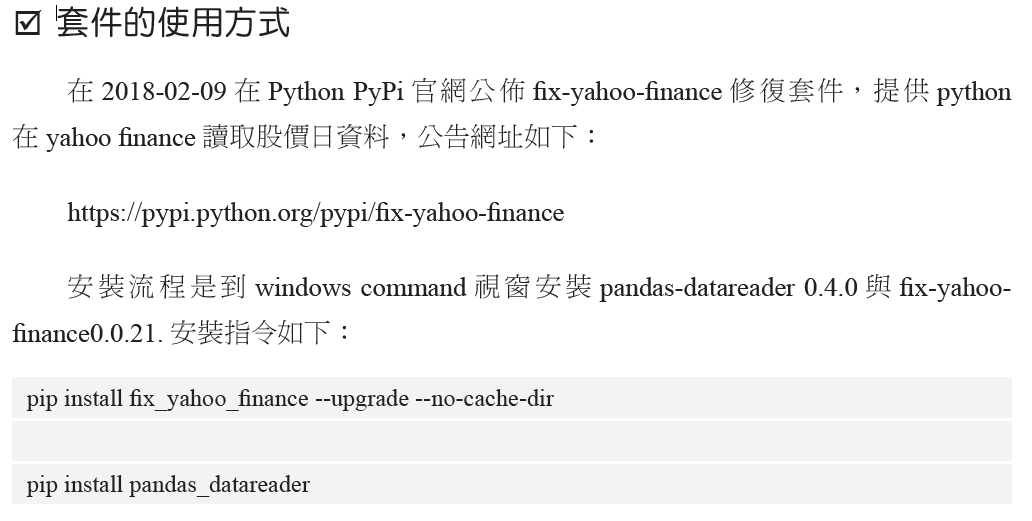
書籍勘誤資訊表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 書號 |  | 書名 | Python程式設計入門 金融商管實務案例 第二版 | | |
| 勘誤位置 | 勘誤前內容 | | | 勘誤後資訊 | 備註 |
| 目錄 | 加入GITHUB網址 | | | https://github.com/letylin/pyprogbook |  |
| 1-7 | scikit-learn 計算統計模型與統計檢定以及 matplotlib 做統計圖表等 | | | statsmodels 計算統計模型與統計檢定以及 matplotlib 做統計圖表等 |  |
| 1-20 | 圖片的Djanngo | | | Django |  |
| 2-19 | 圖2-26 　Spyde 更改工作目錄 | | | 圖2-26 　Spyder 更改工作目錄 |  |
| 2-20 | 圖2-27 　Spyde 線上求助 | | | 圖2-27 　Spyder 線上求助 |  |
| 3-21 | 格式  [0,0] ->[0][0] (如右圖) | | | C:\Users\pc_user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-21.tif |  |
| 3-26 | 格式  [0,0] ->[0][0]  (如右圖) | | | C:\Users\pc_user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-26.tif |  |
| 3-35 | In 12 id=dic2.get(2498)  13 id | | | In 12 value=dic2.get(2498)  13 value |  |
| 4-8 | slice() 可以幫串序型別的元素做切片，可以指定某串序從第 0 個元素開始（第 1 個參數）至結束（第 2 個參數）n 到 1 個元素做切片， | | | slice() 可以幫串列型別的元素做切片，可以指定某串列從第 0 個元素開始（第 1 個參數）至結束（第 2 個參數）n 減 1 個元素做切片， |  |
| 4-8 | 如 slice(0,10,2) 即第 0 元素開始到第 9 個元素做切 | | | 如 slice(1,10,2) 即第 1 元素開始到第 9 個元素做切 |  |
| 4-8、4-9程式 | 2~14行錯位應在地散行的a = 2跑到第二行去了  (見pdf) | | |  |  |
| 4-12 | 跳一列或換列的指令最常用是 \n（見第 8 到 11 列）。 | | | 跳一列或換列的指令最常用是 \n（見第 9 到 11 列）。 |  |
| 4-14 | 檢查傳入參數是否為可呼叫的 | | | 檢查傳入參數是否為可呼叫的抽象資料型別 |  |
| 4-23 | 輸出 ( 11/23  + 38 × 5) − 200)2 結果。 | | | 輸出 ( (11/23  + 38 × 5) − 200)2 結果。 |  |
| 4-23 | 1. 建立一個 list 為 range(1,15,2)，利切片的方式取出前五個數字 | | | 9. 建立一個 list 為 range(1,15,2)，利用切片的方式取出前五個數字 |  |
| 4-23 | 9. …隨機 1-5 產生一個整數 x 後  ，輸出 list 在 x 的數字。 | | | 9. …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。。 |  |
| 5-7 | 即將 num 加上 10 後再指派給 total（見第第 7 列）， | | | 即將 num 加上 10 後再指派給 total（見第 7 列）， |  |
| 5-9 | 表示 score 的值一定大於等於 60 同時小於 70，則執行指令區（僅一列）第 5 列，指派「丁等」給 grade。 | | | 表示 score 的值一定大於等於 60 同時小於 70，則執行指令區（僅一列）第 7 列，指派「丁等」給 grade。 |  |
| 5-10 | 若條件 2 為真則執行指令區（僅一列）第 7 列，指派「丙等」給 grade，以此類推。 | | | 若條件 3 為真則執行指令區（僅一列）第 9 列，指派「丙等」給 grade，以此類推。 |  |
| 5-15 | 第 1 列輸入猜測的數字，第 2 輸入使用者性別 | | | 第 2 列輸入猜測的數字，第 3 列 輸入使用者性別 |  |
| 5-17 | 遞增：range(0, 10, 2) 取出的元素順序為 0,2, 4, 6, 8，取到 end 的前一個；  遞減：range(10, 0, -2)-2 則是由大到小遞減，取出的元素順序為 10, 8, 6, 4, 2， 這些元素會逐一指派給計數器，共有 5 個元素，表示迴圈會執行 5 圈。 | | | 遞增：range(0, 10, 2) 取出的元素順序為 0, 2, 4, 6, 8，取到 end 的前一個；  遞減：range(10, 0, -2) 則是由大到小遞減，取出的元素順序為 10, 8, 6, 4, 2， 這些元素會逐一指派給計數器，共有 5 個元素，表示迴圈會執行 5 圈。 |  |
| 5-20 | 回到迴圈（見第 3 列） | | | 回到迴圈（見第 4 列） |  |
| 5-29 | 程式7、8列交換 | | |  |  |
| 5-30 | 當 num 的值不等於 0，即條件成立為真（見第 5 列） | | | 當 num 的值不等於 0，即條件成立為真（見第 6 列） |  |
| 5-44 | 進入指令區進行整數轉換（見第 3 列）； | | | 進入指令區進行整數轉換（見第 4 列）； |  |
| 5-44 | 若第 2 列判斷結果不成立，表示輸入的字串不是數值資料則跳過 3 到 6 列 | | | 若第 3 列判斷結果不成立，表示輸入的字串不是數值資料則跳過 4 到 6 列 |  |
| 5-45 | if score>=45 or score<9:  print('錯誤')  elif score>=37:  print('紅燈')  elif score>=31:  print('黃紅燈')  elif score>=22:  print('綠燈')  elif score>=16:  print('黃藍燈')  elif score>=9:  print('藍燈') | | | if score>45 or score<9:  print('錯誤')  elif score>37:  print('紅燈')  elif score>31:  print('黃紅燈')  elif score>22:  print('綠燈')  elif score>16:  print('黃藍燈')  else:  print('藍燈') |  |
| 5-49 | 應繳稅額為 15000 | | | 應繳稅額為 15000.0 |  |
| 5-50 | 因為， 要計 算離差平方和的平均（見第 8 列） | | | 因為， 要計算離差平方和的平均（見第 7 到 8 列） |  |
| 5-60 | 小芳 20 年後就可以買到房子。 | | | 小芳幾年後就可以買到房子。 |  |
| 5-61 |  | | | 請輸入A陣列  6 3 1  5 2 4  4 1 2  輸出範例：  9 4 4  8 4 8  9 2 6 |  |
| 5-63 | 2000 年不是閏年 | | | 2000 年是閏年 |  |
| 6-5 | 則 n 也會被指派 5 的值見第 1 列請對照圖6-2。 | | | 則 n 也會被指派 5 的值見第 2 列請對照圖6-2。 |  |
| 6-8 | 成功呼叫自訂函數 callfunc（ 見第 5 到 6 列）。 | | | 成功呼叫自訂函數 callfunc（ 見第 5 到 7 列）。 |  |
| 6-34 | result = {' 甲': 90, ' 丙': 70, ' 乙': 80} | | | (乙丙對調)  result = {' 甲': 90, ' 乙': 80, ' 丙': 70} |  |
| 7-10 | 或讀取光碟片第 7 章的檔案做為範例 | | | 或到本書github下載第 7 章的檔案做為範例 |  |
| 7-15 | print(line, end=”) | | | print(line, end='')  (雙單引號) |  |
| 7-32 | 程式改成如下表 | | |  |  |

|  |  |
| --- | --- |
| 1 | #E\_7\_15: 上網路抓取股價資料 |
| 2 | from pandas\_datareader import data as pdr |
| 3 | import fix\_yahoo\_finance as yf |
| 4 | import pandas as pd |
| 5 | import datetime |
| 6 | #下載資料起始日與股票代號 |
| 7 | yf.pdr\_override() # <== that's all it takes :-) |
| 8 | start = datetime.datetime(2016,1,1) |
| 9 | end = datetime.datetime(2018,3,1) |
| 10 | df = pdr.get\_data\_yahoo('2330.tw',start,end) |
| 11 | #日股價資料寫入excel檔 |
| 12 | writer=pd.ExcelWriter('./file/2330.xlsx') |
| 13 | df.to\_excel(writer,'Sheet1') |
| 14 | writer.save() |

|  |  |  |  |
| --- | --- | --- | --- |
| 7-32 | 匯入 pandas, pandas\_datareader.data 與 datetime 等套件（見第 2 到 4 列）。使用 datetime 套件內的 datetime() 方法設定起迄日（見第 6 到 7 列）。使用 web 套件別名的 DataReader() 方法讀取 yahoo 網站，股票代號為 2330.tw，依起迄時間下載資料到 df 資料框架（見第 8 列）。依 pd 別名套件的 ExcelWriter() 方法開啟一個 excel 檔，指派給檔案物件，名稱為 writer（ 見第 10 列）。將 df 依資料框架的方法 to\_excel() 將 df 寫 入到 writer（ 見第 11 列）。將 Excel 存檔（見第 12 列）。 | 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。使用 datetime 套件內的 datetime() 方法設定起迄日（見第 8 到 9 列）。使用 pdr 套件別名的 DataReader() 方法讀取 yahoo 網站，股票代號為 2330.tw，依起迄時間下載資料到 df 資料框架（見第 10 列）。依 pd 別名套件的 ExcelWriter() 方法開啟一個 excel 檔，指派給檔案物件，名稱為 writer（ 見第 12 列）。將 df 依資料框架的方法 to\_excel() 將 df 寫 入到 writer（ 見第 13 列）。將 Excel 存檔（見第 14 列）。 |  |
| 7-32 | 加入內容如下圖(同二部曲)  右方為未排版文字 | 套件的使用方式  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：  https://pypi.python.org/pypi/fix-yahoo-finance  安裝流程是到windows command 視窗安裝pandas-datareader 0.4.0 與fix-yahoo-finance0.0.21. 安裝指令如下：  pip install fix\_yahoo\_finance --upgrade --no-cache-dir  pip install pandas\_datareader |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 7-33、7-34 | 更換程式碼如下表 |  |  |

|  |  |
| --- | --- |
| 1 | #E\_7\_15: 上網路抓取股價資料 |
| 2 | from pandas\_datareader import data as pdr |
| 3 | import fix\_yahoo\_finance as yf |
| 4 | import pandas as pd |
| 5 | import datetime |
| 6 | import time |
| 7 | starttime = time.clock() |
| 8 | #下載資料起始日與股票代號 |
| 9 | yf.pdr\_override() # <== that's all it takes :-) |
| 10 | start = datetime.datetime(2016,1,1) |
| 11 | end = datetime.datetime(2018,3,1) |
| 12 | stockid=('2303', '2330', '3008', '2498', '2311', '2409', '2357', '2317') |
| 13 | writer=pd.ExcelWriter('./file/stocprice\_revised.xlsx') |
| 14 | print(type(stockid)) |
| 15 | for i in range(0,len(stockid)): |
| 16 | sid=stockid[i]+'.tw' |
| 17 | df = pdr.get\_data\_yahoo(sid, start, end) |
| 18 | df.to\_excel(writer,stockid[i]) |
| 19 | writer.save() |
| 20 | endtime = time.clock() |
| 21 | print(' 程式執行時間 = %d %s' %(round(endtime - starttime), ' 秒')) |

|  |  |  |  |
| --- | --- | --- | --- |
| 7-34 | 匯入 pandas，pandas\_datareader.data，datetime 與 time 等套件（見第 2 到 5 列）。 | 匯入 pandas，pandas\_datareader.data，fix\_yahoo\_finance，datetime 與 time 等套件（見第 2 到 6 列）。 |  |
| 7-34 | 設定 8 檔股票代號以 tuple 資料型別儲存指派給 stockid（ 見第 12 列）。迴圈設 定 i 為計數器自 0 到 stockid 的元素長度（見第 14 列）。進迴圈指令區，第 i 檔股票股票代號 stockid[i] 與 '.tw' 做字串合併（見第 15 列）。使用 DataReader 方法代入合併後股票代號與起迄日期抓取網路資料並指派給 df 資料框架（見第 16 列）。利用 to\_excel() 方法將 df 的資料寫入 writer 檔案的指定工作表名稱 stockid[i]（ 見第 17 列）。離開迴圈後儲存檔案（見第 19 列）。 | 設定 8 檔股票代號以 tuple 資料型別儲存指派給 stockid（ 見第 12 列）。迴圈設 定 i 為計數器自 0 到 stockid 的元素長度（見第 15 列）。進迴圈指令區，第 i 檔股票股票代號 stockid[i] 與 '.tw' 做字串合併（見第 16 列）。使用 DataReader 方法代入合併後股票代號與起迄日期抓取網路資料並指派給 df 資料框架（見第 17 列）。利用 to\_excel() 方法將 df 的資料寫入 writer 檔案的指定工作表名稱 stockid[i]（ 見第 18 列）。離開迴圈後儲存檔案（見第 19 列）。 |  |
| 7-36 | 更換程式碼如下表 |  |  |

|  |  |
| --- | --- |
| 1 | #RC\_7\_1: 上網路抓取股價資料畫高低圖 |
| 2 | from pandas\_datareader import data as pdr |
| 3 | import fix\_yahoo\_finance as yf |
| 4 | import pandas as pd |
| 5 | import datetime |
| 6 | #下載資料起始日與股票代號 |
| 7 | yf.pdr\_override() # <== that's all it takes :-) |
| 8 | start = datetime.datetime(2016,4,1) |
| 9 | end = datetime.datetime(2016,4,19) |
| 10 | df = pdr.get\_data\_yahoo('2330.tw',start,end) |
| 11 | #日股價資料寫入excel檔 |
| 12 | writer=pd.ExcelWriter('./file/2330.xlsx') |
| 13 | df.to\_excel(writer,'2330') |
| 14 | workbook = writer.book |
| 15 | worksheet = writer.sheets['2330'] |
| 16 | #畫高低圖 |
| 17 | chart = workbook.add\_chart({'type': 'stock'}) |
| 18 | chart.add\_series({'name': '=2330!$B$1','categories': '=2330!$A$2:$A$14','values': '=2330!$B$2:$B$14'}) |
| 19 | chart.add\_series({'name': '=2330!$C$1','categories': '=2330!$A$2:$A$14','values': '=2330!$C$2:$C$14'}) |
| 20 | chart.add\_series({'name': '=2330!$D$1','categories': '=2330!$A$2:$A$14','values': '=2330!$D$2:$D$14'}) |
| 21 | chart.set\_title ({'name': 'High-Low-Close'}) |
| 22 | chart.set\_x\_axis({'name': 'Date'}) |
| 23 | chart.set\_y\_axis({'name': 'Share price'}) |
| 24 | worksheet.insert\_chart('I2', chart) |
| 25 | writer.save() |

|  |  |  |  |
| --- | --- | --- | --- |
| 7-37 | 匯入 pandas，pandas\_datareader.data，datetime 套件（見第 2-4 列）。 | 匯入 pandas，pandas\_datareader.data，datetime 套件（見第 2到5 列）。 |  |
| 7-37 | 在活頁簿中新增一張統計圖，型別設為 stock，即為高低圖並指派給 chart（ 第 見15 列）。第 16 到 21 列同【範例 7-12】。 | 在活頁簿中新增一張統計圖，型別設為 stock，即為高低圖並指派給 chart（ 第 見17 列）。第 18 到 21 列同【範例 7-12】。 |  |
| 7-37 | 設定統計圖的標題（見第 22 列），第 23 到 25 列同【範例 7-12】。 | 設定統計圖的標題（見第 21 列），第 22 到 23 列同【範例 7-12】。 |  |
| 7-39、7-40 | 更換程式碼如下表 |  |  |

|  |  |
| --- | --- |
| 1 | #RC\_7\_3: 上網抓資料並畫K線圖 |
| 2 | #需要pip install mpl\_finance |
| 3 | import matplotlib.pyplot as plt |
| 4 | import matplotlib.dates as dts |
| 5 | import fix\_yahoo\_finance as yf |
| 6 | from mpl\_finance import candlestick\_ohlc |
| 7 | from pandas\_datareader import data as pdr |
| 8 |  |
| 9 | #下載資料起迄日, 日期格式與股票代號 |
| 10 | start = ('2016-4-1') |
| 11 | end = ('2016-4-25') |
| 12 | weekFormatter = dts.DateFormatter('%b %d') # 例如, Jan 03 2016 |
| 13 |  |
| 14 | yf.pdr\_override() # <== that's all it takes :-) |
| 15 | #以pandas\_datareader的get\_data\_yahoo抓取資料 |
| 16 | quotes = pdr.get\_data\_yahoo('AAPL', start, end) |
| 17 | #轉成陣列 |
| 18 | quotes = quotes.to\_records(convert\_datetime64=True).tolist() |
| 19 | #將陣列中，每個元素第一個欄位(日期)都改成數字型態，並縮成5個元素一組，以符合candlestick\_ohlc |
| 20 | quotes = [ (dts.date2num(d), o, h, l, c) for d, o, h, l, c, adj, v in quotes] |
| 21 |  |
| 22 | #若抓取的資料是空字串則離開系統 |
| 23 | if len(quotes) == 0: |
| 24 | raise SystemExit |
| 25 |  |
| 26 | #設定繪圖區域的格式化 |
| 27 | fig, ax = plt.subplots() |
| 28 | ax.xaxis\_date() |
| 29 | plt.setp(plt.gca().get\_xticklabels(), rotation=45, horizontalalignment='right') |
| 30 | #畫K線圖並顯示 |
| 31 | candlestick\_ohlc(ax, quotes, width=0.6) |
| 32 | plt.show() |

|  |  |  |  |
| --- | --- | --- | --- |
| 7-40 | 匯入 matplotlib.pyplot，matplotlib.dates 與 matplotlib.finance 相關套件模組。matplotlib.pyplot 用來設定繪圖格式與顯示；matplotlib.dates 用來設定繪圖的 時間軸；matplotlib.finance 用來繪製 K 線圖（見第 2 到 4 列）。 | 匯入 matplotlib.pyplot，matplotlib.dates 與 mpl\_finance 等相關套件模組。matplotlib.pyplot 用來設定繪圖格式與顯示；matplotlib.dates 用來設定繪圖的 時間軸；mpl\_finance 用來繪製 K 線圖（見第 2 到 7 列）。 |  |
| 7-40 | 設定起迄日期（見第 6 到 7 列）；設定 x 軸顯示的日期格式月日年（見第 8 列）；使用 quotes\_historical\_yahoo\_ohlc() 方法依起迄日期上 yahoo 網站抓蘋果股票資料，股票代號 : AAPL，結果指派給 quotes（ 見第 9 列）。 | 設定起迄日期（見第 10 到 11 列）；設定 x 軸顯示的日期格式月日年（見第 12 列）；使用 get\_data\_yahoo () 方法依起迄日期上 yahoo 網站抓蘋果股票資料，股票代號 : AAPL，結果指派給 quotes（ 見第 16 列），取得資料後將資料傳換陣列型態(見第18列)，將陣列中，每個元素第一個欄位(日期)都改成數字型態，並縮成5個元素一組，以符合candlestick\_ohlc(見第20列)。 |  |
| 7-40 | 使用 if 指令判斷上網抓的資料若是空白資料（len() 方法計算的長度為 0）， 則啟動例外機制離開程式（見第 11 到 12 列） | 使用 if 指令判斷上網抓的資料若是空白資料（len() 方法計算的長度為 0）， 則啟動例外機制離開程式（見第 23到 24 列） |  |
| 7-40 | 新增一張空白繪圖指派給 fig 與 ax（ 見第 14 列）。指定 x 軸為日期資料（見第 15 列）。設定繪圖的格式，包含 x 軸日期資料轉換角度為 45 度，水平對齊等（見第 16 列）。 | 新增一張空白繪圖指派給 fig 與 ax（ 見第 27 列）。指定 x 軸為日期資料（見第 28 列）。設定繪圖的格式，包含 x 軸日期資料轉換角度為 45 度，水平對齊等（見第 29 列）。 |  |
| 7-40 | 使用 candlestick\_ohlc() 方法代入格式設定 ax 與資料 quotes 以及設定 K 線的 寬度畫 K 線圖（見第 18 列）；顯示 K 線圖（見第 19 列）。 | 使用 candlestick\_ohlc() 方法代入格式設定 ax 與資料 quotes 以及設定 K 線的寬度畫 K 線圖（見第 31 列）；顯示 K 線圖（見第 32 列）。 |  |
| 7-46、7-47 | 更換程式碼如下表 |  |  |

|  |  |
| --- | --- |
| 1 | #RC\_7\_5: 計算技術指標- 相對強弱指標RSI |
| 2 | from pandas\_datareader import data as pdr |
| 3 | import fix\_yahoo\_finance as yf |
| 4 | import pandas as pd |
| 5 | import datetime |
| 6 |  |
| 7 | yf.pdr\_override() # <== that's all it takes :-) |
| 8 | start=datetime.datetime(2013,1,1) |
| 9 | end=datetime.datetime(2015,12,31) |
| 10 | df1 = pdr.get\_data\_yahoo('2379.tw',start,end) |
| 11 | roi=df1[df1.columns[0]].pct\_change() |
| 12 |  |
| 13 | def rsi(n,roi): |
| 14 | rsiv=[] |
| 15 | for i in range (n, roi.count()): |
| 16 | ups=0 |
| 17 | downs=0 |
| 18 | u=0 |
| 19 | d=0 |
| 20 | ud=0 |
| 21 | for j in range(i-n+1,i+1): |
| 22 | if roi[j]>=0.00000000: |
| 23 | u=u+roi[j] |
| 24 | else: |
| 25 | d=d+roi[j] |
| 26 | ups=u/n |
| 27 | downs=abs(d/n) |
| 28 | ud=ups/downs |
| 29 | rsiv.append(abs(100-(100/(1+ud)))) |
| 30 | return rsiv |
| 31 |  |
| 32 | for n in range(5, 11, 5): |
| 33 | rsi\_data=[] |
| 34 | rsi\_data=rsi(n,roi)#呼叫rsi自訂函數 |
| 35 | Colname='rsi'+str(n) |
| 36 | df2=pd.DataFrame(columns=[Colname],index=df1.index) |
| 37 | for i in range(n,len(df1.index)-1): |
| 38 | df2['rsi'+str(n)][i+1]=rsi\_data.pop(0) |
| 39 | df1=pd.concat([df1,df2],axis=1) |
| 40 |  |
| 41 | writer=pd.ExcelWriter('./file/2379.xlsx') |
| 42 | df1.to\_excel(writer,'2379') |
| 43 | workbook = writer.book |
| 44 | worksheet = writer.sheets['2379'] |

|  |  |  |  |
| --- | --- | --- | --- |
| 7-48 | 匯入 pandas, numpy, pandas\_datareader.data 與 datetime 相關套件模組（見第2 到5 列）。 | 匯入 pandas , pandas\_datareader.data, fix\_yahoo\_finance 與 datetime 相關套件模組（見第2 到5 列）。 |  |
| 7-48 | 上網下載 2379（瑞昱半導體）使用 DataReader() 方法讀取 yahoo 網路股價資料，用法參考【實務案例 7-1】，見第 7 到 9 列。 | 上網下載 2379（瑞昱半導體）使用 DataReader() 方法讀取 yahoo 網路股價資料，用法參考【實務案例 7-1】，見第 8 到 10 列。 |  |
| 7-48 | 將 rsi\_data 插入到 df2 資料框架（見第39 列）。將原 df1 的敘述統計與 df2 做合併（見第40 列）。 | 將 rsi\_data 插入到 df2 資料框架（見第38列）。將原 df1 的敘述統計與 df2 做合併（見第39 列）。 |  |
| 8-11 | 第 5 到 6 列的寫法待下一節說明 | 第 5 到 8 列的寫法待下一節說明 |  |
| 8-15 | 其回傳值是物件實體（見第 8 列）。 | 其回傳值是物件實體（見第 7 列）。 |  |
| 8-19 | 程式進入點在第 10 列）， | 程式進入點在第 10 列， |  |
| 8-23 | 實際是執行程式碼的第 30 到 32 列） | 實際是執行程式碼的第 30 到 32 列 |  |
| 8-33 | 。計算累加功能見 7 到 9 列） | 。計算累加功能見 7 到 9 列 |  |
| 封底 | 圖片的Djanngo | Django |  |

排版問題

|  |  |  |
| --- | --- | --- |
| 3-17 | int, f loat, string, tuple, list, set, dict 等， | int, float, string, tuple, list, set, dict 等， |
| 4-10 | print 函數依參數內容格式化輸出，最常用的格 式化輸出的形式（見表4-8）。使用格 式說明如下： | print 函數依參數內容格式化輸出，最常用的格式化輸出的形式（見表4-8）。使用格 式說明如下： |
| 4-13 | 整數為不 可迭代； | 整數為不可迭代； |
| 4-14 | 整數與串列等 資料型別是可以被呼叫的（見第 7 到 10 列）。 | 整數與串列等資料型別是可以被呼叫的（見第 7 到 10 列）。 |
| 4-15 | 若輸入 help() 沒有傳入任何參 數，則會自動進入互動式線上求助系統（見圖 4-1）。在 help> 後輸入待查詢的函數名 稱 | 若輸入 help() 沒有傳入任何參數，則會自動進入互動式線上求助系統（見圖 4-1）。在 help> 後輸入待查詢的函數名稱 |
| 5-2 | 下半部灰色文字內縮一個TAB  如下：  if判斷條件式：  指令1  指令2  ...  指令n |  |
| 5-5 | 灰色文字內縮一個TAB  如下：  if判斷條件式：  指令1  指令2  ...  指令n  else：  指令1  指令2  ...  指令n |  |
| 5-7 | 灰色文字內縮一個TAB  如下：  if 判斷條件 1：  指令1 　當條件1 成立時執行本區指令  ⋯  指令n  elif 判斷條件 2：  指令1 　當條件2 成立時執行本區指令  ⋯  指令n |  |
| 5-8 | 灰色文字內縮一個TAB  如下：  elif 判斷條件 3：  指令1 　當條件3 成立時執行本區指令  ⋯  指令n  else：  指令1 　當以上條件式均不成立時執行本區指令  ⋯  指令n |  |
| 5-14 | 10~23行程式層級對齊 (各行最前面少一個空格，如右圖) |  |
| 5-29 | 第10列對齊第9列(如右圖) |  |
| 5-30 | 即可進入迴圈 內執行指令區的程式碼（見第 6 到 8 列）。 | 即可進入迴圈內執行指令區的程式碼（見第 6 到 8 列）。 |
| 5-38 | 離開迴 圈印出 area（見第 9 列）。 | 離開迴圈印出 area（見第 9 列）。 |
| 5-39 | 第8列對齊第10列 |  |
| 5-42 | 程式對齊如右圖 |  |
| 5-49 | 10~16行對齊第9行(如右圖) |  |
| 5-50 | 因為， 要計 算離差平方和的平均（見第 8 列） | 因為， 要計算離差平方和的平均（見第 7 到 8 列） |
| 5-51 | 3~7列程式排版層級如下  for x in range(6):  for y in range(11):  for z in range(51):  if 10\*x+5\*y+z == 50:  count = count + 1 |  |
| 6-5 | 程式第10行對齊第9行 |  |
| 6-8 | 程式12、13、15、18、19、21行對齊第8行(如右圖) |  |
| 6-20 | 程式10、11對齊第9行(如右圖) |  |
| 6-29 | 12、14、16行對齊第7行(如右圖) |  |
| 6-29 | 13、15行對齊第8行(如右圖) |
| 6-29 | 10、11行對齊第9行(如右圖) |
| 6-30 | 縮排不明顯(格式如右圖) |  |
| 6-36 | 5、7行縮排  排版如下  if n==1:  sex='先生'  else:  sex='小姐' |  |
| 6-41 | 程式第10行對齊第9行(如右圖) |  |
| 6-45 | 程式第10行對齊第9行(如右圖) |  |
| 6-50 | 程式第10行對齊第9行(如右圖) |  |
| 6-58 | 程式第10、11行對齊第9行(如右圖) |  |
| 6-70 | 程式第10、11行對齊第9行(如右圖) |  |
| 7-13 | 縮排一階  排版如下：  with open ('oil\_write.txt', 'w') as fout:  for line in fin.readlines():  if line.startswith('日'): #判断表頭的第1個字  continue |  |
| 7-16 | 程式第10行對齊第9行(如右圖) |  |
| 7-18 | 第9行字體大小不一(格式如右圖) |  |
| 7-20 | 程式10到12行對齊第9行(如右圖) |  |
| 7-21 | 程式10到17行對齊第9行(如右圖) |  |
| 7-25 | 程式10到19行對齊第9行(如右圖) |  |
| 7-29 | 程式11、14行對齊第9行(如右圖) |  |
| 7-30 | 程式第10行對齊第9行(如右圖) |  |
| 8-8 | 程式10、13、14、15、16行對齊第9行(如右圖) |  |
| 8-8 | 程式11、12行對齊第8行(如右圖) |
| 8-10 | 程式10、14、17對齊第5行 | 如下圖 |
| 8-10 | 程式11、12、15、16行對齊第8行 |
| 8-10 | 程式19到28行對齊第3行 |
|  | | |
| 8-12 | 程式5到16行對齊第4行(如右圖) |  |
| 8-16 | 程式10、11、13、14對齊第7行(如右圖) |  |
| 8-16 | 程式第12行對齊第9行(如右圖) |
| 8-18 | 程式10到14行對齊第2行(如右圖) |  |
| 8-21 | 程式25、26行縮排，排版如下  class CheckingAccount(CreateBankAccount):  def \_\_init\_\_(self, id, name):  super().\_\_init\_\_(id, name) # 呼叫父類別\_\_init\_\_()  self.overdraftlimit = 30000 #透支額度  #考慮透支額度的提款方法  def withdraw(self, amount):  if amount <= self.balance + self.overdraftlimit:  self.balance -= amount  else:  raise ValueError('超出信用額度')  return self.balance | 整體排版如下 |
| 8-21 | 程式19行對其第3行 |
| 8-21 | 程式20、24、30行對齊第5行 |
| 8-21 | 程式21、22、25、27、29、31行對齊第8行 |
|  | | |
| 8-24 | 程式11到14行對齊第8行(如右圖) |  |
| 8-24 | 程式第10行對齊第7行(如右圖) |
| 8-26 | 程式11、15、19、20對齊第7行 | 整體排版如下 |
| 8-26 | 程式12、16、21對齊第8行 |
| 8-26 | 程式10、13、14、17、18對齊第9行 |
|  | | |
| 8-30 | 程式第10行對齊第9行(如右圖) |  |
| 8-33 | 程式11、14、16、18行對齊第6行 |  |
| 8-33 | 程式12、13、15、17、19、20行對齊第9行 | 整體排版如下 |
|  | | |
| 8-46 | 程式11到14行對齊第2行 | 整體排版如下 |
| 8-46 | 程式10、15、17、20行對齊第8行 |
| 8-46 | 程式16、18、19行對齊第9行 |
|  | | |
| 8-50 | 10到20行對齊第9行(如右圖) |  |
| 8-51 | 10到12行對齊第7行(如右圖) |  |