書籍勘誤資訊表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 書號 |  | 書名 | Python程式設計入門 金融商管實務案例 第三版 | | |
| 勘誤位置 | 勘誤前內容 | | | 勘誤後資訊 | 備註 |
| 1-2 | Python 與 Perl 同為開放原始碼軟體（Open Source Software），降低使用單位，如 企業的開發和維護的成本。 | | | Python 與 Perl 同為開放原始碼軟體（Open Source Software），降低使用費用，如 企業的開發和維護的成本。 |  |
| 2-17 | 使用者可以更改環境偏好設定。點選「Tools」下的「Preferences」即可進行偏好 設定視窗（見圖2-24）。點選「Editor」，改字型大小「Size」為 12，按「OK」，即可完成。 | | | 使用者可以更改環境偏好設定。點選「Tools」下的「Preferences」即可進行偏好 設定視窗（見圖2-24）。點選「General」，改字型大小「Size」為 12，按「OK」，即可完成。 |  |
| 3-12 | 表格3-3最後一行   |  |  | | --- | --- | | join(stt1, str2) | 連接字串。 | | | | 表格3-3最後一行  join(str1, str2) |  |
| 3-21 |  | | | 第二列應為 5 2 4 |  |
| 3-22 | range(start,end,step) | | | [start:end:step] |  |
| 3-23 | 表3-6  reverse()：串列的元素由大到小排序。 | | | reverse()：串列的元素倒著排序。 |  |
| 3-43 | 9. 建立兩個 list 分別為 lis1 內容是 range(5,20,1) 和 lis2 內容 (9,24,1)，計算兩個 list 的總合，將總和加入 ~~list~~ 到各自 list 當中，~~兩個 list 由大到小排序~~，即 lis1 做加總新增到 lis1 最後 , lis2 做加總新增到 lis2 最後。lis1 與 lis2 分別做排序，分別取出最後一個數字相乘輸出。 | | | 9. 建立兩個 list 分別為 lis1 內容是 range(5,20,1) 和 lis2 內容 是range(9,24,1)，計算兩個 list 的總合，將總和加入到各自 的 list 當中，即 lis1 做加總新增到 lis1 的最後一個元素 , lis2 做加總新增到 lis2 的最後一個元素。最後將兩個 list 由大到小排序，分別取出最後一個數字相乘輸出。 | 改寫描述 |
| 4-10 | 其中，%5.2f 指含整數位、小數位以及句 號，合計 5 位 | | | 其中，%5.2f 指顯示整數位5位、小數位以下2位 |  |
| 4-13 | (預設是 1；1 為遞減 1） | | | (預設是 1；-1 為遞減 1） |  |
| 4-23 | 1. 建立一個 list 為 range(1,15,2)，利用切片的方式取出前五個數字，匯入隨機的模組，隨機 1-5 產生一個整數 x 後，將 x 放入到 list ~~串列~~中，輸出 list 第 x 個數字。 2. 承上題，將 lis1 建立的前 5 個 list 由大到小排續後選出前三個數字，搭配上連續整數配對，使用迭代輸出全部資料。 | | | 1. 建立一個變數名稱為lis1的list ，其值以 range(1,15,2)建立，利用切片的方式取出前五個數字並取代lis1，匯入隨機的模組，隨機 1-5 產生一個整數 x 後，將 x 放入到 lis1 中，輸出lis1 , x 以及 lis1 第 x 個數字。 2. 承上題，將 lis1 建立的前 5 個 list 由大到小排續後選出前三個數字，搭配上連續整數配對，使用迭代輸出全部資料。 | 補充描述 |
| 5-20 | 3. 第 1 圈 sumi 為 0 加 i ~~為 1~~ 後，~~己~~ sumi 得到 1；第 2 圈後 sumi 為 1 加 i ~~為 2~~ 後sumi 得到 3，以此類推，最後一圈 sumi 的總和為 15（見圖5-7）。 | | | 3. 第 1 圈 i 為 1， sumi 為 0 加 i 後， sumi 得到 1；第 2 圈 i 為 2， sumi 為 1 加 i 後，sumi 得到 3，以此類推，最後一圈 sumi 的總和為 15（見圖5-7）。 |  |
| 5-25 | 程式第6行  print('%3d'%(num), end='') | | | print('%3d'%(num), end=””) | 單引號改雙引號 |
| 5-27 | 程式第4行  print('%d\*%d=%2d '%(i,j,i\*j), end='') | | | print('%d\*%d=%2d '%(i,j,i\*j), end=””) | 單引號改雙引號 |
| 5-33 | 程式第11行  print('%d\*%d=%2d '%(i,j,i\*j), end='') | | | print('%d\*%d=%2d '%(i,j,i\*j), end=””) | 單引號改雙引號 |
| 5-40 | 若要偵測某一列，可以在那一列最左邊（列號左邊）點選左鍵一次，會出現紅點 | | | 若要偵測某一列，可以在那一列最左邊（列號左邊）點選左鍵兩次，會出現紅點 |  |
| 5-42 | 本節只試算第 7 列的迴圈，第 1 圈 i 的初始值 5 之後進迴圈，因為停在第 8 列， 此時第 8 列並未執行 prodi=prodi\*i。因此，prodi 的值仍為初始值 1（第 7 列未執行過的結果），必須點選 後，才會實際執行第 8 列後回到迴圈，繼續下一圈。第 2 圈時 i 的值已更新為 4，prodi 更新為 5（第 1 圈執行過的結果（ 1\*5=5），第 2 圈尚未執行）（見圖5-19）以此類推，往下追蹤。 | | | 本節只試算第 7 列的迴圈，以m = 5 與n = 3為例，第 1 圈 i 的初始值 5 之後進迴圈，因為停在第 8 列， 此時第 8 列並未執行 prodi=prodi\*i。因此，prodi 的值仍為初始值 1（第 7 列未執行過的結果），必須點選 後，才會實際執行第 8 列後回到迴圈，繼續下一圈。第 2 圈時 i 的值已更新為 4，prodi 更新為 5（第 1 圈執行過的結果（ 1\*5=5），第 2 圈尚未執行）（見圖5-19）以此類推，往下追蹤。 |  |
| 5-60 | 1. …利用 for 迴圈撰寫一個程式，阿貴今年 25 歲，薪資所得每月 3 萬，薪資的年成長率為 3%，每月支出額為 1 萬 9，支出額每年以 2% 成長，求 28 歲時，阿貴的薪資所得、支出額、存款餘額為多少？結果以 print 函數顯示，如：薪資所得 123456 元。 | | | 1. …利用 for 迴圈撰寫一個程式，阿貴今年 25 歲，薪資所得每月3 萬，薪資的年成長率為 3%，每月支出額為 1 萬 9，支出額每年以 2% 成長，求 28 歲時，阿貴的薪資所得、支出額、存款餘額為多少？(假設每月初存入，每年底複利一次)   結果以 print 函數顯示，如：薪資所得 123456 元。 |  |
| 5-60 | 2. …小芳 20 年後就可以買到房子。 | | | 2. …小芳幾年後就可以買到房子。 |  |
| 6-25 | result=polyequ(sumple) | | | result=polyequ(sample) |  |
| 6-41 | 在主程式設定 x 等於 3 （見~~第~~第 6 列） | | | 在主程式設定 x 等於 3 （見第 6 列） |  |
| 6-74 | 8. …則最每次訂購量為 1200 單位。 | | | 8. 則最適每次訂購量為 1200 單位。 |  |
| 6-74 | 9. …延用【範例6-3】的資料 | | | 9. …延用【範例6-6-3】的資料 |  |
| 6-74 | 9. 單位折舊率 = | | | 9. 單位折舊率 = |  |
| 6-74 | 9. …0.45100,000 ＝ 45,000 | | | 9. …0.45 \* 100,000 ＝ 45,000 |  |
| 7-8 | |  | | --- | | with 運算式as 物件名稱: | | 指令1 | | ⋯ 指令n | | | | |  | | --- | | with 運算式as 物件名稱: | | 指令1 | | ⋯ 指令n | | 縮排調整 |
| 7-28 | 程式第7行  os.rename('./file/WangWei\_poetry\_1.txt', ' 辛夷塢王維.txt') | | | os.rename('./file/WangWei\_poetry\_1.txt', './file/辛夷塢王維.txt') | 使重新命名的檔案在同一個路徑下 |
| 7-28 | E\_7\_13操作：  因為此程式為一次性的操作，故要將以下檔案複製一份並改名：  WangWei\_poetry\_1 - origin.txt 複製後改成WangWei\_poetry\_1.txt  WangWei\_poetry\_2 - origin.txt 複製後改成WangWei\_poetry\_2.txt | | | |  |
| 7-32 | 程式第3行  import fix\_yahoo\_finance as yf | | | import yfinance as yf | 套件改名 |
| 7-32 | 在2018-02-09 在Python PyPi 官網公佈fix-yahoo-finance 修復套件，提供python 在yahoo finance 讀取股價日資料，公告網址如下： | | | 在2018-02-09 在Python PyPi 官網公佈fix-yahoo-finance 修復套件，並在2019年05月26日改名為yfinance，提供python 在yahoo finance 讀取股價日資料，公告網址如下： | 套件改名 |
| 7-32 | pip install fix\_yahoo\_finance --upgrade --no-cache-dir | | | pip install yfinance | 套件改名 |
| 7-33 | stockid=('2303', '2330', '3008', '2498', '2311', '2409', '2357', '2317')  writer=pd.ExcelWriter('./file/stocprice\_revised.xlsx')  print(type(stockid))  for i in range(0,len(stockid)):  sid=stockid[i]+'.tw'  df = pdr.get\_data\_yahoo(sid, start, end) df.to\_excel(writer,stockid[i]) | | | stockid=('2303', '2330', '3008', '2498', '2409', '2357', '2317')  writer=pd.ExcelWriter('./file/stocprice\_revised.xlsx')  print(type(stockid))  for i in range(0,len(stockid)):  sid=stockid[i]+'.tw'  try:  df = pdr.get\_data\_yahoo(sid, start, end)  except:  print(stockid[i] +"出錯")  continue  df.to\_excel(writer,stockid[i]) | 2311已經合併，並加入防錯機制(github程式碼也一併更新) |
| 7-33 | 匯入 pandas, pandas\_datareader.data, fix\_yahoo\_finance 與 datetime 等套件（見第  2 到 5 列）。第 7 行由於pandas\_datareader 的get\_data\_yahoo() 已棄用，並將會在未來  版本中刪除。所以我們使用fix\_yahoo\_finance 套件中提供的pdr\_override() 明確地覆  蓋它。 | | | 匯入 pandas, pandas\_datareader.data, yfinance 與 datetime 等套件（見第  2 到 5 列）。第 7 行由於pandas\_datareader 的get\_data\_yahoo() 已棄用，並將會在未來  版本中刪除。所以我們使用yfinance 套件中提供的pdr\_override() 明確地覆  蓋它。 | 套件改名 |
| 7-34 | 程式第3行  import fix\_yahoo\_finance as yf | | | import yfinance as yf | 套件改名 |
| 7-34 | 第7、20行  time.clock() | | | time.process\_time() | 套件更新 |
| 7-35 | 匯入 pandas，pandas\_datareader.data，fix\_yahoo\_finance，datetime 與 time 等套件（見第 2 到 6 列）。 | | | 匯入 pandas，pandas\_datareader.data，yfinance，datetime 與 time 等套件（見第 2 到 6 列）。 | 套件改名 |
| 7-37 | 程式第3行  import fix\_yahoo\_finance as yf | | | import yfinance as yf | 套件改名 |
| 7-39 | 第4、8行  time.clock() | | | time.process\_time() | 套件更新 |
| 7-40 | 程式第3行  import fix\_yahoo\_finance as yf | | | import yfinance as yf | 套件改名 |
| 7-48 | 程式第3行  import fix\_yahoo\_finance as yf | | | import yfinance as yf | 套件改名 |
| 7-48 | 程式10行  df1 = pdr.get\_data\_yahoo('2379.tw',start,end) | | | df1 = pdr.get\_data\_yahoo('1101.tw',start,end) | 更改標的，原標的抓不到資料 |
| 7-48 | 程式26行新增 | | | if u == 0:  u = 0.00000001  if d == 0 :  d = 0.00000001 | 加入防錯機制 |
| 7-48 | 程式41~44行新增  writer=pd.ExcelWriter('./file/2379.xlsx')  　df1.to\_excel(writer,'2379')  　workbook = writer.book  　worksheet = writer.sheets['2379'] | | | writer=pd.ExcelWriter('./file/1101.xlsx')  　df1.to\_excel(writer,'1101')  　workbook = writer.book  　worksheet = writer.sheets['1101']  writer.save() | 更改標的，原標的抓不到資料，修正沒有儲存的問題 |