

1.17 例外處理進階 try-except (額外教材)

你將學習到

1. 捕捉某個種類例外
2. 取得例外種類名稱
3. 用 try-except-else-finally 處理例外
4. 手動拋出例外 raise Exception()

捕捉某個種類的例外

```
try:
    1/0
    1+"a"
    b
except ZeroDivisionError as e:      # 只捕捉 ZeroDivisionError
    print( "ZeroDivisionError")
except TypeError as e:              # 只捕捉 TypeError
    print( "Type Error" )
except Exception as e:              # 捕捉所有 Exception
    print( "Exception" )
```

我們可以寫多於一個 except-block, 而且每個 except-block 只捕捉某個例外類型。

這樣我們就可以根據不同的例外類型, 做不同的錯誤處理動作。

至於如何知道例外類型的名稱, 請見下頁。

取得例外類型名稱

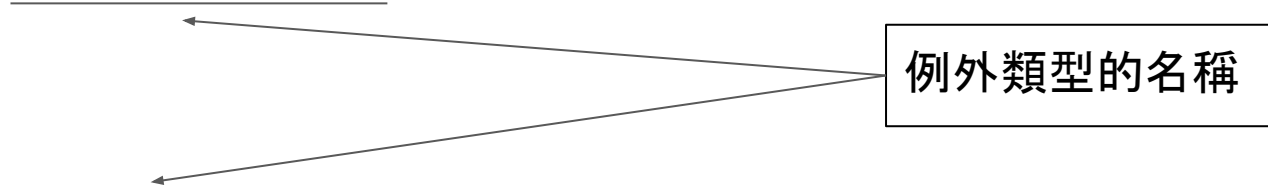
你可以直接在 Console 輸入會發生錯誤的語句，然後顯示的錯誤訊息內容就包含了例外類型的名稱。

```
1/0
```

```
# 輸出 "ZeroDivisionError: division by zero"
```

```
1 + "a"
```

```
# 輸出 "TypeError: unsupported operand type(s) for +: 'int' and 'str'"
```



例外類型的名稱

try - except - else - finally

我們還可為 try-except 加上 else 或 finally。如果 try 發生錯誤, except 會被執行。如果 try 沒有發生錯誤, else 會被執行。而不論 try 有無錯誤, finally 都會被執行。
finally 一般會放入如儲存檔案等即使程式錯誤也必須執行的動作。

try:

```
print( "計算中..." )
```

```
calculation()
```

calculation() 可能會發生錯誤

except:

```
print( "計算失敗" )
```

如果 try 發生錯誤, except 會被執行

else:

```
print( "計算成功" )
```

如果 try 沒有發生錯誤, else 會被執行

finally:

```
print( "結束計算" )
```

不論 try 有無錯誤, finally 都會被執行

try - except - else - finally

```
try:  
    print( "計算中..." )  
    calculation()  
except:  
    print( "計算失敗" )  
else:  
    print( "計算成功" )  
finally:  
    print( "結束計算" )
```

def calculation(): a = 1/5	def calculation(): a = 1/0
計算中...	計算中...
計算成功	計算失敗
結束計算	結束計算

手動拋出例外 raise Exception()

你可以自定義例外，例如以下例子，當輸入數字大於 2 時會發生例外。

```
try:
    num = input("輸入數字:")
    num = int(num)
    if num > 2:
        raise Exception("數字必須小於或等於2")
        # 拋出例外後，函數會立刻跳出，之後的語句不會執行
    else:
        print(1 / num)

    print("這句只有在沒有執行 raise Exception時，才會被執行")

except Exception as e:
    print(e)
```

練習

1. 定義函數 `divide(a, b)` 返回 `a/b` 的結果, 如遇上 `b = 0`, 須拋出例外, 錯誤訊息為 "Division by zero!"。
2. 撰寫主程式, 運行 `divide(1,0)`, 捕捉例外並顯示錯誤訊息。

答案

```
def divide(a, b):  
    if b == 0:  
        raise Exception("Division by zero!")  
    return a/b;  
  
try:  
    divide(1, 0)  
except Exception as e:  
    print(e)
```

學習回顧

1. 捕捉某個種類例外
2. 取得例外種類名稱
3. 用 try-except-else-finally 處理例外
4. 手動拋出例外 raise Exception()