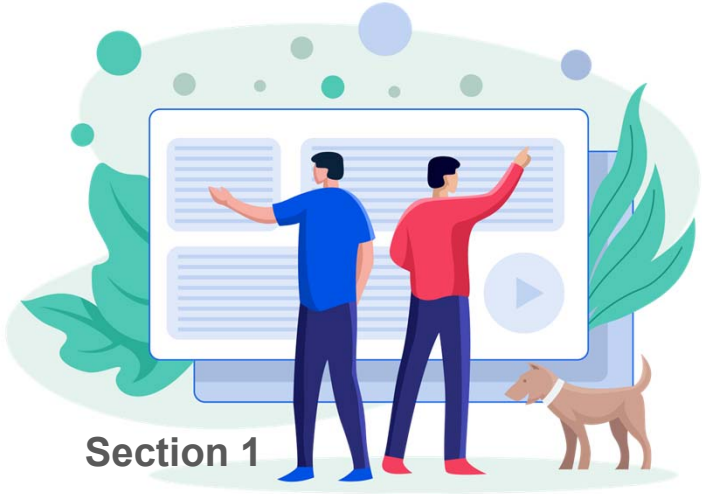
異常處理

學習目標

- Python程式例外(Exception) 流程
- 攔截程式例外
- else，finally 的用法

異常處理

- Python 的異常(Exceptions) 機制
- 如何處理異常



錯誤(Errors)與異常(Exceptions)

- Python 最常見的錯誤有兩種，語法錯誤(Syntax errors) 與異常(Exceptions)
- 語法錯誤
  - 又稱為剖析錯誤(Parsing errors)，這是初學 Python 語言時最常發生的錯誤
  - >>> while True print('Hello world')
    File "<stdin>", line 1
      while True print('Hello world')
                 ^
    SyntaxError: invalid syntax

異常(Exceptions)

- 異常是指 Python 程式在執行時發生的錯誤
- Python 有內建一些異常，可參考 Python 文件
  - https://docs.python.org/3/library/exceptions.html
- 即使語法是正確的，但是在執行時仍可能會引發錯誤。這種在執行時期會發生的錯誤稱為異常。異常不一定會導致嚴重的結果
- >>> 10 * (1/0)
  Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
  ZeroDivisionError: division by zero

處理異常 -1

- 使用 try/except 區塊來攔截錯誤與復原程式
- 語法
  - try:
    stmt
    ...
  except Exception1:
    If there is Exception1 then execute this block
  except Exception2:
    If there is Exception2, then execute this block
    ...
  else:
    If there is no exception then execute this block
  finally:
    This would always be executed

處理異常 -2

- 介於 try 與 except 之間的程式開始執行
- 如果沒有錯誤發生，則略過 except 子句，try 敘述結束
- 如果有錯誤發生，則 try 子句內的其餘程式會略過
  - 接著開始評估異常區塊
  - 如果錯誤符合異常類型，則執行該 except 子句，然後繼續 try-except 之後的程式
  - 如果錯誤不符合任一個異常類型，則該錯誤會往外層的 try 敘述送
  - 如果外層都沒有錯誤處理機制存在，則程式停止執行，並顯示錯誤訊息

處理異常 -3

- try 敘述可以有一或多個 except 子句，來處理不同情況的異常 (exceptions)
- try 配對的多個 except 區塊只能有一個 except 子句執行
- except 子句有先後順序，以第一個符合的來處理
- 一個 except 子句可以處理多個類型的異常，使用 tuple 方式表達
  - except (RuntimeError, TypeError, NameError):
    pass

異常的引數

- 異常(Exception) 可以接一個物件，提供錯誤相關的資訊
- 物件內容會依異常類型而有不同
- 範例

```
try:
    f = open('myfile.txt')
    s = f.readline()
    i = int(s.strip())
except OSError as err:
    print('OS error: {0}'.format(err))
```

異常處理語法

- try/except 子句格式

子句	說明
except:	攔截所有異常類型
except name:	只攔截特定異常
except name as e:	攔截特定異常，配合錯誤物件
except (name1, name2):	攔截多個異常
except (name1, name2) as e:	攔截多個異常，配合錯誤物件
else:	沒有異常發生時執行
finally:	不管有沒有發生異常，總是會執行的區塊

引發(raise)異常 -1

- 使用 “raise” 敘述可以強制引發一個異常(exception)
- 通常用來重新引發異常，讓呼叫者(Caller) 也可以處理相同的異常

```
try:
    f = open('myfile.txt')
    s = f.readline()
    i = int(s.strip())
except ValueError:
    print('Could not convert data to an integer.')
except:
    print('Unexpected error:')
    raise
```

引發(raise)異常 -2

- 也可以用來引發其他異常或使用者客製化的異常
- try:
```
    lst1 = [1,2,3,4]
    idx = int(input('Enter an index: '))
    if not 0<=idx<4:
        raise IndexError('Index out of range')
    print('Value is', lst1[idx])
except IndexError as e:
    print('Index Error :', e)
```
- 執行結果
  - Index Error : Index out of range

異常處理(Demo)

- 介紹 Python 異常處理方式

本章重點精華回顧

- Python Exception 的流程
- 如何攔截(try) 或觸發(raise) 程式異常

Lab: 異常處理

- Lab01: 處理 Python 程式異常

Lab01: 處理Python程式異常

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “except1.py” 程式，了解異常處理的基本格式
3. 關閉except1.py程式視窗
4. 使用 “File/Open...” 開啟 “except2.py” 程式，了解else與finally的用法
5. 關閉except2.py程式視窗
6. 使用 “File/Open...” 開啟 “except3.py” 程式，了解異常物件的用法
7. 關閉except3.py程式視窗
8. 使用 “File/Open...” 開啟 “except4.py” 程式，了解raise的用法
9. 關閉except4.py程式視窗