

基礎檔案存取

File Processing



Python Fundamental



基本輸入輸出 - input()

- ◆ 在程式執行的過程中，可以使用 `input()` 函數取得使用者的輸入。
- ◆ `input()` 函數可以指定提示文字，使用者輸入的文字則以字串傳回。

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

請輸入你的名稱：張三
歡迎 張三



基本輸入輸出 - print()

- ◆ 基本輸出語法：

```
print(value, ..., sep=' ', end='\n', file=sys.stdout)
```

- ◆ 除了指定值 (value) 輸出之外，還可使用sep與end 。
- ◆ sep 指定每個輸出值之間的分隔字元，預設值為一個空格 。
- ◆ end 指定輸出後最後一個字元，預設值是'\n' 。



輸出至檔案

- ◆ 預設輸出是系統標準輸出，可以使用 `file` 指定其它的輸出。
- ◆ 以下的例子會將指定的值輸出至 `data.txt` 。

```
# print 輸出至檔案範例

text = ""將此文的內容全部存起來

1. A

2. B

3. C"""

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



open() 函數

- ◆ 將資料寫入檔案或從檔案讀出，可以使用 `open()` 函數以及兩個參數：`file` 與 `mode`。

`open(file, mode='r')`

字串	說明
r	讀取模式（預設）。
w	寫入模式，會先清空檔案內容。
x	只在檔案不存在時才建立新檔案並開啟為寫入模式，若檔案已存在會引發 <code>FileExistsError</code> 。
a	附加模式，若檔案已存在，寫入的內容會附加至檔案尾端。



寫入檔案

◆ 建立檔案物件

- 我們可以使用open() 函數建立檔案物件。建立成功時會傳回檔案物件；建立失敗時則會發生錯誤。

open(file, mode)

- 範例：

```
file = open('C:\\Users\\Alex\\Desktop\\test.txt', 'r') → 使用跳脫字元
```

- 這個敘述亦可寫成如下：

```
file = open(r'C:\\temp\\test.txt', 'r') → 不用跳脫字元 r → 「raw string」
```

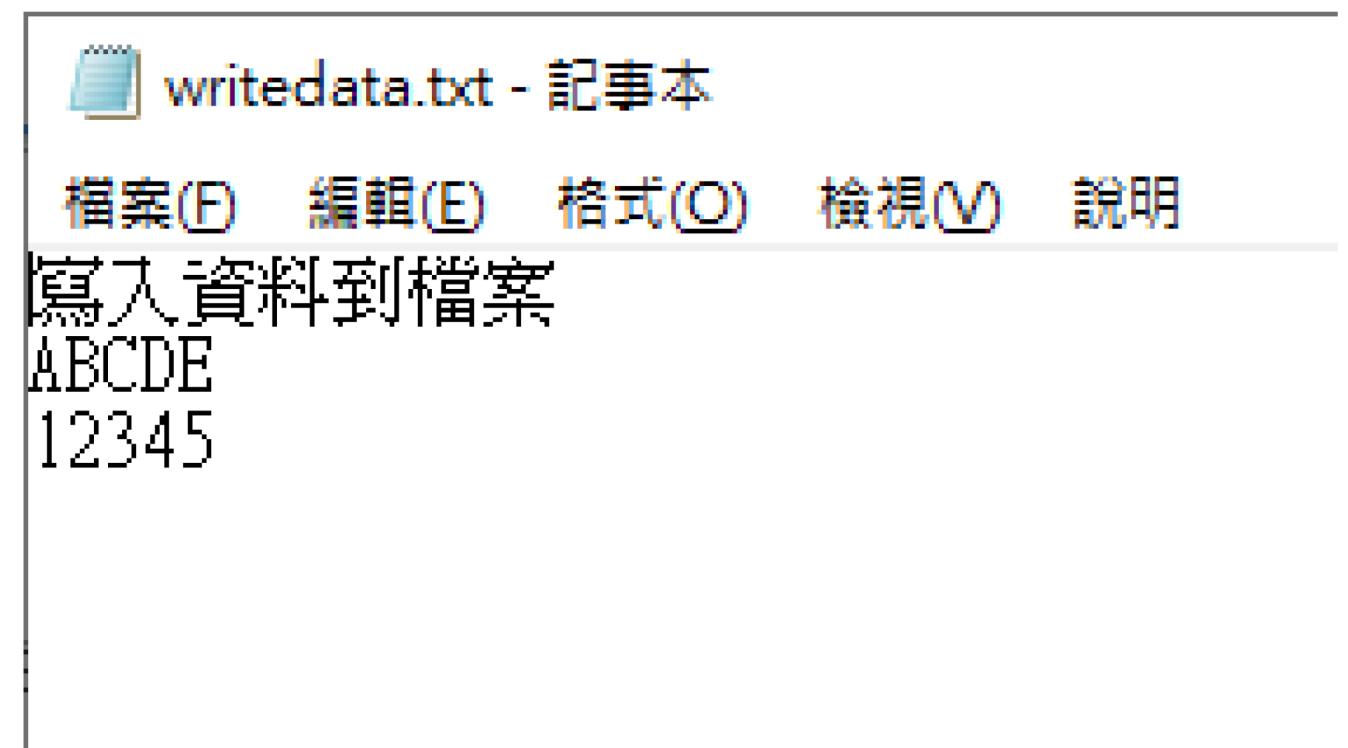


寫入檔案

◆ 將資料寫入檔案的步驟如下：

1. 開啟檔案：使用open()函數建立檔案物件。
2. 寫入資料：使用檔案物件提供的write(s)函數將參數s所指定的字串寫入檔案。
3. 關閉檔案：使用檔案物件提供的close()函數關閉檔案。

```
file = open('writedata.txt', 'w')
file.write('寫入資料到檔案\n')
file.write('ABCDE\n')
file.write('12345')
file.close()
```





讀取檔案

- ◆ 從檔案讀取資料的步驟如下：

1. 開啟檔案：使用 `open()` 函數建立檔案物件。
2. 讀取資料：使用檔案物件提供的 `read()` 函數讀取資料。
3. 關閉檔案：使用檔案物件提供的 `close()` 函數關閉檔案。

- ◆ 開啟檔案後，我們可以使用檔案物件提供的 `read()` 函數讀取檔案內容，之後使用 `close()` 函數將檔案關閉。

```
file = open('writedata.txt', 'r')
content = file.read()
print(content)
file.close()
```

寫入資料到檔案
ABCDE
12345



with 敘述

- ◆ with敘述可將存取檔案的動作包裝在一個區塊。當區塊裡面的程式執行完畢時，就會自動關閉檔案。

```
with open(file, mode) as 檔案物件名稱 :  
    .....  
    # 存取檔案的動作
```

```
with open("poem.txt", "r") as file :  
  
    content = file.read()  
  
    print(content)
```

The world puts off its mask of vastness to its lover.
It becomes small as one song, as one kiss of the eternal

Q & A