」、「石頭」或「布」？」，輸入結果指定給變數my。
- 第5 行：顯示「電腦出」與變數computer在螢幕上。


- 第6到12行：若變數my等於「剪刀」，表示玩家出剪刀，則若電腦出「剪刀」，則顯示「平手」，否則若電腦出「石頭」，則顯示「電腦獲勝」，否則(電腦出「布」)顯示「玩家獲勝」(第7到12行)。

- 第13到19行：若變數my等於「石頭」，表示玩家出石頭，則若電腦出「剪刀」，則顯示「玩家獲勝」
- 否則若電腦出「石頭」，則顯示「平手」，否則(電腦出「布」)顯示「電腦獲勝」(第14 到19 行)。

- 第20到26行：否則(變數my等於「布」)，表示玩家出布，則若電腦出「剪刀」，則顯示「電腦獲勝」
- 否則若電腦出「石頭」，則顯示「玩家獲勝」，否則(電腦出「布」)顯示「平手」(第21到26行)。

- 第27到31行：若「`__name__`」等於「`__main__`」，則使用for迴圈執行函式run十次
- 否則顯示「我不是獨立執行的python 程式」


- 若使用「import」匯入獨立執行的腳本程式，則「if __name__ == '__main__':」判斷結果為False，本程式會執行else，顯示「我不是獨立執行的python 程式」。

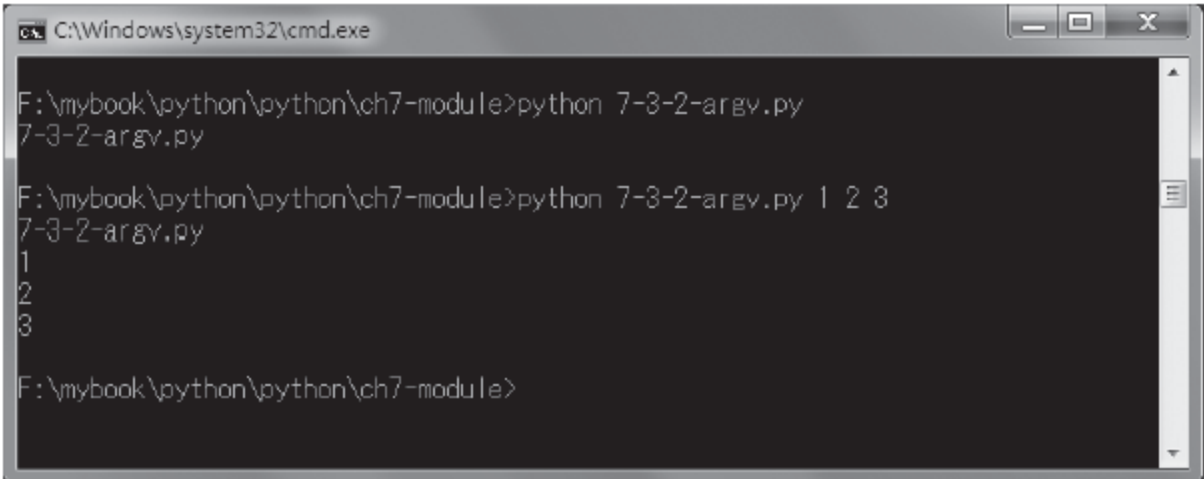
| 行號 | 範例 ( : ch7\7-3-1b-import.py) | 執行結果 |
|----|--|--------------------|
| 1 | <code>import guess2</code> | 我不是獨立執行的 python 程式 |

- 程式解說
 - 第1 行：匯入模組guess2。

7-3-1 實作腳本程式

- 執行Python 腳本程式時，可以於執行腳本程式命令列後方加入引數，例如：
「python 7-3-2-argv.py 1 2 3」

指令列引數範例 ( : ch7\7-3-2-argv.py) , 如下。

| 行號 | 範例 | 執行結果 |
|-------------|--|--|
| 1 2 3 | import sys for i in sys.argv: print(i) | <p>在命令列使用「python」獨立執行 Python 腳本程式，接著在 Python 腳本程式後方加入引數，第一次執行時不加入引數，第二次執行後方加入引數「1 2 3」，執行結果如下圖。</p>  <p>The screenshot shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. It displays two commands and their outputs. The first command is 'python 7-3-2-argv.py' which outputs '7-3-2-argv.py'. The second command is 'python 7-3-2-argv.py 1 2 3' which outputs '1', '2', and '3' on separate lines.</p> |

程式解說

- 第1 行：匯入sys系統模組。
- 第2 到3 行：使用for 迴圈依序從sys.argv中取出每一個元素顯示在螢幕上。