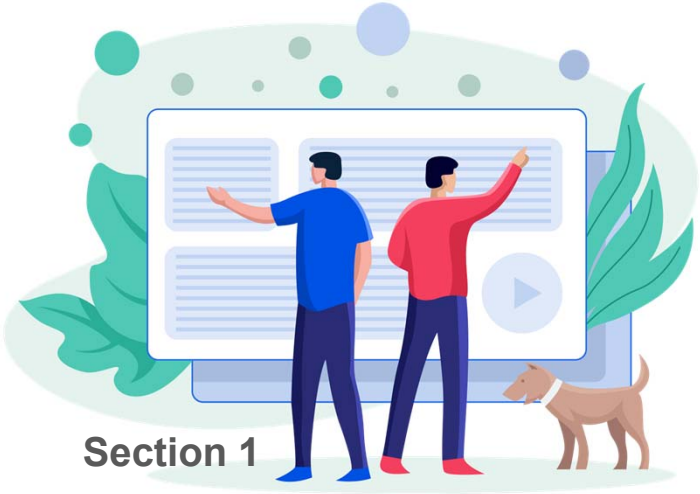
檔案的讀寫與組織管理

學習目標

- Python 內建函數 open() 的用法
- read()，write() 的用法
- with 敘述的應用

基本檔案輸出輸入

- 瞭解 open() 函數的用法
- 如何讀取與寫入資料到檔案



標準輸出

- 要將資料輸出最簡單的方式就是使用 print() 函數，可以傳入零或多個引數，以逗點隔開
- print() 函數會將資料轉成文字格式，然後寫入標準輸出(Standard output，sys.stdout)，預設為螢幕
  - print('Python is funny, try it!')
  - ```
    n1=33
    n2=66
    print('Sum of',n1,'and',n2,'=',n1+n2)
    ```
    ```

標準輸入

- Python 內建 input() 函數可以接受使用者輸入的資料
- input() 函數會從標準輸入(Standard input，sys.stdin) 讀取資料，預設為鍵盤
- eval() 內建函數可以將輸入的資料做運算，傳回運算結果
- >>> print(input('What is your name? '))
  What is your name? John
  John
  >>> print(eval(input('Do some math: ')))
  Do some math: 12+20*7
  152

開啟檔案

- Python 預設提供基本的函數(Function) 與方法(Methods) 來維護檔案
- 使用 open() 函數
  - f = open(file_name [, access_mode][, buffering])
  - 第一個引數為檔案名稱
  - 第二個引數為開檔模式
  - 假如 buffering 值為 0，則關閉 buffer 功能。假如為 1，則使用 line buffering
    - 預設為 -1，使用系統預設(Linux 為 1024 (20GB內的硬碟容量)或 4096 (20GB以上的硬碟容量) bytes)

檔案存取模式

- 預設模式為 'rt'　(讀取文字檔)

模式	說明
'r'	開檔讀資料 (預設)
'w'	開檔寫資料，會先清空檔案
'x'	開檔寫資料，如果檔案存在會失敗
'a'	開檔寫資料，如果檔案存在會附加在檔案結尾
'b'	以二進位模式開啟
't'	以文字模式開啟 (預設)
'+'	開啟檔案做更新 (可以讀寫資料)

關閉檔案

- file.close() 方法
  - 寫入快取的資料然後關閉檔案
  - 當檔案的參考指向另一個檔案時，Python 會自動關閉原檔案
- file.name 屬性
  - 取得檔案名稱
- 範例
  - ```
    fo = open('foo.txt', 'w')
    print('Name of the file: ', fo.name)
    # Close opened file
    fo.close()
    ```
    ```

寫入檔案

- file.write(string) 方法
  - 將 string 寫入檔案
  - string 可以是二進位資料，不一定只是文字資料
- write() 不會加換行字元('\n') 到字串結尾
- 範例
  - ```
    fo = open('foo.txt', 'w')
    fo.write('Python is a great language!!\n')
    # Close opend file
    fo.close()
    ```
    ```

讀取檔案 -1

- file.read(size) 方法
  - 從檔案讀資料，傳回一個字串
  - 讀取 size 個資料，如果 size 省略或是負數，則讀入整個檔案
  - 傳回的字串可以是二進位資料，不一定只是文字資料
- 範例
  - ```
    fo = open('foo.txt', 'r')
    str = fo.read(10)
    print('Read String is : ', str)  # 'Python is '
    # Close opened file
    fo.close()
    ```
    ```

讀取檔案 -2

- file.readline() 方法
  - 一次讀取一列資料，會保留換行字元(\n) 在字串結尾
  - 如果 readline() 傳回空字串，代表檔案結尾(EOF)
- 範例
  -
```
fo = open('foo.txt', 'r')
while True:
    line=fo.readline()
    if not line:
        break
    print(line, end='')
fo.close()
```

with 敘述 -1

- 還沒有使用 with 敘述時，程式必須處理異常與自行關閉檔案
- 範例

```
try:
    fp = open('note.txt', 'r')
    data = fp.read()
    print('Content:')
    print(data)
except:
    print('Error: File I/O error! ')
finally:
    fp.close()
```

with 敘述 -2

- 使用 “with” 敘述，可以得到更好的語法以及異常處理方式
- with 敘述簡化了異常處理，它封裝了異常的準備與清理工作
- 範例
  - with open('note.txt', 'r') as fp:
    data = fp.read()
    print('Content:')
    print(data)

基本檔案輸出輸入(Demo)

- 使用 open() 存取檔案

本章重點精華回顧

- Python 基本檔案輸出輸入
- with 敘述的應用

Lab: 檔案的讀寫與組織管理

- Lab01: Python 基本檔案輸出輸入

Lab01: Python基本檔案輸出輸入

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “ex19_1.py” 程式，了解寫入檔案的方式
3. 檢視note.txt檔案內容
4. 關閉ex19_1.py程式視窗
5. 使用 “File/Open...” 開啟 “ex19_2.py” 程式，了解讀取檔案的方式
6. 關閉ex19_2.py程式視窗
7. 使用 “File/Open...” 開啟 “withstmt1.py” 程式，了解檔案異常處理與with敘述的用法
8. 關閉withstmt1.py程式視窗
9. 使用 “File/Open...” 開啟 “withstmt2.py” 程式，了解with敘述簡化檔案處理流程
10. 關閉withstmt2.py程式視窗