

Python程式設計

林奇賦 daky1983@gmail.com

Outline

- 例外處理
- 輸入輸出

例外處理

- 大部分執行中的錯誤，Python 直譯器 (interpreter) 會以發起例外 (exception) 的方式來中斷程式的執行。實際上，很多情況下我們需要自行控制可能會產生例外的程式碼，因為例外並不全然是程式的邏輯錯誤，例如程式中打算開啟檔案，然而實際檔名並不存在，這種情況下，我們需要的是例外發生後的處理動作，而非中止程式的執行。
- 凡是可能會產生例外的程式碼，Python 利用 **try-except** 陳述 (try-except statement) 讓程式設計師自行處理例外。 **try-except** 為關鍵字 (keyword) 之一，專門用來例外處理 (exception handling) 的。

為什麼需要例外處理

- 除以0

```
a = 10
```

```
b = 0
```

```
print(a/b)
```

```
print("hello python")
```

例外處理

- 基本形式就是把可能會產生例外的程式碼放在 **try** 之後的程式區塊 (**block**)，**except** 則放例外發生時的處置。

加入例外處理後

```
a = 10
```

```
b = 0
```

```
try:
```

```
    print(a/b)
```

```
except:
```

```
    print("例外發生，不可除以0")
```

```
print("hello python")
```

例外處理

- 例外的名稱也可以寫在 **except** 之後，如

```
a = 10
b = 0
try:
    print(a/b)
except NameError:
    print("例外發生，NameError")
except ZeroDivisionError:
    print("例外發生，不可除以0")

print("hello python")
```

例外處理

- **try-except** 也可以和 **else** 連用，**else** 後的程式區塊放的是沒有發生例外，程式所執行的工作，例如

```
a = 10
b = 0
try:
    print(a/b)
except NameError:
    print("例外發生，NameError")
except ZeroDivisionError:
    print("例外發生，不可除以0")
else:
    print("沒有例外發生")
print("hello python")
```


關鍵字 **finally**

- 加入另一個關鍵字 **finally**，無論例外有沒有發生都會執行 **finally** 後的程式區塊。例如

```
a = 10
b = 0
try:
    print(a/a)
except NameError:
    print("例外發生，NameError")
except ZeroDivisionError:
    print("例外發生，不可除以0")
else:
    print("沒有例外發生")
finally:
    print("finally一定會執行到")
print("hello python")
```

例外處理 **raise**陳述

- 例外 (except) 除了會由直譯器 (interpreter) 自動發生外，我們也可以自己在程式中利用 **raise** 陳述 (raise statement) 引起例外。 **raise** 為關鍵字 (keyword) 之一，用來引起例外。
- 最簡單的形式為單一的 **raise** 陳述
 - **raise NameError**

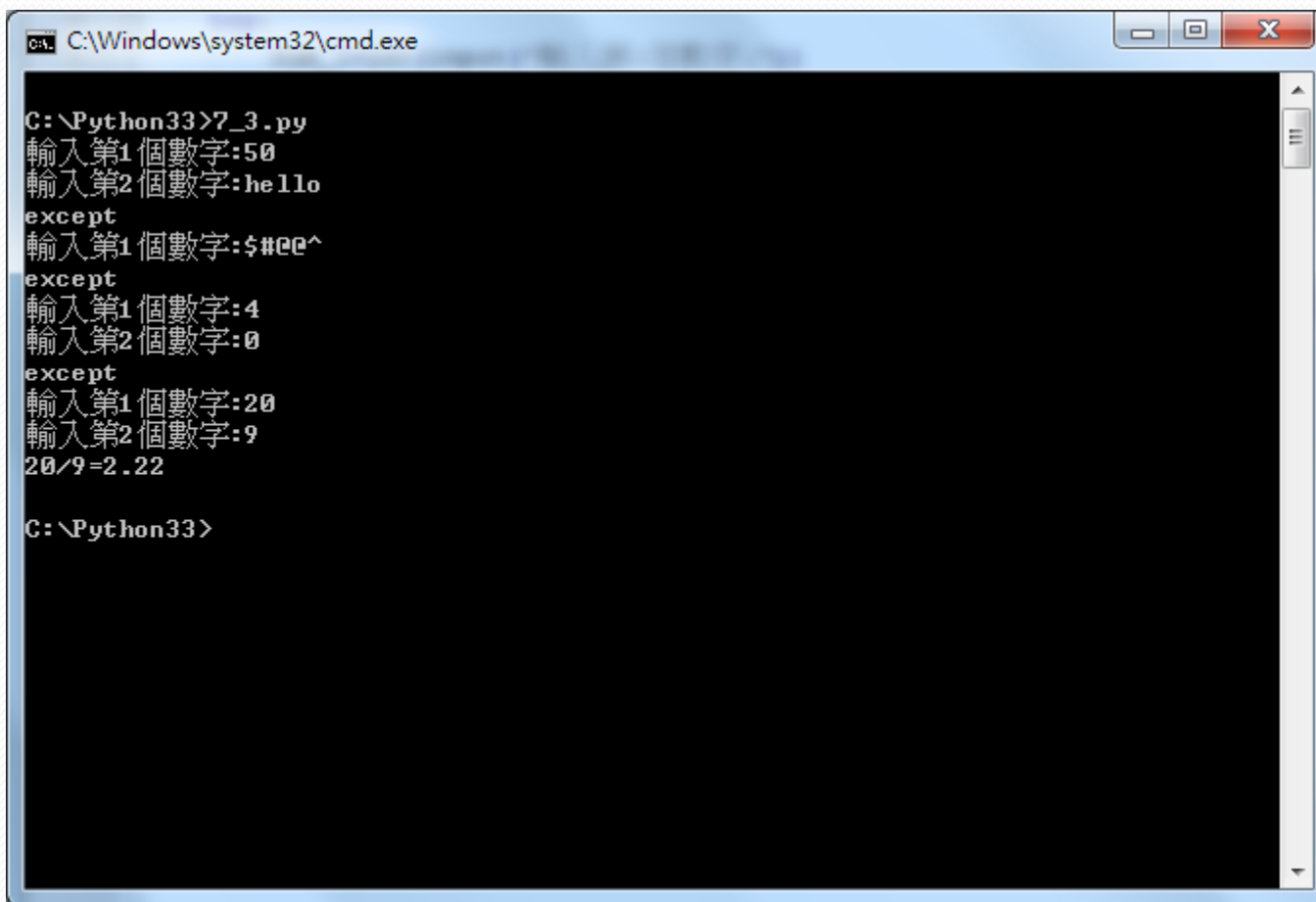
例外處理 raise陳述

```
try:
    raise ZeroDivisionError
except NameError:
    print("發生例外,NameError")
except ZeroDivisionError:
    print("發生例外,ZeroDivisionError")
print("after exception....")
```

課堂練習

- 利用的方法，輸入兩個數字，並將兩個數字相除的結果印出。
- 加入例外處理機制，若發生任何例外則讓使用者重新輸入直到正確將相除的結果印出才終止程式

執行範例



```
C:\Windows\system32\cmd.exe

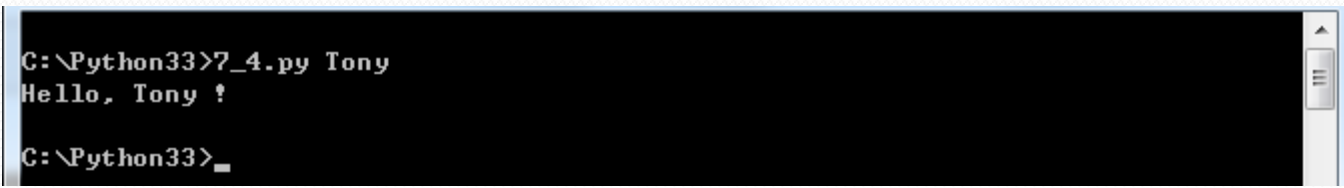
C:\Python33>7_3.py
輸入第1個數字:50
輸入第2個數字:hello
except
輸入第1個數字:$#00^
except
輸入第1個數字:4
輸入第2個數字:0
except
輸入第1個數字:20
輸入第2個數字:9
20/9=2.22

C:\Python33>
```

基本輸入輸出

- 如果要取得使用者的輸入，可以使用`sys.argv`，使用者輸入的命令列引數，會收集為字串陣列並給`sys.argv`參考，索引0是啟動的模組名稱，之後則陸續是使用者所輸入的引數。例如：

```
import sys  
print('Hello, %s !'%sys.argv[1])
```

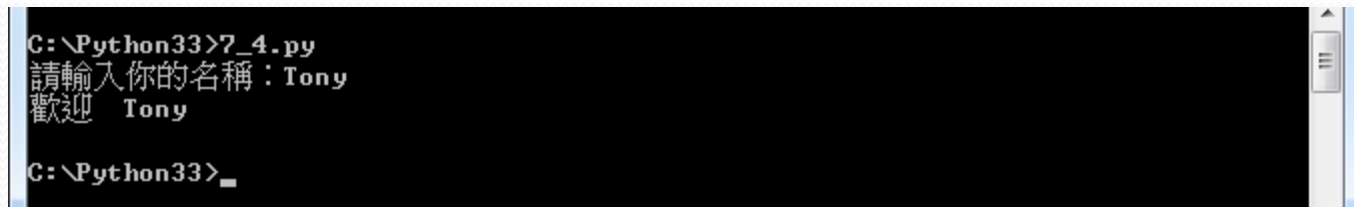
A screenshot of a Windows command prompt window with a black background and white text. The prompt is 'C:\Python33>'. The user has entered '7_4.py Tony', and the output is 'Hello, Tony !'. The prompt is now 'C:\Python33>_'.

```
C:\Python33>7_4.py Tony  
Hello, Tony !  
C:\Python33>_
```

input()

- 在程式執行的過程中，可以使用**input()**函式取得使用者的輸入，**input()**可以指定提示文字，使用者輸入的文字則以字串傳回。例如：

```
name = input('請輸入你的名稱：')  
print('歡迎 ', name)
```

A screenshot of a Windows command prompt window with a black background and white text. The prompt shows the execution of a Python script named 7_4.py. The script prompts the user to enter their name, and the user has entered 'Tony'. The script then prints a welcome message '歡迎 Tony'. The prompt ends with a new line.

```
C:\Python33>7_4.py  
請輸入你的名稱：Tony  
歡迎 Tony  
C:\Python33>_
```

print()

- `print(value, ..., sep=' ', end='\n', file=sys.stdout)`
- 可以看到，除了指定值輸出之外，還可以使用**sep**指定每個輸出值之間的分隔字元，預設值為一個空白，可以使用**end**指定輸出後最後一個字元，預設值是'**\n**'

輸出至檔案

- 預設的輸出是系統標準輸出，可以使用**file**指定至其它的輸出。例如以下會將指定的值輸出至data.txt：

```
print('hello python', file = open('data.txt', 'w'))
```

print 格式化字串

- 格式化字串時，所使用的%d、%f、%s等與C語言類似，之後使用%接上一個tuple，也就是範例中以()包括的實字表示部份。

```
print('%d %.2f %s' % (100, 199.3678, 'Tony'), file = open('data.txt', 'w'))
```

open() 函式

- 將資料寫入檔案或從檔案讀出，可以使用**open()**函式：
**open(file,mode="r",buffering=None,encoding=None,
errors=None,newline=None,closefd=True)**

讀取檔案

- `name = input('請輸入檔名：')`
`file = open(name, 'r', encoding='UTF-8')`
`content = file.read()`
`print(content)`
`file.close()`
- **read()**方法會一次讀取所有的檔案內容，在不使用檔案時，可以使用**close()**將檔案關閉以節省資源。

readline()

```
name = input('請輸入檔名：')  
file = open(name, 'r', encoding='UTF-8')  
while True:  
    line = file.readline()  
    if not line: break  
    print(line, end="")  
file.close()
```

file.readlines()

- **readlines()**方法會用一個字串陣列收集讀取的每一行，for迴圈每次取出字串陣列中的一個字串元素，並使用print()函式顯示

```
name = input('請輸入檔名：')  
file = open(name, 'r', encoding='UTF-8')  
print(file.readlines()[0])  
file.close()
```

- 所以可以修改上張投影片的while迴圈的讀取方式，以for迴圈的方式讀取

file.readlines()

```
name = input('請輸入檔名：')  
file = open(name, 'r', encoding='UTF-8')  
for line in file.readlines():  
    print(line, end="")  
file.close()
```

readlines()

- 事實上，更有效率的方式，是呼叫檔案物件的**next()**方法，**next()**方法每次傳回下一行，並在沒有資料可讀取時丟出**StopIteration**。可以使用**for**迴圈自動呼叫**next()**方法，並在捕捉到**StopIteration**時離開迴圈。例如：

```
name = input('請輸入檔名：')  
for line in open(name, 'r', encoding='UTF-8'):  
    print(line, end="")
```

- 這種寫法會自動關閉檔案

寫資料至檔案

- 使用`open()`函式時，指定模式為'`w`'，並使用`write()`方法進行資料寫入。例如：

```
name = input('請輸入檔名：')  
file = open(name, 'w', encoding = 'UTF-8')  
file.write('寫入檔案測試')  
file.close()
```

課堂練習

- 讀取 stores_old.csv 檔案後，將 sid,name,tel,wifi 篩選出來並寫入檔案至 stores_new.csv 中

	A	B	C	D	E	F	G	H
1	sid	longitude	latitude	name	address	tel	wifi	notes
2	1	121.740089	25.132571	海景門市	200基隆市仁愛區港西街6之2號	02-2428-4223	Y	
3	2	121.51767	25.047029	新世界門市	100台北市中正區忠孝西路一段49號B1	02-2371-7092		
4	3	121.509098	25.041583	寶中門市	100台北市中正區寶慶路32號	02-2331-0388	Y	
5	4	121.531439	25.017797	羅斯福門市	106台北市大安區羅斯福路三段301號	02-2364-6845	Y	
6	5	121.514167	25.046001	懷寧門市	100台北市中正區懷寧街10號	02-2389-3776	Y	
7	6	121.57694	25.062673	舊宗門市	114台北市內湖區舊宗路一段188號一樓	02-8792-8408	Y	
8	7	121.563588	25.041256	聯合門市	110台北市信義區忠孝東路四段561號1樓	02-2749-5657	Y	
9	8	121.548296	25.051753	環亞門市	105台北市松山區南京東路三段337號1F	02-8712-0625	Y	
10	9	121.566807	25.082169	環山門市	114台北市內湖區內湖路一段289號	02-2799-5897	N	
11	10	121.540795	25.06212	龍權門市	104台北市中山區民權東路三段22號	02-2517-9259	Y	
12	11	121.548336	25.041369	龍門門市	106台北市大安區忠孝東路四段134號	02-2740-6782	Y	
13	12	121.543721	25.03332	興南門市	106台北市大安區復興南路一段323號1樓	02-2325-9473	Y	
14	13	121.544467	25.053455	興北門市	105台北市松山區復興北路147號1、2F	02-2712-8189	Y	
15	14	121.557886	25.048184	德復門市	105台北市松山區八德路三段251號1-2F	2578-8732	Y	
16	15	121.507392	25.043915	漢中門市	108台北市萬華區漢中街51號1-3樓	02-2370-5893	Y	
17	16	121.546205	25.083801	實業門市	104台北市中山區大直街70號1樓	02-2532-5223	Y	
18	17	121.553019	25.041263	鼎豐門市	106台北市大安區忠孝東路四段218號一樓	2773-7207	Y	
19	18	121.557309	24.999556	萬芳門市	116台北市文山區興隆路三段111號	02-2933-6950	Y	
20	19	121.569521	25.078441	瑞傳門市	114台北市內湖區瑞光路476號	02-8798-3266	Y	
21	20	121.572749	25.077284	瑞光門市	114台北市內湖區瑞光路356號	02-2659-3912	Y	
22	21	121.567366	25.035954	新光門市	110台北市信義區松壽路11號1樓	02-8789-4099	N	
23	22	121.527607	25.118405	新天母門市	111台北市士林區天母西路33號1樓	02-2876-9351	N	
24	23	121.547395	25.152201	陽明山門市	112台北市北投區湖山路一段14號之1 1樓	02-2861-0399	Y	
25	24	121.556934	25.042442	華南門市	106台北市大安區光復南路180巷14號	02-8771-9692	Y	
26	25	121.575919	25.075292	港嶼門市	114台北市內湖區瑞光路216號	02-8753-5105	Y	

stores_old.csv

	A	B	C	D	E	F	G	H	I
1	sid	name	address	wifi					
2	1	海景門市	200基隆市仁愛區港西街6之2號	Y					
3	2	新世界門市	100台北市中正區忠孝西路一段49號B1						
4	3	寶中門市	100台北市中正區寶慶路32號	Y					
5	4	羅斯福門市	106台北市大安區羅斯福路三段301號	Y					
6	5	懷寧門市	100台北市中正區懷寧街10號	Y					
7	6	舊宗門市	114台北市內湖區舊宗路一段188號一樓	Y					
8	7	聯合門市	110台北市信義區忠孝東路四段561號1樓	Y					
9	8	環亞門市	105台北市松山區南京東路三段337號1F	Y					
10	9	環山門市	114台北市內湖區內湖路一段289號	N					
11	10	龍權門市	104台北市中山區民權東路三段22號	Y					
12	11	龍門門市	106台北市大安區忠孝東路四段134號	Y					
13	12	興南門市	106台北市大安區復興南路一段323號1樓	Y					
14	13	興北門市	105台北市松山區復興北路147號1、2F	Y					
15	14	德復門市	105台北市松山區八德路三段251號1-2F	Y					
16	15	漢中門市	108台北市萬華區漢中街51號1~3樓	Y					
17	16	實業門市	104台北市中山區大直街70號1樓	Y					
18	17	鼎豐門市	106台北市大安區忠孝東路四段218號一樓	Y					
19	18	萬芳門市	116台北市文山區興隆路三段111號	Y					
20	19	瑞傳門市	114台北市內湖區瑞光路476號	Y					
21	20	瑞光門市	114台北市內湖區瑞光路356號	Y					
22	21	新光門市	110台北市信義區松壽路11號1樓	N					
23	22	新天母門市	111台北市士林區天母西路33號1樓	N					
24	23	陽明山門市	112台北市北投區湖山路一段14號之1 1樓	Y					
25	24	華南門市	106台北市大安區光復南路180巷14號	Y					
26	25	港嶼門市	114台北市內湖區瑞光路216號	Y					

stores_new.csv