

前言

近年來，Python 憑藉其強大的擴充性和豐富的第三方模組，不僅受到程式設計師和程式設計愛好者的青睞，而且在辦公自動化領域大顯身手，許多白領紛紛加入學習 Python 的行列。為滿足廣大職場人士用 Python 實作 Excel 辦公自動化的需求，我們於 2020 年 8 月編寫出版了《超簡單：用 Python 讓 Excel 飛起來》一書，該書面市後受到眾多讀者的歡迎。

為了能更好地滿足讀者的學習需求，我們再接再厲，在《超簡單：用 Python 讓 Excel 飛起來》的基礎上根據讀者的反饋進行改進，採用新的編寫思維和內容架構，編寫了這本《超簡單：Python 與 Excel 整合應用 150 技—pandas、xlwings、matplotlib、pycharts 大活用》。

本書淡化理論，以實作技巧為主體，內容按照 Excel 操作劃分章節，每一章的內容又分出多個小專題，力求每個專題解決一個問題。全書共 8 章。第 1 章主要講解 Python 程式設計環境的搭建、模組的安裝與匯入、Python 語法基礎知識、初學者常見問題等內容，為後面的案例整合應用打下堅實的基礎。第 2～8 章透過大量典型案例講解如何利用 Python 程式來操控 Excel，自動化和大量批次完成活頁簿檔案的相關操作、工作表的相關操作、列和欄的相關操作、儲存格的相關操作、資料處理與分析的相關操作、圖表的相關操作、列印的相關操作等。這些案例從工作中的應用場景入手，用 Python 程式設計解決對應的問題。書中附有詳細程式碼、易懂的注釋說明，能有效幫助讀者快速理解程式碼的適用範圍及編寫思維。此外還對程式碼涉及的重點語法和函式等知識進行延伸講解，引導讀者拓展思路，從機械式套用程式碼進階到能夠隨機應變修改程式碼，獨立解決更多實際問題。



本書理論知識精練，案例典型實用，隨附案例的相關檔案，不僅適合有一定 Excel 基礎又想進一步提高工作效率的辦公人員有系統地學習 Python 辦公自動化知識與技能，而且適合作為方便速查速用的實用手冊，對於 Python 程式設計愛好者來說也是不錯的參考資料。

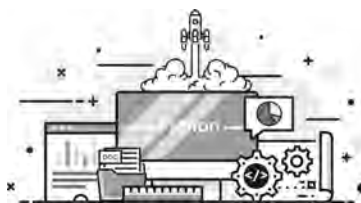
雖然編者已盡力，但本書難免有不足之處，懇請廣大讀者批評指教。

編者

2021 年 5 月

線上下載程式及相關應用檔案

書中每個案例技巧標題下都會列出該案用到的程式檔和資料檔，請至碁峰網站 http://books.gotop.com.tw/v_ACL065200 下載，其資料夾結構會對應每章內容，共有 8 章，讀者可在學習時進入其中對應資料夾取用相關檔案。由於使用者的電腦目錄路徑各不相同，請依照各章節內容的指示修改程式中使用到的活頁簿檔案路徑。提醒您，線上下載之內容僅供合法持有本書的讀者使用，未經授權不得抄襲、轉載或任意散佈。



第 2 章

Excel 活頁簿檔案 的相關操作

Excel 是各行各業比較常用的辦公軟體，它在資料編輯、處理和分析方面的表現都很出色，但用 Excel 處理重複性和機械性的工作時仍需要花費大量時間。本章將詳細介紹如何透過 Python 程式設計有效率地完成活頁簿的相關操作，如批次新建大量活頁簿、拆分和合併活頁簿、分類和尋找活頁簿檔案等。



030

移動並重新命名活頁簿檔案



程式檔：030-移動並重新命名活頁簿檔案.py

資料檔：員工檔案.xlsx

應用場景

檔案的移動和重新命名是檔案管理的基本操作。要在 Python 中移動和重新命名檔案，可以使用 `pathlib` 模組中的 `rename()` 函式。本案例將使用該函式把如下所示的活頁簿檔案移到另一個資料夾中，並重新命名。



實作程式碼

```
01 | from pathlib import Path # 匯入 pathlib 模組中的 Path 類別
02 | old_file_path = Path('F:\\python\\第 2 章\\員工檔案.xlsx') # 指定要移動並重新命名
    的活頁簿的路徑
03 | new_file_path = Path('F:\\table\\員工資訊表.xlsx') # 指定活頁簿移動和重新命名後
    的路徑
04 | old_file_path.rename(new_file_path) # 執行活頁簿的移動和重新命名操作
```

執行結果

執行本案例的程式碼，資料夾「F:\python\第 2 章」中的活頁簿「員工檔案.xlsx」會被移動到資料夾「F:\table」中，且檔案名稱變為「員工資訊表.xlsx」，如下圖所示。



程式碼解析

第 1 行程式碼用於匯入 `pathlib` 模組中的 `Path` 類別。`pathlib` 模組是 Python 的內建模組，無須安裝。該模組主要用於完成檔案和資料夾路徑的相關操作。

第 2 行程式碼用於指定要移動並重新命名的活頁簿的檔案路徑，這裡指定的是位於資料夾「F:\python\第 2 章」中的「員工檔案.xlsx」，讀者可根據實際需求修改這個路徑。

第 3 行程式碼用於指定活頁簿移動和重新命名後的路徑，這裡指定目的資料夾為「F:\table」，新檔案名為「員工資訊表.xlsx」，讀者可根據實際需求修改目的資料夾和檔案名。需要注意的是，目的資料夾（如這裡的「F:\table」）必須真實存在，且目的資料夾中不能有同名檔案（如這裡的「員工資訊表.xlsx」），否則執行時會回報錯誤訊息。

第 4 行程式碼用於執行活頁簿的移動和重新命名操作，動作是把資料夾「F:\python\第 2 章」下的活頁簿「員工檔案.xlsx」移動到資料夾「F:\table」下，並重新命名為「員工資訊表.xlsx」。

知識延伸

- 1) 第 1 行程式碼中匯入的 `Path` 類別代表作業系統中資料夾和檔案的路徑。要使用 `Path` 類別的功能，需先將其實例化為一個路徑物件，第 2 行和第 3 行程式碼就是在做這件事。類別、物件、實例化是物件導向程式設計中的概念，讀者可以先不深究，只需記住程式碼的書寫格式。
- 2) 第 2 行和第 3 行程式碼用於將路徑字串新建為路徑物件。這裡要注意路徑字串的書寫格式：因為在 Python 中「\」有特殊含義，如「\n」表示換行，所以通常在路徑中使用兩個「\」，以跳脫單個「\」的特殊含義。



- 3) 新建了路徑物件後，即可呼叫路徑物件的屬性或函式來獲取資訊或完成操作，基本語法格式為「物件名稱.屬性」和「物件名稱.函式」。第 4 行程式碼中呼叫的 `rename()` 函式主要用於重新命名檔案或資料夾，如果來源路徑和目標路徑所在的資料夾不同，還可以達到移動檔案或資料夾的效果。但是，`rename()` 函式只能在同一個磁碟分割區中移動檔案或資料夾。例如，本案例中的活頁簿「員工檔案.xlsx」只能被移動到 F 磁碟的資料夾中，不能被移動到 C、D、E 磁碟的資料夾中。

031

解析活頁簿檔案的路徑資訊



程式檔：031-解析活頁簿檔案的路徑資訊.py

資料檔：出庫表.xlsx

應用場景

在工作中處理活頁簿時，可能需要從活頁簿的檔案路徑中提取資料夾、檔案名稱、副檔名等資訊，使用 `pathlib` 模組中路徑物件的 `parent`、`name`、`stem`、`suffix` 等屬性可以快速達到目的。

實作程式碼

```
01 | from pathlib import Path # 匯入 pathlib 模組中的 Path 類別
02 | file_path = Path('F:\\python\\第 2 章\\出庫表.xlsx') # 指定要操作處理的活頁簿的檔案路徑
03 | path = file_path.parent # 提取活頁簿所在資料夾的路徑
04 | file_name = file_path.name # 提取活頁簿的檔案名稱
05 | stem_name = file_path.stem # 提取活頁簿檔案的主檔名
06 | suf_name = file_path.suffix # 提取活頁簿的副檔名
07 | print(path) # 輸出活頁簿所在資料夾的路徑
08 | print(file_name) # 輸出活頁簿的檔案名稱
09 | print(stem_name) # 輸出活頁簿的主檔名
10 | print(suf_name) # 輸出活頁簿的副檔名
```

執行結果

本案例的程式執行結果如下：



程式碼解析

第 2 行程式碼將要操作的活頁簿的檔案路徑字串新建為路徑物件，並指定給變數 `file_path`。讀者可根據實際需求修改檔案路徑。

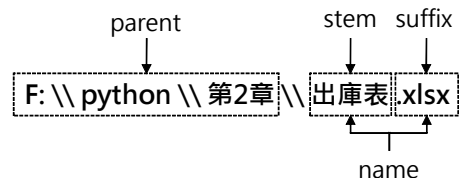
第 3~6 行程式碼分別用於從活頁簿的檔案路徑中提取資料夾路徑、檔案名稱、主檔名和副檔名。

第 7~10 行程式碼分別輸出提取結果。

知識延伸

1) 路徑通常有絕對路徑和相對路徑兩種表示方式。絕對路徑表示從根資料夾開始的完整路徑。例如，在 Windows 中以磁碟代號 (C:、D:等) 作為根資料夾。第 2 行程式碼中的「F:\python\第 2 章\出庫表.xlsx」就是一個絕對路徑。相對路徑表示相對於目前執行的程式檔的路徑。例如，如果將第 2 行程式碼修改為「`file_path = Path('出庫表.xlsx')`」，那麼該路徑物件指向的就是程式檔所在資料夾下的活頁簿「出庫表.xlsx」。

2) 第 3~6 行程式碼中的 `parent`、`name`、`stem`、`suffix` 是 `pathlib` 模組中路徑物件的屬性，分別用於返回資料夾路徑、檔案名稱、主檔名和副檔名，如右圖所示。





032

提取資料夾內所有活頁簿的檔名



程式檔：032-提取資料夾內所有活頁簿的檔名.py

資料檔：工作資訊表（資料夾）

應用場景

如下圖所示，資料夾「F:\python\第 2 章\工作資訊表」中存放了多種類型的檔案。如果要從中提取特定類型的所有檔案的檔名，如提取所有 Excel 活頁簿的檔名，可以結合使用路徑物件的 `glob()` 函式和 `name` 屬性快速達到目的。

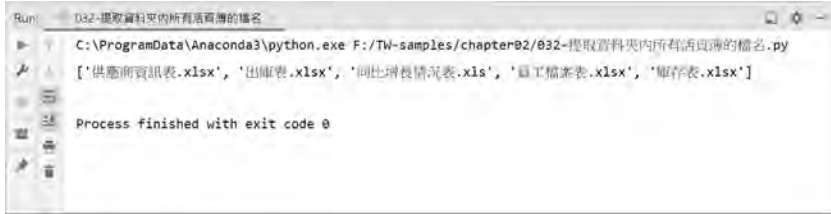


實作程式碼

```
01 | from pathlib import Path # 匯入 pathlib 模組中的 Path 類別
02 | folder_path = Path('F:\\python\\第 2 章\\工作資訊表\\') # 列出要操作的資料夾路徑
03 | file_list = folder_path.glob('*.xls*') # 獲取資料夾下所有活頁簿的檔案路徑
04 | lists = [] # 新建一個空串列
05 | for i in file_list: # 遍訪獲取的檔案路徑
06 |     file_name = i.name # 提取活頁簿的檔名
07 |     lists.append(file_name) # 將提取的檔名新增到串列中
08 | print(lists) # 輸出提取結果
```

執行結果

本案例的程式碼執行結果如下：



程式碼解析

第 2 行程式碼將要操作的資料夾路徑字串新建為路徑物件，讀者可根據實際需求修改這個路徑。

第 3 行程式碼用於在資料夾「F:\python\第 2 章\工作資訊表」下尋找名稱符合規則「*.xls*」的所有檔案。「*.xls*」表示檔名含有關鍵字「.xls」，即 Excel 活頁簿的檔案。如果要尋找其他類型的檔案，如 Word 文件，則將括弧裡的參數修改為「*.doc*」。

第 4 行程式碼新建了一個空串列，用於儲存提取出來的活頁簿名稱。

第 5~7 行程式碼用 for 語句建構了一個迴圈，從檔案路徑中提取檔案名並新增到第 4 行程式碼新建的串列中。第 6 行程式碼從檔案路徑中提取檔名，如果要提取主檔名或副檔名，將程式碼修改為「file_name = i.stem」或「file_name = i.suffix」。第 7 行程式碼在串列中附加提取的檔名。

第 8 行程式碼輸出串列的內容，即提取到的所有檔名。

知識延伸

- 1) 第 3 行程式碼中的 `glob()` 函式用於尋找名稱符合指定規則的檔案或資料夾並返回它們的路徑。括弧裡的參數是尋找條件，並且可以在條件中使用萬用字元：「*」可以匹配任意數量的（包括 0 個）字元，「?」可以匹配單個字元，「[]」可以匹配指定範圍內的字元。如果不使用萬用字元，表示進行精確搜尋；如果使用萬用字元，則表示進行模糊搜尋。
- 2) 第 7 行程式碼中的 `append()` 函式用於在串列的尾部新增元素。需要注意的是，該函式一次只能附加一個元素。示範程式碼如下：



```
01 | a = [1, 2, 3, 4, 5, 6]
02 | a.append(7)
03 | print(a)
```

執行結果如下：

```
01 | [1, 2, 3, 4, 5, 6, 7]
```

033

新建並儲存一個活頁簿



程式檔：033-新建並儲存一個活頁簿.py

應用場景

新建活頁簿是 Excel 中最基本的操作之一。本案例要使用 Python 中的 xlwings 模組新建一個活頁簿並將其儲存到指定路徑下，為下一個案例學習批次新建一堆活頁簿檔案做準備。

實作程式碼

```
01 | import xlwings as xw # 匯入 xlwings 模組
02 | app = xw.App(visible=False, add_book=False) # 啟動 Excel 程式
03 | workbook = app.books.add() # 新建活頁簿
04 | workbook.save('F:\\test\\1 月銷售表.xlsx') # 儲存新建的活頁簿
05 | workbook.close() # 關閉活頁簿
06 | app.quit() # 退出 Excel 程式
```

執行結果

執行本案例的程式碼後，可在資料夾「F:\test」下看到生成了一個活頁簿「1 月銷售表.xlsx」，如下圖所示。



程式碼解析

第 1 行程式碼用於匯入 `xlwings` 模組並簡寫為 `xw`，這行程式碼是新建活頁簿的基礎。

第 2 行和第 3 行程式碼分別用於啟動 Excel 程式並新建一個活頁簿。

第 4 行程式碼按照指定的路徑儲存新建的活頁簿。這裡使用的是絕對路徑，表示將活頁簿儲存到資料夾「`F:\test`」（需提前建好該目錄）中，檔名為「`1 月銷售表.xlsx`」。也可以使用相對路徑，如「`workbook.save('1 月銷售表.xlsx')`」，表示在程式檔所在資料夾下儲存活頁簿「`1 月銷售表.xlsx`」。

第 5 行和第 6 行程式碼分別用於關閉目前活頁簿並退出 Excel 程式。案例 028 中講過，完成所需操作後及時關閉活頁簿並退出 Excel 程式很重要，請讀者重視和注意。

知識延伸

- 1) 第 1 行程式碼中匯入的 `xlwings` 是一個用於操控 Excel 的 Python 第三方協力廠商開放原始碼模組，需由使用者自行安裝，具體方法見案例 006 和案例 007。
- 2) 第 2 行程式碼中的 `App` 是 `xlwings` 模組中的物件，用於啟動 Excel 程式。該物件有兩個常用初始化參數：參數 `visible` 用於設定 Excel 程式視窗的可見性，為 `True` 時表示顯示視窗，為 `False` 時表示隱藏視窗；參數 `add_book` 用於設定啟動 Excel 程式後是否新建活頁簿，為 `True` 時表示新建一個活頁簿，為 `False` 時表示不新建活頁簿。



- 3) 第 3 行程式碼中的 `add()` 為 `xlwings` 模組中 `Books` 物件的函式，用於新建活頁簿，該函式沒有參數。
- 4) 第 4 行程式碼中的 `save()` 和第 5 行程式碼中的 `close()` 都是 `xlwings` 模組中 `Book` 物件的函式，分別用於儲存活頁簿和關閉活頁簿。`save()` 函式括弧裡的參數為活頁簿的儲存路徑。如果儲存路徑指向的活頁簿已經存在，`save()` 函式會直接將其覆蓋。`close()` 函式則沒有參數。
- 5) 第 6 行程式碼中的 `quit()` 是 `xlwings` 模組中 `App` 物件的函式，用於退出 Excel 程式，該函式沒有參數。

034

批次新建並儲存多個活頁簿



程式檔：034-批次新建並儲存多個活頁簿.py

應用場景

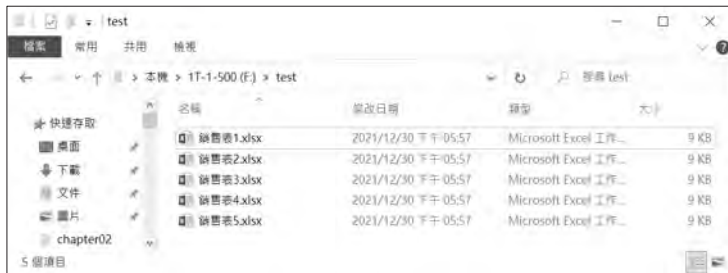
在上一個案例中學習了如何使用 `xlwings` 模組新建一個活頁簿。在此基礎上結合使用 `for` 語句，就可以快速批次新建多個檔案名稱有規律的活頁簿檔案。本案例以批次新建檔名為「銷售表×.xlsx」（「×」為整數序列）的活頁簿為例來進行講解。

實作程式碼

```
01 | import xlwings as xw # 匯入 xlwings 模組
02 | app = xw.App(visible=False, add_book=False) # 啟動 Excel 程式
03 | for i in range(1, 6):
04 |     workbook = app.books.add() # 新建活頁簿
05 |     workbook.save(f'F:\\test\\銷售表{i}.xlsx') # 儲存新建的活頁簿
06 |     workbook.close() # 關閉活頁簿
07 | app.quit() # 退出 Excel 程式
```

執行結果

執行本案例的程式碼後，在資料夾「F:\test」下可以看到新建的 5 個活頁簿，檔案名分別為「銷售表 1.xlsx」「銷售表 2.xlsx」「銷售表 3.xlsx」「銷售表 4.xlsx」「銷售表 5.xlsx」，如下圖所示。



程式碼解析

第 3~6 行程式碼用 for 語句建構了一個迴圈來完成活頁簿的批次新建和儲存。本案例將批次新建 5 個活頁簿。如果要新建更多活頁簿，可將第 3 行程式碼中 range() 函式的第 2 個參數值改為更大的數值。

第 4 行程式碼新建一個活頁簿後，第 5 行程式碼接著在指定資料夾下以有規律的檔案名儲存新建的活頁簿。這裡的儲存路徑使用的是絕對路徑，如果寫成相對路徑，如「workbook.save(f'銷售表{i}.xlsx)」，則會在程式檔所在的資料夾下儲存活頁簿。路徑中的「銷售表{i}.xlsx」指定了活頁簿的檔名，可以根據實際需求修改。其中的「銷售表」和「.xlsx」是檔名中的固定部分，而「{i}」則是可變部分，執行時會被取代為 i 的實際值，即依次變為 1、2、3、4、5。

知識延伸

- 1) 第 3 行程式碼中的 range() 函式在案例 015 中講解過，當時只設定了 1 個參數，表示明確指定終止值，而起始值則使用預設值 0。這裡的 range(1,6) 則設定了兩個參數，表示明確指定起始值和終止值，生成的整數序列為 1、2、3、4、5（注意「左閉右開」的特性）。
- 2) 第 5 行程式碼在建構有規律的檔名時使用了 f-string 方法來拼接字串。該方法以 f 或 F 為修飾符號引領字串，然後在字串中以「{ }」標明要取代為



變數值的內容。使用該方法無須事先轉換資料型別就能將不同類型的資料拼接成字串。示範程式碼如下：

```
01 | name = '小王'
02 | score = 100
03 | a = f'{name}考了{score}分。'
04 | print(a)
```

執行結果如下：

```
01 | 小王考了 100 分。
```

035

打開一個現有的活頁簿檔案



程式檔：035-打開一個現有的活頁簿檔案.py

資料檔：出庫表.xlsx

應用場景

打開現有活頁簿檔案也是 Excel 中最基本的操作之一。本案例要使用 `xlwings` 模組打開一個已存在的活頁簿檔案，為下一個案例學習批次打開多個活頁簿檔案做準備。

實作程式碼

```
01 | import xlwings as xw # 匯入 xlwings 模組
02 | app = xw.App(visible=True, add_book=False) # 啟動 Excel 程式
03 | file_path = 'F:\\python\\第 2 章\\出庫表.xlsx' # 指定要打開的活頁簿的檔案路徑
04 | app.books.open(file_path) # 打開指定的活頁簿
```

執行結果

執行本案例的程式碼，可看到用 Excel 打開了指定活頁簿，如下圖所示。



	A	B	C	D	E	F	G	H	I	J	K	L
1	配件編號	配件名稱	出庫數量	單位	單價							
2	FB05211450	組合鎖	10	個	20							
3	FB05211451	擰旋杆	20	個	60							
4	FB05211452	轉接表	50	塊	200							
5	FB05211453	里燈表	600	塊	280							
6	FB05211454	組合表	30	個	850							
7	FB05211455	總接器	70	個	30							
8	FB05211456	膠墊	80	個	30							
9	FB05211457	避壓墊	111	個	90							
10	FB05211458	調整墊	540	個	6							
11	FB05211459	上襯套	20	個	10							
12	FB05211460	主軸	30	個	60							
13	FB05211461	下襯套	60	個	10							

程式碼解析

第 2 行程式碼用於啟動 Excel 程式。這裡為了查看程式碼的執行結果，將 App 物件的初始化參數 `visible` 設定為 `True`，表示顯示 Excel 程式視窗。

第 3 行程式碼指定要開啟的活頁簿的檔案路徑，這裡指定的是資料夾「F:\python\第 2 章」下的「出庫表.xlsx」活頁簿檔案。讀者可根據實際需求修改檔案路徑。

第 4 行程式碼用於開啟指定的活頁簿。需要注意的是，第 3 行程式碼列出的路徑指向的活頁簿必須真實存在，而且不能處於已打開的狀態，否則執行至第 4 行程式碼時會回報錯誤訊息。

知識延伸

第 4 行程式碼中的 `open()` 是 `xlwings` 模組中 `Books` 物件的函式，用於開啟活頁簿，括弧裡的參數為活頁簿的檔案路徑。



036

打開資料夾下的所有活頁簿檔案



程式檔：036-打開資料夾下的所有活頁簿檔案.py

資料檔：工作資訊表（資料夾）

應用場景

在實際工作中經常要同時查看多個活頁簿檔案時，如果逐個開啟，需要花上好幾分鐘。本案例將透過 Python 程式設計，快速批次開啟下圖所示的資料夾中的多個活頁簿。

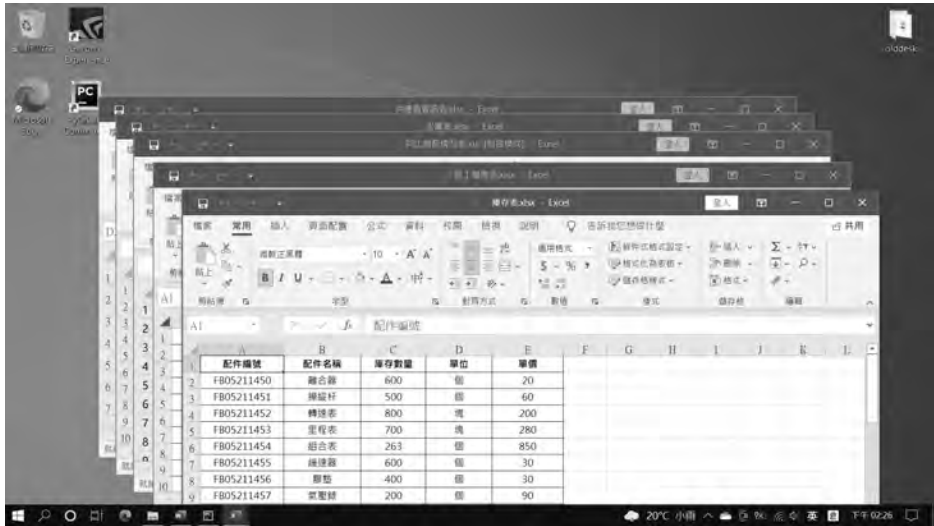


實作程式碼

```
01 | from pathlib import Path # 匯入 pathlib 模組中的 Path 類別
02 | import xlwings as xw # 匯入 xlwings 模組
03 | app = xw.App(visible=True, add_book=False) # 啟動 Excel 程式
04 | folder_path = Path('F:\\python\\第 2 章\\工作資訊表\\') # 列出要開啟的活頁簿所在
    的資料夾路徑
05 | file_list = folder_path.glob('*.xls*') # 獲取資料夾下所有活頁簿的檔案路徑
06 | for i in file_list: # 遍訪取得的檔案路徑
07 |     app.books.open(i) # 開啟活頁簿
```

執行結果

執行本案例的程式碼，自動打開指定資料夾中的所有活頁簿，如下圖所示。



程式碼解析

第 4 行程式碼用於指定要打開的活頁簿所在的資料夾路徑，這裡指定的是資料夾「F:\python\第 2 章\工作資訊表」。讀者可根據實際需求修改這個路徑。

第 5 行程式碼用於獲取指定資料夾下所有活頁簿的檔案路徑。

第 6 行和第 7 行程式碼用 for 語句遍訪獲取的活頁簿檔案路徑，並依次根據路徑打開活頁簿。

知識延伸

第 5 行程式碼中的 glob() 函式在案例 032 介紹過，第 7 行程式碼中的 open() 函式在案例 035 介紹過，這裡不再贅述。