

Chapter

08

實戰：Facebook 貼文、按讚與照片下載

Facebook 是目前最流行的社群網站，日常生活的分享、相關資訊的吸收，以及人與人之間交流，似乎都離不開這個無形的領域。

如何使用 Python 來進行 Facebook 上的操作，是許多人很有興趣的主題，這裡將以一些實用的功能進行實例的操作。

本章重點在了解使用 Facebook 應用程式的開發工具：Graph API Explorer 後學習如何進行 Facebook 功能的操作，讓 Python 透過 facebook-sdk 套件，在取得存取權限後可以透過 Facebook 的 API 套件直接存取 Facebook 的資料，以達到許多實用的功能。



Python 初學特訓班





8.2.4 應用：Facebook 貼文

put_wall_post(message ,attachment [,profile_id])

`put_wall_post(message ,attachment [,profile_id])` 可以將 `message` 訊息張貼到指定使用者的動態牆上。參數 `message` 表示文章內容，`attachment` 表示詳細內容，`profile_id` 為指定的 id，省略時預設為自己的動態牆。

使用 Python 程式張貼文章，請先開啟「publish_actions」權限。



例如：在自己的動態牆上發佈一則貼文。完成後可在 Facebook 上看到這樣畫面。
(<fbpost.py>)



```
1 import facebook
2
3 token="EAACEdEose0cBAG4KWXYEsx3R50DadEe3e...略 "
4 graph = facebook.GraphAPI(access_token=token,version='2.7')
5
6 message = " 測試使用 Python 程式在 fb 上貼文!"
7 attachment = {
8     'name': ' 中老年人快樂學拍照、攝影 ',
9     'link': 'http://www.e-happy.com.tw/indexbookshow.asp?bid=278',
10    'caption': ' 新書介紹 ',
```

```
11 'description': '智慧型手機或平板電腦擁有多功能且方便隨身攜帶的優  
勢，拍出來的相片、影片還可以透過軟體直接編修美化與分享更是方便。本書以熟齡  
者用智慧型手機或平板電腦拍攝的角度出發，簡單易懂的說明與詳細的步驟講解，讓  
人人都能輕鬆用手機拍下身邊的風景，並與家人、朋友分享生活每一刻的心情。',  
12 'picture': 'http://www.e-happy.com.tw/images/bookimg/10-056-S.jpg'  
13 }  
14  
15 graph.put_wall_post(message=message, attachment=attachment)  
16 print(" 貼文成功!")
```

8.2.5 應用：Facebook 自動按讚

put_like(object_id)

使用 `put_like(object_id)` 可以對指定的物件按讚，我們只要取得文章的 `id`，就可以利用 Python 程式對這篇文章按讚。

例如：自己對自己張貼的每篇文章按讚。(<fbputlike.py>)

```
Python console  
Console 1/A  
798156653603892_1126322320787322 按讚成功!  
798156653603892_1124793047606916 按讚成功!  
798156653603892_1124784960941058 按讚成功!  
798156653603892_1077005465719008 按讚成功!  
798156653603892_1045394972213391 按讚成功!  
798156653603892_1020248361394719 按讚成功!  
798156653603892_1020234718062750 按讚成功!  
798156653603892_922760621143494 按讚成功!
```

```
1 import facebook  
2  
3 token="EAACEdEose0cBAG4KWXYEsx3R5ODadEe3eIZB7... 略"  
4 graph = facebook.GraphAPI(access_token=token, version='2.7')  
5  
6 pages = graph.get_connections(id='me', connection_name='posts')  
7 posts = pages['data']  
8  
9 for p in posts:  
10 graph.put_like(p['id'])  
11 print (p['id'], " 按讚成功!")
```

如果改為「`connection_name='feed'`」則會對即時動態的每一篇文章按讚。

Chapter

10

實戰：Word 文件處理

Office 的文件是我們日常生活工作都常用到的文件格式，其中 Word 格式的檔案更是重要。

Python 語言可透過 Win32com 套件對 Microsoft Office 文件進行存取，而 Python 已內含 Win32com 套件，不需另外安裝。若要使用 Win32com 套件處理 Microsoft Office 文件，電腦必須已安裝 Microsoft Office 軟體。

本章利用 Win32com 套件製作兩個實際應用：自動建立整個月份的營養午餐菜單 Word 文件，及自動取得指定目錄中所有 Word 文件 (包含子目錄) ，並對所有 Word 檔案進行置換文字功能。



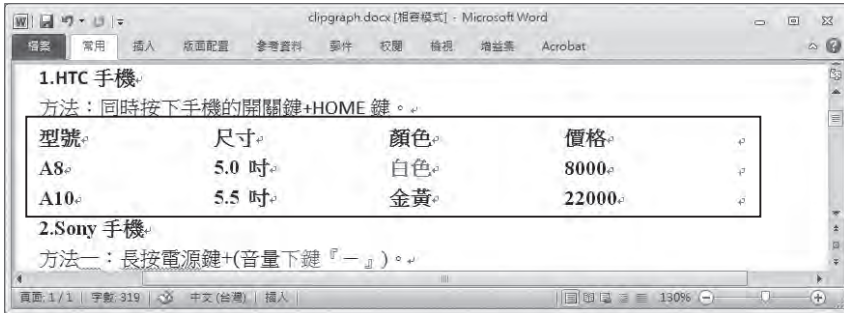
Python 初學特訓班





範例：Word 文件新增表格及儲存格內容

在 Word 文件中新增一個 3 列 4 行的表格，再使用迴圈設定儲存格內容。



程式碼：ch10\tabledocx1.py

```
.....略
7 doc = word.Documents.Open(cpath + "\\media\\clipgraph.docx")
8 data = [ [" 型號 ", " 尺寸 ", " 顏色 ", " 價格 "],
           [ "A8", "5.0 吋", " 白色 ", "8000"], \
9         [ "A10", "5.5 吋", " 金黃 ", "22000" ] ]
10 paragraphs = doc.Paragraphs
11 range1 = paragraphs(4).Range
12 table = doc.Tables.Add(range1, 3, 4)
13 for i in range(1,table.Rows.Count+1):
14     for j in range(1,table.Columns.Count+1):
15         table.Cell(i,j).Range.Text = data[i-1][j-1]
16 table.Cell(2,3).Range.Font.Color = 0x0000FF
17 #doc.Close()
18 #word.Quit()
```

程式說明

- 8-9 建立 3x4 的二維串列做為表格儲存格的內容。
- 10-11 因段落 4 只有一個換行符號，所以用段落 4 做為插入表格的位置。
- 12 建立 3 列 4 行的表格。
- 13-14 使用迴圈逐一加入儲存格內容：注意儲存格編號是由 1 開始，所以 range 函式範圍要由 1 到列或行數量加 1。
- 15 儲存格編號是由 1 開始，串列索引值則由 0 開始，所以串列索引要使用「data[i-1][j-1]」。
- 16 設定第 2 列第 3 行儲存格內容顏色為紅色。

10.1.5 加入圖片

在 Word 文件加入圖片的語法為：

```
範圍變數.InlineShapes.AddPicture(檔案路徑, 連結, 儲存)
```

- **連結**：是布林值，True 表示連結到原始圖片檔，False 表示不連結到圖片檔。
- **儲存**：是布林值，True 表示將原始圖片檔存於 Word 檔中，False 表示 Word 檔中不儲存圖片檔。

例如在 <clipgraph.docx> 段落 4 插入 <cell.jpg> 圖片，並將圖片檔儲存於 Word 檔內：(<imagedocx1.py>)

……略

```
doc = word.Documents.Open(cpath + "\\media\\clipgraph.docx")
paragraphs = doc.Paragraphs
range1 = paragraphs(4).Range
range1.InlineShapes.AddPicture(cpath + "\\media\\cell.jpg", False, True)
```

執行結果為：



10.1.6 取代文字

Win32com 套件提供自動取代文件中指定文字的功能。通常使用取代文字前會先清除搜尋文字及取代文字的格式，以免因格式影響取代效果，語法為：

```
Word 應用變數.Selection.Find.ClearFormatting()
```

```
Word 應用變數.Selection.Find.Replacement.ClearFormatting()
```



取代 Word 文件特定文字的語法為：

Word 應用變數 `.Selection.Find.Execute` (被取代文字, 比對 1, 比對 2, 比對 3, 比對 4, 比對 5, 方向, 動作, 格式, 取代文字, 取代次數)

- **比對 1**：布林值，True 為區分大小寫字母，False 為不分大小寫字母。
- **比對 2**：布林值，True 為完整文字，False 為部分文字也取代。
- **比對 3**：布林值，True 為可使用萬用字元，False 為不可使用萬用字元。
- **比對 4**：布林值，True 為可使用 Like 比對，False 為不可使用 Like 比對。
- **比對 5**：布林值，True 為完全符合時態，False 為不分時態 (例如過去式、過去分詞也可以)。
- **方向**：布林值，True 表示向前搜尋，False 表示向後搜尋。
- **動作**：搜尋到被取代文字後所執行的動作，可能的常數值有：
 - constants.wdFindAsk**：顯示對話方塊詢問使用者是否繼續搜尋。
 - constants.wdFindContinue**：取代後繼續搜尋。
 - constants.wdFindStop**：取代後停止搜尋。
- **格式**：布林值，True 表示需符合格式，False 表示不需符合格式。
- **取代次數**：設定取代的次數，可能的常數值有：
 - constants.wdReplaceNone**：不取代。
 - constants.wdReplaceOne**：只取代一次。
 - constants.wdReplaceAll**：取代全部。

範例：Word 文件取代文字

將 Word 文件中所有「方法」都取代為「method」。



Chapter

12

實戰：臉部辨識及 驗證碼圖片破解

OpenCV 是一個開發原始碼、跨平台的電腦視覺程式庫，可以在商業和研究領域中免費使用，目前已應用於人機互動、臉部識別、動作識別、運動跟蹤等領域。

要進行特定圖像辨識最重要的是要有辨識對象特徵檔，OpenCV 已內建臉部辨識特徵檔，只要使用 OpenCV 的 `CascadeClassifier` 類別即可辨識臉部。

在許多網站都會利用圖形驗證碼來阻擋網站上不當或惡意的訪問動作，如果要進行驗證碼圖片破解，要將圖形驗證碼轉換為文字。Python 可以透過圖形處理套件將大部分圖片背景去除，再以 Tesseract 套件以 OCR 功能讀取圖片文字進行破解。



Python 初學特訓班



12.2.3 實戰：驗證碼破解

應用程式總覽

程式執行後會顯示驗證碼圖形，按任意鍵後會在命令視窗顯示辨識結果 (<format.py>)



應用程式內容

程式碼：ch02\format.py

```
1 import cv2, subprocess
2 img = cv2.imread("media\\bank.jpg") # 讀圖
3 cv2.namedWindow("Image")
4 cv2.imshow("Image", img) # 顯示圖形
5 cv2.waitKey(0)
6 cv2.destroyWindow("Image")
7 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # 轉為灰階
8 _, inv = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY_INV)
9 # 轉為反相黑白
10 for i in range(len(inv)): # i 為每一列
11     for j in range(len(inv[i])): # j 為每一行
12         if inv[i][j] == 255: # 顏色為白色
13             count = 0
14             for k in range(-2, 3):
15                 for l in range(-2, 3):
16                     try:
17                         if inv[i + k][j + l] == 255:
18                             count += 1
19                     except IndexError:
20                         pass
21             if count <= 6: # 週圍少於等於 6 個白點
22                 inv[i][j] = 0 # 將白點去除
23 dilation = cv2.dilate(inv, (8,8), iterations=1) # 圖形加粗
24 cv2.imwrite("media\\bank_t.jpg", dilation) # 存檔
```

Chapter

16

實戰： PyGame 遊戲開發

遊戲開發是許多程式語言很喜歡的一個領域，因為遊戲開發需要使用的技術範圍相當的廣，除了多媒體音效、圖片動畫，程式設計應用更是其中的核心。

PyGame 是為了讓 **Python** 能夠進行遊戲開發工作所發展出來的套件，它能幫助 **Python** 控制音效音樂、圖片動畫，並進程式的運作，是一個十分強大，功能完整的套件。

在本章中將詳細說明 **PyGame** 的使用方式，並利用實例範例帶領讀者學習其中重要的技巧，最後再利用一個有趣又好玩的遊戲進行專題開發，讓您也可以利用 **Python** 快速的進入遊戲開發的世界。



Python 初學特訓班



16.3 實戰：打磚塊遊戲

三十餘年前，街頭電玩遊戲機中最流行的遊戲就是「打磚塊」。時至今日，雖然網路連線遊戲日新月異，「打磚塊」這款小遊戲仍在許多人心中佔有一席之地，本應用製作一個簡單的打磚塊遊戲。

16.3.1 應用程式總覽

開始時下方會顯示「按滑鼠左鍵開始遊戲」訊息，使用者按滑鼠左鍵就顯示遊戲畫面。使用者移動滑鼠控制滑板，滑板只能左右移動，位置與滑鼠 x 坐標相同；共有 60 個磚塊，被球撞到的磚塊會消失，同時分數會增加，球撞到磚塊及滑板會發出不同音效。



如果球體碰到下邊界就表示球體已出界，顯示「失敗，再接再勵！」訊息並結束程式；若全部磚塊都消失則顯示「恭喜，挑戰成功！」訊息並結束程式。





16.3.2 球體、磚塊、滑板角色類別

本遊戲主角是球體、磚塊及滑板，都設計為角色類別。首先是球體角色，與 16.2.4 節的球體角色類似：

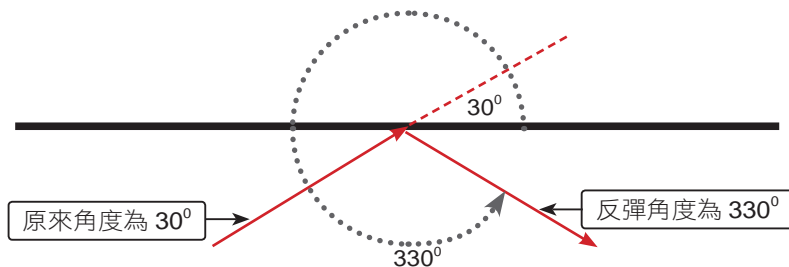
程式碼：ch16\brickgame.py

```
1 import pygame, random, math, time
2
3 class Ball(pygame.sprite.Sprite): # 球體角色
4     dx = 0 #x 位移量
5     dy = 0 #y 位移量
6     x = 0 # 球 x 坐標
7     y = 0 # 球 y 坐標
8     direction = 0 # 球移動方向
9     speed = 0 # 球移動速度
10
11     def __init__(self, sp, srx, sry, radius, color):
12         pygame.sprite.Sprite.__init__(self)
13         self.speed = sp
14         self.x = srx
15         self.y = sry
16         self.image = pygame.Surface([radius*2, radius*2]) #繪製球體
17         self.image.fill((255,255,255))
18         pygame.draw.circle(self.image, color, (radius,radius), radius, 0)
19         self.rect = self.image.get_rect() # 取得球體區域
20         self.rect.center = (srx,sry) # 初始位置
21         self.direction = random.randint(40,70) # 移動角度
22
23     def update(self): # 球體移動
24         radian = math.radians(self.direction) # 角度轉為強度
25         self.dx = self.speed * math.cos(radian) # 球水平運動速度
26         self.dy = -self.speed * math.sin(radian) # 球垂直運動速度
27         self.x += self.dx # 計算球新坐標
28         self.y += self.dy
29         self.rect.x = self.x # 移動球圖形
30         self.rect.y = self.y
31         if(self.rect.left <= 0 or self.rect.right >=
32             screen.get_width()-10): # 到達左右邊界
33             self.bouncelr()
34         elif(self.rect.top <= 10): # 到達上邊界
35             self.rect.top = 10
36             self.bounceup()
```

```
36         if(self.rect.bottom >= screen.get_height()-10):
37             # 到達下邊界出界
38             return True
39         else:
40             return False
41     def bounceup(self): # 上邊界反彈
42         self.direction = 360 - self.direction
43
44     def bouncelr(self): # 左右邊界反彈
45         self.direction = (180 - self.direction) % 360
```

程式說明

- 11-21 建立球體圖形、設定初始位置、隨機設定起始移動方向。
- 23-39 球體移動自訂函式。
- 24-26 根據移動方向計算水平及垂直速度。
- 27-30 計算球體新坐標並移到新位置。
- 31-32 球體碰到左、右邊界時進行反彈。
- 33-35 球體碰到上邊界時進行反彈：34 列在碰到上邊界時設其 y 坐標為 10，避免連續碰撞上邊界。
- 36-39 若球體碰到下邊界就傳回 True 表示球出界，否則傳回 False 表示球未出界。
- 41-42 球體碰到上邊界時的反彈處理函式：「反彈角度 = 360° - 原來角度」。例如下圖原來角度為 30° 度，反彈角度為 330° 度。



- 44-45 球體碰到左、右邊界時的反彈處理函式：「反彈角度 = 180° - 原來角度」。因為反彈角度可能得到負值，所以再用除以 360 的餘數將其轉為正數。例如下圖原來角度為 30° 度，反彈角度為 150° 度。