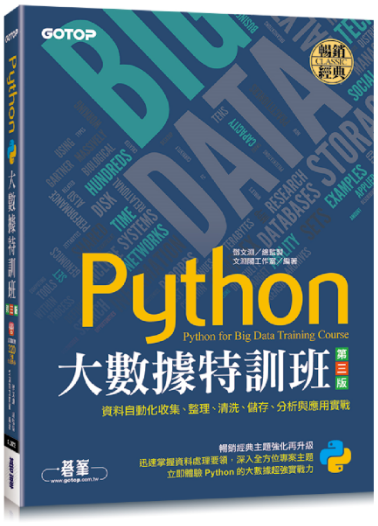


碁峰資訊

CHAPTER

03

數據資料的儲存與讀取

碁峰資訊

GOTOP

Python 大數據特訓班

3.1 檔案的讀寫

3.1.1 檔案的建立與寫入

使用內建的函式open 可以開啟指定的檔案，包括文字檔案和二進位檔案，以便進行檔案內容的讀取與寫入。

open() 函式

```
open( 檔案名稱 [, 模式 ][,encoding= 編碼 ])
```

- **檔案名稱**：設定檔案的名稱，它是字串型態，可以是相對路徑或絕對路徑，如果沒有設定路徑，則會預設為目前執行程式的目錄。

GOTOP

Python 大數據特訓班

- **模式**：設定檔案開啟的模式，是字串型態，省略預設為 r 讀取模式。模式設定時再加上t 表示為文字檔案，b 是二進位檔案，如果省略時預設為t。

模式	說明	模式	說明
r	讀取模式，此為預設模式。	r+	可讀寫模式，指標會置於檔頭。
w	覆寫模式，若檔案已存在，內容將會被覆蓋。	w+	可讀寫模式，指定檔案不存在時會建立檔案再寫入檔案，若檔案已存在，寫入內容會覆蓋原內容。
a	附加模式，若檔案已存在，內容會被附加至檔案尾端。	a+	可讀寫模式，指定檔案不存在時會建立檔案再寫入檔案，若檔案已存在，寫入內容會附加至檔案尾端。

- **encoding**：指定檔案的編碼模式。

GOTOP Python 大數據特訓班

檔案的寫入

open() 函式會建立一個檔案物件，利用這個物件就可以處理檔案，例如要寫入資料時可以使用write() 函數，最後當檔案處理結束必須以close() 函式關閉檔案。例如：

```
[2]  1  content='''Hello Python
     2  中文字測試
     3  Welcome'''
     4  f=open('file1.txt', 'w' ,encoding='utf-8')
     5  f.write(content)
     6  f.close()
```

GOTOP Python 大數據特訓班

使用with 敘述開啟檔案

檔案的開啟也可以使用with 敘述，因為敘述結束後會自動關閉檔案，不需要再以close() 關閉檔案。例如用with 敘述的方式改寫剛才的程式碼：

```
[4]  1  content='''Hello Python
     2  中文字測試
     3  Welcome'''
     4  with open('file1.txt', 'w' ,encoding='utf-8') as f:
     5    f.write(content)
```

GOTOP Python 大數據特訓班

3.1.2 檔案讀取及處理

函式	說明
close()	關閉檔案，檔案關閉後就不能再進行讀寫的操作。
flush()	檔案在關閉時會將資料寫入檔案中，也可以使用 flush() 強迫將緩衝區的資料立即寫入檔案中，並清除緩衝區。
read([size])	由目前位置讀取 size 長度的字元，並將目前位置往後移動 size 個字元。如果未指定長度則會讀取所有字元。
readable()	測試是否可讀取。
readline([size])	讀取目前文字指標所在列中 size 長度的文字內容，若省略參數，則會讀取一整列，包括 "\n" 字元。
readlines()	讀取所有列，它會傳回一個串列。
next()	移動到下一列。
seek()	將指標移到文件指定的位置。
tell()	傳回文件目前位置。
write(str)	將指定的字串寫入文件中，它沒有返回值。
writelines(list)	將指定的串列寫入文件中，它沒有返回值。
writable()	測試是否可寫入。

GOTOP Python 大數據特訓班

read()

```
[5]  1  with open('file1.txt', 'r', encoding='utf-8') as f:
     2    output_str=f.read(5)
     3    print(output_str)   # Hello
```

readline()

讀取目前文字指標所在列中size 長度的文字內容

```
[7]  1  with open('file1.txt', 'r', encoding ='UTF-8') as f:
     2    print(f.readline())
     3    print(f.readline(3))
```

readlines()

讀取全部文件內容，它會以串列方式傳回，每一列會成為串列中的一個元素。

```
[8]  1  with open('file1.txt', 'r', encoding='utf-8') as f:
     2    content=f.readlines()
     3    print(type(content))
     4    print(content)
```

GOTOP Python 大數據特訓班

BOM 的處理

BOM(Byte Order Mark) 是用來標示文件編碼的標記。

例如：讀取UTF-8 編碼的<file2.txt> 檔案的文件內容。

```
[9]  1  with open('file2.txt', 'r', encoding ='UTF-8') as f:
     2    print(f.readlines())
```

使用open() 函式開啟含有BOM 的檔案時，可以設定「encoding='UTF-8-sig'」，如此會將BOM 與文件分離單獨處理。

例如：讀取UTF-8 編碼的<file2.txt> 檔案的文件內容，並分離BOM。

```
[11]  1  with open('file2.txt', 'r', encoding ='UTF-8-sig') as f:
      2    print(f.readlines())
```

GOTOP Python 大數據特訓班

3.2 csv 資料的讀取與寫入

3.2.1 認識CSV

CSV(Comma Separated Values) 是一種以符號分隔值的資料格式並以純文字的方式儲存為檔案，其中常用的符號為「,」。

3.2.2 csv 模組的使用

可以使用串列或字典資料類型，將資料寫入csv 檔案。而串列的寫入方式又分為：

1. csv 寫入物件.writerow()：寫入一維串列。
2. csv 寫入物件.writerows()：寫入二維串列。

GOTOP

Python 大數據特訓班

將一維串列資料寫入csv 檔案

```
[13]  1  import csv
      2  # 開啟輸出的 csv 檔案
      3  with open('test1.csv', 'w', newline='') as csvfile:
      4    # 建立 csv 檔寫入物件
      5    writer = csv.writer(csvfile)
      6
      7    # 寫入欄位名稱
      8    writer.writerow(['姓名', '身高', '體重'])
      9    # 寫入資料
     10    writer.writerow(['chiou', 170, 65])
     11    writer.writerow(['David', 183, 78])
```

將二維串列資料寫入csv 檔案

```
[18]  1  import csv
      2  # 建立csv二維串列資料
      3  csvtable = [
      4    ['姓名', '身高', '體重'],
      5    ['Chiou', 170, 65],
      6    ['David', 183, 78],
      7  ]
      8  # 開啟輸出的 csv 檔案
      9  with open('test2.csv', 'w', newline='') as csvfile:
     10    # 建立 csv 檔寫入物件
     11    writer = csv.writer(csvfile)
     12
     13    # 寫入二維串列資料
     14    writer.writerows(csvtable)
```

GOTOP

Python 大數據特訓班

將字典資料寫入csv 檔案

```
[3]   1  import csv
      2  with open('test.csv', 'w', newline='') as csvfile:
      3    # 定義欄位
      4    fieldnames = ['姓名', '身高', '體重']
      5
      6    # 將 dictionary 寫入 csv 檔
      7    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
      8
      9    # 寫入欄位名稱
     10    writer.writeheader()
     11    # 寫入資料
     12    writer.writerow({'姓名': 'chiou', '身高': 170, '體重': 65})
     13    writer.writerow({'姓名': 'David', '身高': 183, '體重': 78})
```

GOTOP Python 大數據特訓班

3.2.3 csv 檔案讀取

將csv 檔案中資料，讀取為串列或字典格式，方法如下：

1. 讀取為串列格式：csv.reader()。

2. 讀取為字典格式：csv.DictReader()。

讀取 csv 檔案為串列資料

```
[4]  1  import csv
     2  # 開啟 csv 檔案
     3  with open('test1.csv', newline='') as csvfile:
     4    # 讀取 csv 檔案內容
     5    rows = csv.reader(csvfile)
     6    # 以迴圈顯示每一列
     7    for row in rows:
     8      print(row)
```

GOTOP Python 大數據特訓班

讀取 csv 檔案為字典資料

```
[ ]  1  import csv
     2  # 開啟 csv 檔案
     3  with open('test1.csv', newline='') as csvfile:
     4    # 讀取 csv 檔內容，將每一列轉成 dictionary
     5    rows = csv.DictReader(csvfile)
     6    # 以迴圈顯示每一列
     7    for row in rows:
     8      print(row['姓名'],row['身高'],row['體重'])
```

GOTOP

Python 大數據特訓班

3.3 json 資料的儲存與讀取

3.3.1 認識json

JSON 是利用 **資料物件(object)** 及 **陣列(Array)** 的方式來描述資料結構與內容：

1. **資料物件**：是用來描述單筆資料，內容是使用「{...}」符號包含起來。一個物件中包含一系列非排序的鍵(名稱)/ 值對，鍵和值之間使用「：」隔開，多個鍵/值對之間使用「,」分割。
2. **清單陣列**：是用來描述多筆資料，內容是使用「[...]」符號包含起來。每筆資料之間使用「,」區
3. **資料物件(object):** `{ "key1": value1, "key2": value2, ... }`,
   陣列(Array): `[ object1, object2, ... ]`

回首頁

GOTOP

Python 大數據特訓班

3.3.2 json 模組的使用

函數	說明
json.load(檔案物件)	由 json 格式檔案載入為 json 資料。
json.loads(字串)	由 json 格式字串載入為 json 資料。
json.dump(字串 , 檔案物件)	將 json 資料寫入到檔案。
json.dumps(字串)	將 json 資料輸出為字串。

GOTOP

Python 大數據特訓班

3.3.3 json 讀取資料

讀取json 字串

程式碼：jsonload1.py

```
import json
class_str = """
{
  "一年甲班": [
    {
      "座號": 1,
      "姓名": "葉大雄",
      "國文": 65,
      "英文": 62,
      "數學": 40
    },
    {
      "座號": 2,
      "姓名": "陳靜香",
      "國文": 85,
      "英文": 90,
      "數學": 87
    },
    {
      "座號": 3,
      "姓名": "王聰明",
      "國文": 92,
      "英文": 90,
      "數學": 95
    }
  ]
}
"""
datas = json.loads(class_str)
print(type(datas))
for data in datas["一年甲班"]:
    print(data, data['姓名'])
```

GOTOP

Python 大數據特訓班

load 讀取json 檔案

以下的範例將把剛才程式中json 字串另存到<class_str.json> 中，請上傳<class_str.json> 檔到Colab 專案中，利用json.load() 讀取檔案中的資料：

```
[6]  1 import json
     2 with open('class_str.json', 'r', encoding='utf-8') as f:
     3   datas = json.load(f)
     4   print(type(datas))
     5   for data in datas["一年甲班"]:
     6     print(data, data['姓名'])
```

GOTOP

Python 大數據特訓班

3.3.4 json 輸出資料

python 程式中可以由字串或是檔案中輸出成為json 的資料內容。

dumps 輸出 json 字串

```
[9]  1  import json
     2  with open('class_str.json', 'r', encoding='utf-8') as f:
     3    datas = json.load(f)
     4  print(datas, type(datas))
     5  dumpdata = json.dumps(datas, ensure_ascii=False)
     6  print(dumpdata, type(dumpdata))
```

dump 輸出json 檔案

```
[16] 1  import json
     2  with open('class_str.json', 'r', encoding='utf-8') as f:
     3    datas = json.load(f)
     4  with open('new_class_str.json', 'w', encoding='utf-8') as f:
     5    json.dump(datas, f, ensure_ascii=False)
```

GOTOP

Python 大數據特訓班

3.4 Excel 資料儲存與讀取

3.4.1 Excel 檔案新增及儲存

openpyxl 模組新增及儲存的流程

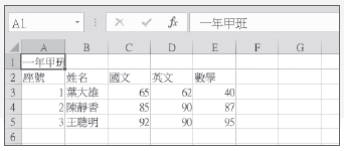
回首頁

GOTOP

Python 大數據特訓班

用openpyxl 模組儲存xlsx 檔

```
[1]  1  import openpyxl
     2  # 建立一個工作簿
     3  workbook=openpyxl.Workbook()
     4  # 取得第 1 個工作表
     5  sheet = workbook.worksheets[0]
     6  # 以儲存格位置寫入資料
     7  sheet['A1'] = '一年甲班'
     8  # 以串列寫入資料
     9  listtitle=['座號', '姓名', '國文', '英文', '數學']
    10  sheet.append(listtitle)
    11  listdatas=[[1, '葉大雄', 65, 62, 40],
    12             [2, '陳靜香', 85, 90, 87],
    13             [3, '王聰明', 92, 90, 95]]
    14  for listdata in listdatas:
    15      sheet.append(listdata)
    16  # 儲存檔案
    17  workbook.save('test.xlsx')
```



GOTOP

Python 大數據特訓班

3.4.2 Excel 檔案讀取及編輯

openpyxl 模組讀取檔案及編輯的流程

載入模組

```
import openpyxl
```

開啟工作簿

活頁簿物件

```
workbook=openpyxl.load_workbook(excel 檔案 )
```

用 load_workbook(檔案) 開啟活頁簿物件

取得工作表

工作表物件

```
sheet=workbook.worksheets[0]
```

用 worksheets[] 依索引值取工作表

取得資料

```
❶ sheet.max_row        取得總列數
❷ sheet.max_column     取得總欄數
❸ sheet.cell(row, column)  取得指定欄列中的值
```

儲存工作簿

```
workbook.save('file.xlsx')
```

用 save() 儲存檔案

GOTOP

Python 大數據特訓班

用openpyxl 模組讀取xlsx 檔

```
1  import openpyxl
2  #  讀取檔案
3  workbook = openpyxl.load_workbook('test.xlsx')
4  # 取得第 1 個工作表
5  sheet = workbook.worksheets[0]
6  # 取得指定儲存格
7  print(sheet['A1'], sheet['A1'].value)
8  # 取得總行、列數
9  print(sheet.max_row, sheet.max_column)
10 # 顯示 cell資料
11 for i in range(1, sheet.max_row+1):
12     for j in range(1, sheet.max_column+1):
13         print(sheet.cell(row=i, column=j).value,end="    ")
14     print()
15 sheet['A1'] = '二年甲班'
16 workbook.save('test.xlsx')
```

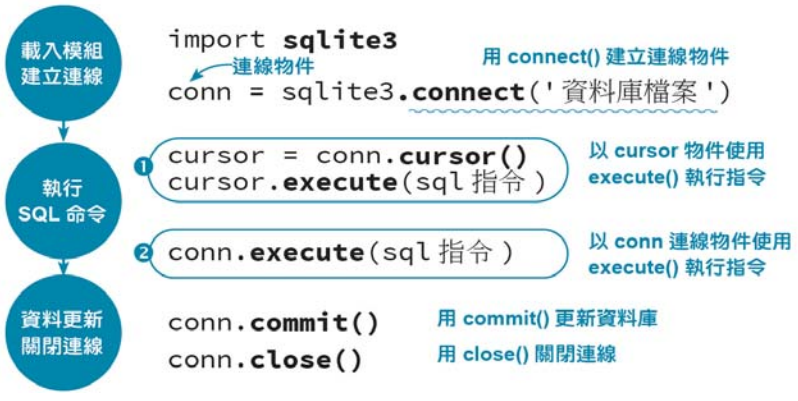
GOTOP

Python 大數據特訓班

3.5 SQLite 資料庫的操作

3.5.1 使用 sqlite3 模組

以下是使用sqlite3 模組來操作SQLite 資料庫的流程:



GOTOP

Python 大數據特訓班

3.5.2 使用 cursor 物件操作資料庫

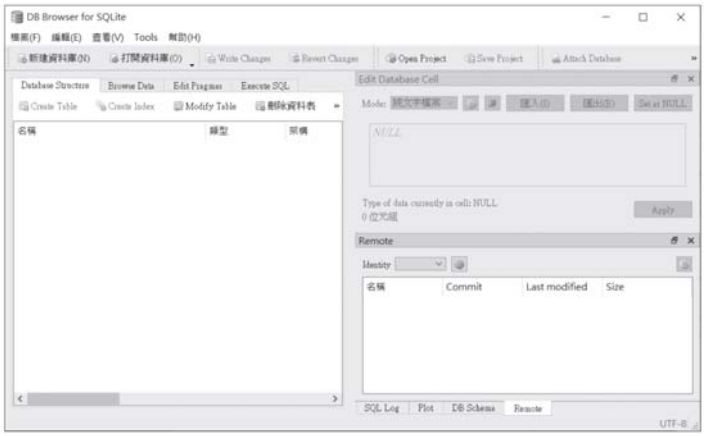
```
[1]  1  import sqlite3
     2  conn = sqlite3.connect('school.db') # 建立資料庫連線
     3  cursor = conn.cursor() # 建立 cursor 物件
     4  # 建立一個資料表
     5  sqlstr='''CREATE TABLE IF NOT EXISTS scores \
     6  ("id"  INTEGER PRIMARY KEY NOT NULL,
     7   "name"  TEXT NOT NULL,
     8   "chinese"  INTEGER NOT NULL,
     9   "english"  INTEGER NOT NULL,
    10   "math"  INTEGER NOT NULL
    11   )
    12  '''
    13  cursor.execute(sqlstr)
    14
    15  # 新增記錄
    16  cursor.execute('insert into scores values(1, "葉大雄", 65, 62, 40)
    17  cursor.execute('insert into scores values(2, "陳靜香", 85, 90, 87)
    18  cursor.execute('insert into scores values(3, "王聰明", 92, 90, 95)
    19  conn.commit() # 更新
    20  conn.close()  # 關閉資料庫連線
```

GOTOP

Python 大數據特訓班

3.5.3 檢視SQLite 資料庫內容

SQLite 是檔案型的資料庫，若要直接檢視內容其實不大容易。建議學習時可以安裝 DB Browser for SQLite (https://sqlitebrowser.org/) 來協助，它是一個很好用的SQLite 圖形化介面的管理工具。



GOTOP Python 大數據特訓班

3.5.4 使用連線物件操作資料庫

```
import sqlite3
conn = sqlite3.connect('資料庫檔案') # 建立資料庫連線
conn.execute(SQL 命令)
```

新增資料表

```
[2]
1  import sqlite3
2  conn = sqlite3.connect('school.db') # 建立資料庫連線
3  # 建立一個資料表
4  sqlstr='''CREATE TABLE IF NOT EXISTS scores2 \
5  ("id"  INTEGER PRIMARY KEY NOT NULL,
6   "name"  TEXT NOT NULL,
7   "chinese"  INTEGER NOT NULL,
8   "english"  INTEGER NOT NULL,
9   "math"  INTEGER NOT NULL
10  )
11  '''
12 conn.execute(sqlstr)
13 conn.commit() # 更新
14 conn.close()  # 關閉資料庫連線
```

GOTOP Python 大數據特訓班

新增資料

新增資料的SQL 命令語法為：

```
insert 資料表 (欄位1, 欄位2, ...) VALUES (值1, 值2, ...)
```

```
[13]
1  import sqlite3
2  conn = sqlite3.connect('school.db') # 建立資料庫連線
3  # 定義資料串列
4  datas = [[1,'葉大雄',65,62,40],
5           [2,'陳靜香',85,90,87],
6           [3,'王聰明',92,90,95]]
7
8  # 新增資料
9  for data in datas:
10   conn.execute("INSERT INTO scores2 (id, name, chinese, english,\
11    math) VALUES ({}, '{}', {}, {}, {})".format(data[0], \
12    data[1], data[2], data[3], data[4]))
13 conn.commit() # 更新
14 conn.close()  # 關閉資料庫連線
```

GOTOP Python 大數據特訓班

更新資料

更新資料的SQL 命令語法為：

```
update 資料表 set 欄位1= 值1, 欄位2= 值2 ... where 條件式
```

```
[14]  1  import sqlite3
      2  conn = sqlite3.connect('school.db') # 建立資料庫連線
      3  # 更新資料
      4  conn.execute("UPDATE scores2 SET name='{}' WHERE id={}".format
      5  ('林胖虎', 1))
      6  conn.commit() # 更新
      7  conn.close()  # 關閉資料庫連線
```

GOTOP Python 大數據特訓班

刪除資料

刪除資料的SQL 命令語法為：

```
delete from 資料表 where 條件式
```

```
[15]  1  import sqlite3
      2  conn = sqlite3.connect('school.db') # 建立資料庫連線
      3  # 刪除資料
      4  conn.execute("DELETE FROM scores2 WHERE id={}".format(1))
      5  conn.commit() # 更新
      6  conn.close()  # 關閉資料庫連線
```

刪除資料表與關閉資料庫

刪除整個資料表的語法為：

```
drop 資料表
```

```
[23]  1  conn = sqlite3.connect('school.db') # 建立資料庫連線
      2  conn.execute("DROP TABLE scores2")
      3  conn.close()  # 關閉資料庫連線
```

GOTOP Python 大數據特訓班

3.5.5 執行資料查詢

方法	說明
fetchall()	以串列格式回傳所有符合查詢條件的資料，若無資料傳回 None。
fetchone()	以元組格式回傳符合查詢條件的第一筆資料，若無資料傳回 None。

例如：以fetchall() 顯示scores 資料表所有的資料：

```
[18]  1  import sqlite3
      2  conn = sqlite3.connect('school.db') # 建立資料庫連線
      3  cursor = conn.execute('select * from scores')
      4  rows = cursor.fetchall()
      5  # 顯示原始資料
      6  print(rows)
      7  # 逐筆顯示資料
      8  for row in rows:
      9    print(row[0],row[1])
     10  conn.close()  # 關閉資料庫連線
```

GOTOP Python 大數據特訓班

例如：以fetchone() 顯示scores 資料表中第一筆資料：

```
[19]  1  import sqlite3
      2  conn = sqlite3.connect('school.db') # 建立資料庫連線
      3  cursor = conn.execute('select * from scores')
      4  row = cursor.fetchone()
      5  print(row[0], row[1])
      6  conn.close()  # 關閉資料庫連線
```

GOTOP Python 大數據特訓班

3.6 Google 試算表的操作

3.6.1 連接Google 試算表前的注意事項

使用Google 免費試算表時要注意以下幾點：

1. 一天最多只能建立250 個試算表。
2. 每個使用者100 秒內能寫入次數上限是100 次。
3. 每日讀取寫入的次數沒有限制。

要使用Python 將資料儲存到Google 試算表，必須有以下的條件：

1. 建立Google 應用程式授權憑證：在**Google Developers Console** 建立專案，啟用**Google SheetAPI**，並且建立 **服務帳戶** 和 **服務帳戶金鑰**。
2. 建立Google 試算表並設定權限給程式操作。
3. Python 要安裝gspread、oauth2client 模組。

回首頁

GOTOP Python 大數據特訓班

3.6.2 Google Developers Console 的設定

1. 由「https://console.developers.google.com」進入頁面，按下拉式選單鈕開啟專案管理視窗，在 **選取專案** 視窗中按 **新增專案** 鈕，專案名稱欄位輸入自訂的名稱後按 **建立** 鈕。
2. 選取剛剛建立的專案， 按 **啟用API 服務**鈕， 然後在搜尋欄位輸入「GoogleSheet」，點選搜尋到的**Google Sheet API** 開啟**Google Sheet API**視窗，按 **啟用** 鈕。
3. 在 **憑證** 頁面按下 **建立憑證\ 服務帳戶金鑰**。
4. 在 **建立服務帳戶** 頁面依 1 2 3 步驟的操作，步驟 1 在 **服務帳戶詳細資料** 輸入自訂名稱，然後按 **建立** 鈕。步驟 2 在 **角色** 下拉式清單選擇 **角色管理員**，然後按 **繼續** 鈕。步驟 3 直接按 **完成** 鈕。
5. 選取建立的服務帳戶， 按 **編輯服務帳戶** 圖示。

6. 在 **金鑰** 頁籤中，**新增金鑰** 下拉式清單中選擇 **建立新的金鑰**，**金鑰類型** 選擇 **JSON**，最後點選 **建立** 鈕。服務帳戶完成後將會建立.josn 金鑰檔並下載到本機。
7. 點選 **服務帳戶** 的電子郵件，可以顯示詳細的服務帳戶名稱，請複製 **電子郵件** 以供設定Google 試算表權限時使用。

3.6.3 Google 試算表的權限設定

1. 連到Google 雲端硬碟，新增Google 試算表，可以自訂名稱。
2. 點選 **共用** 鈕後，在開啟的對話方塊的 **新增使用者和群組** 欄輸入於 GoogleDevelopers Console 中所建立的服務帳戶名稱(電子郵件格式)，然後按 Enter鍵完成輸入。
3. 給予服務帳戶「編輯者」的權限，按 **傳送** 鈕完成設定，最後再按 **一律共用** 鈕。

GOTOP Python 大數據特訓班

3.6.4 連結Google 試算表

安裝相關模組

使用pip 安裝Google 試算表的相關模組，包括gspread 和oauth2client 模組。

```
!pip install gspread oauth2client
```

GOTOP Python 大數據特訓班

gspread 模組開啟試算表的流程

GOTOP

Python 大數據特訓班

取得 Google 資料表的 id

接著要取得Google 試算表的id，如下圖在試算表網址中反白處為資料表的id。

連結 Google 試算表開啟工作簿

1. 載入gspread 及oauth2client.service_account 的 ServiceAccountCredentials：

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials as sac
```

GOTOP

Python 大數據特訓班

2. 設定金鑰檔包含檔名的路徑，並設定程式可以操作的範圍。因為要用的是 Google 試算表，範圍是「https://spreadsheets.google.com/feeds」。例如：

```
auth_json = 'PythonConnectGsheet1-6a6086d149c5.json'
gs_scopes = ['https://spreadsheets.google.com/feeds']
```

3. 以 ServiceAccountCredentials 模組的 from_json_keyfile_name 方法，用金鑰檔及操作的範圍設定憑證建立連線物件。例如：

```
cr = sac.from_json_keyfile_name(auth_json, gs_scopes)
gc = gspread.authorize(cr)
```

4. 開啟資料表的方式有二種，第一種是使用檔案名稱，例如：

```
gsheet = gc.open('PythonConnectGSheet')
```

第二種是使用 Google 試算表的 id，例如：

```
gsheet = gc.open_by_key('1SG4KwqzA7p-tGTIohiYHSSauOiFuL9ZnmXw859RHa-c')
```

5. 開啟要使用的工作簿，例如：

```
wks = gsheet.sheet1
```

GOTOP Python 大數據特訓班

3.6.5 操作Google 試算表的資料

讀取試算表的資料

1. **讀取儲存格：acell()** 可以用位址讀取，**cell()** 可以用欄列號來讀取。
2. **讀取整列：row_values(列號)** 可以讀取整列的資料。
3. **讀取整欄：col_values(欄號)** 可以讀取整欄的資料。
4. **讀取所有資料：get_all_values()** 可以讀取所有的資料。

編輯試算表的資料

1. **清除所有資料：clear()** 可以清除工作簿上所有的資料。
2. **寫入儲存格：update_acell()** 可以用位址來寫入儲存格的值，**update_cell()** 可以用欄列號來寫入儲存格的值。
3. **新增列：append_row()** 可以新增一列的資料，新增的值必須為串列。
4. **插入列：insert_row()** 可以插入一列的資料，插入的值必須為串列。

GOTOP Python 大數據特訓班

範例：寫入Google 試算表

程式碼：LinkGoogleSheet.py

```
1   import gspread
2   from oauth2client.service_account import
                              ServiceAccountCredentials as sac
3   # 設定金鑰檔路徑及驗證範圍
4   auth_json = 'PythonConnectGsheet1-6a6086d149c5.json'
5   gs_scopes = ['https://spreadsheets.google.com/feeds']
6   # 連線資料表
7   cr = sac.from_json_keyfile_name(auth_json, gs_scopes)
8   gc = gspread.authorize(cr)
9   # 開啟資料表
10  spreadsheet_key = '1OihpM657yWo1lc3RjskRfZ8m75dCPwL1IPwoDXSvyzI'
11  sheet = gc.open_by_key(spreadsheet_key)
12  # 開啟工作簿
13  wks = sheet.sheet1
14  # 清除所有內容
15  wks.clear()
16  # 新增列
17  listtitle=['座號', '姓名', '國文', '英文', '數學']
18  wks.append_row(listtitle)  # 標題
19  listdatas=[[1, '葉大雄', 65, 62, 40],
20             [2, '陳靜香', 85, 90, 87],
21             [3, '王聰明', 92, 90, 95]]
22  for listdata in listdatas:
23      wks.append_row(listdata)  # 資料內容
```