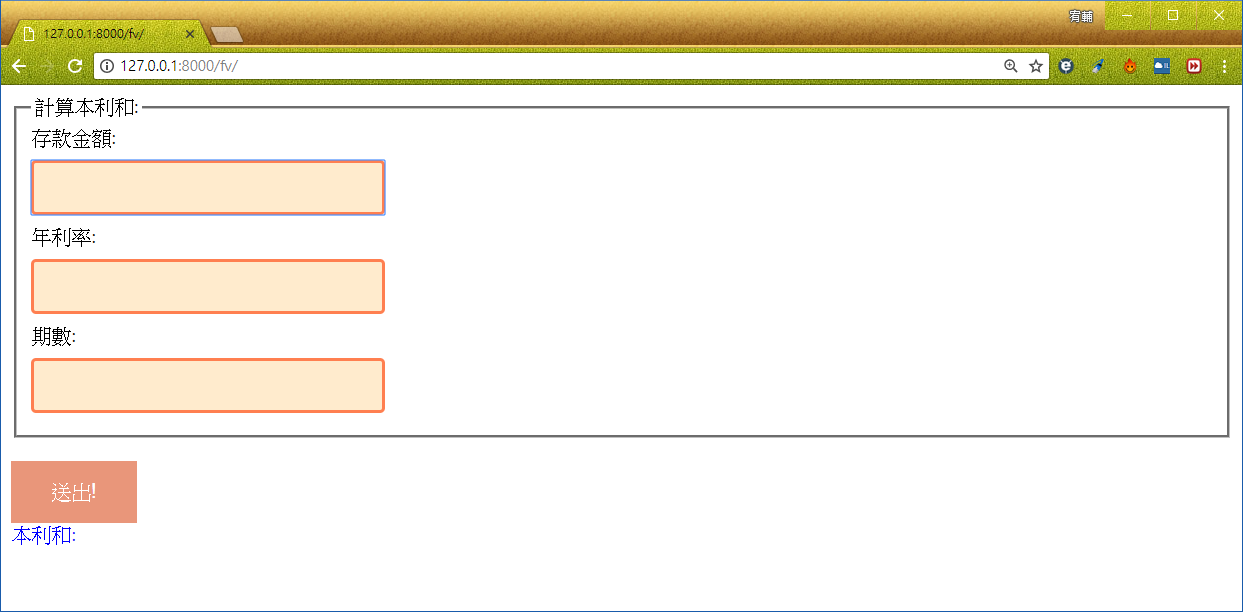
書籍勘誤資訊表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 書號 | MP31805 | 書名 | Python網頁程式交易APP實作：Web + MySQL + Django(第二版) | | |
| 勘誤  位置 | 勘誤前內容 | | | 勘誤後資訊 | 備註 |
| 1-5頁 | 第一段第一行的Hakon | | | Håkon | 與表格名稱一致 |
| 1-11頁 | 等待輸入要安裝的新套件名稱：Markwon preview後按Enter | | | 等待輸入要安裝的新套件名稱：Markdown preview後按Enter |  |
| 1-11頁 | (圖文字) Markdwon preview套件安裝成功的畫面 | | | (圖文字) Markdown preview套件安裝成功的畫面 |  |
| 1-13頁 | (1)Tools功能表下選Build System下的的New Build System | | | (1)Tools功能表下選Build System下的New Build System |  |
| 3-10頁 | strobj.length() | | | strobj.length | 刪除小括孤 |
| 3-12頁 | E\_3\_4.html第7列修正  x = hello | | | x = Hello |  |
| 3-17頁 | |  | | --- | | 運算子 | | or | | and | | not | | | | 3-17頁表格如下   |  |  | | --- | --- | | 運算子 | 功能說明 | | ｜｜ | 或（or）的運算，其中一個運算元為真，即為真。 | | && | 且（and）的運算，所有運算元皆為真，才為真。 | | ! | 非（not）的運算，把真轉為假；把假轉為真。 | | ^ | 互斥或（xor）的運算，其中一個運算元為真，另一個運算元為假，即為真。 | |  |
| 3-18頁 | |  |  |  | | --- | --- | --- | | 運算子 | 功能說明 | 運算符號與變數 | | = | a=9;b=3;s=true;t=false; |  | | += | a+=b | a=a+b | | -= | a-=b | a=a-b | | \*= | a\*=b | a=a\*b | | /= | a/=b | a=a/b | | %= | a%=b | a=a%b | | >>= | a>>=b | a=a>>b | | <<= | a<<=b | a=a<<b | | &= | s&=t | s=s&t | | ｜= | s｜=t | s=｜t | | ^= | s^=t | s=s^t | | | | |  |  |  | | --- | --- | --- | | 運算子 | 功能說明 | 運算符號與變數 | | = | a=9;b=3;s=true;t=false; |  | | += | a+=b | a=a+b | | -= | a-=b | a=a-b | | \*= | a\*=b | a=a\*b | | /= | a/=b | a=a/b | | %= | a%=b | a=a%b | | &= | s&=t | s=s&t | | ｜= | s｜=t | s=｜t | | ^= | s^=t | s=s^t | |  |
| 3-19頁 | 程式12、13行刪除 | | |  |  |
| 當執行完第6列a+=b的時候 | | | 當執行完第7列a+=b的時候 |  |
| 4-5頁 | 以鍵盤輸入華氏（或攝氏）溫度，計算攝氏（或華氏）溫度。  公式：攝氏溫度=（5/9）\*（華氏溫度-32）。（注意：小數點取一位）  華氏溫度=（9/5）\*（攝氏溫度）+32。（注意：小數點取一位） | | | 以鍵盤輸入華氏（或攝氏）溫度，計算攝氏（或華氏）溫度。  公式：攝氏溫度=（5/9）\*（華氏溫度-32）。  華氏溫度=（9/5）\*（攝氏溫度）+32。 |  |
| 4-13頁 | 第8列至第26列為switch case的邏輯判斷式 | | | 第8列至第27列為switch case的邏輯判斷 | \* |
| 4-15頁 | 程式31行  <br/> | | | <br> |  |
| 4-26頁 | 程式第4行  <h3> 重覆產生亂數</h3> | | | <h3> 重覆產生亂數（0 至 6 的整數）0時結束</h3> |  |
| 4-27頁 | 程式碼7、11行  = Math.floor(Math.random() \* 6); | | | = Math.floor(Math.random() \* 7); |  |
| 第7 列的Math.floor(Math.random() \* 6) 可以取得一個 0 至 5 的隨機亂數。 | | | 第7 列的Math.floor(Math.random() \* 7) 可以取得一個 0 至 6 的隨機亂數。 |  |
| 5-6頁 | (圖片)  var n = 10;  ("累加1到10="+sumfunc(n)); | | | var num = 10;  ("累加1到10="+sumfunc(num)) |  |
|  | | | | |
| 5-25頁 | 說明第二段：原本將陣列顯示直接打陣列名… | | | 原本將陣列顯示直接打陣列變數名稱， |  |
| 5-30頁 | 表格Math.pow()說明 - 回傳某數值開n次方根的值 | | | 回傳某數值n次方的值 |  |
| 6-20頁 | 也可以設定為一行：  div{  margin:90px 90px 50px 50px;  } | | | 也可以設定為一行：  div{  margin:90px 50px 90px 50px;  }/\*順序為上右下左\*/ |  |
| 6-23頁 | <option value=0.1>貝萊德世界黃金基金</option>  <option value=0.2>安聯收益成長基金</option>  <option value=0.3>聯博-中國時機基金</option> | | | <!--基金價格皆以2018/04/30之價格為範例 -->  <option value=27.78>貝萊德世界黃金基金</option>  <option value=9.03>安聯收益成長基金</option>  <option value=58.65>聯博-中國時機基金</option> |  |
| 6-24頁 | (程式)  document.getElementById("result").innerHTML = w\*10000; | | | var amount = document.getElementById("amt").value;  document.getElementById("result").innerHTML = amount/w; | 函數內的程式碼要內縮對擠 |
|  | | | | |
| 7-4頁 | 7-4頁圖中<p>下層的<div>改成<u> | | |  |  |
|  | | | | |
| 7-6頁 | E\_7\_2.css要加入前4列指令 | | | ul {  list-style-type: square;  color: Brown;  } |  |
|  | | | | |
| 7-12頁 | <p id = result></p> | | | 第14列刪除 |  |
| 7-13頁 | var selclass = document.querySelector(classname);  alert( "class 的第一個元素是" + classname); | | | 更改如下: (要照原來縮排格式置換)  var selclass = document.querySelector(classname).tagName + "標籤";  alert( "class的第一個元素是" + selclass); |  |
| 8-6頁 | 優點第三項：可支援同個標籤設定多個相同事件，無法應用在複雜的環境(刪除) | | | 可支援同個標籤設定多個相同事件。 |  |
| 8-9頁 | 程式13行  <p> 年利率：</p> | | | <p> 年利率(%)：</p> |  |
| 8-10頁 | 最後false 為程式執行由上而下，反之為true 時，就是由下至上。 | | | 最後false 為程式執行由上而下，反之為true 時，就是由上至下。 |  |
| 8-16頁 | 當使用者點選詩的內容時會觸動z和x。 | | | 當使用者點選詩的內容時會觸動<p>和<div>。 |  |
| 8-17頁 | 8-1-4 修改為capute事件傳播方法(標題) | | | 8-1-4 修改為capture事件傳播方法 |  |
| 9-15頁 | 點選照相功能Download plot as png | | | 點選照相功能Download plot as a png |  |
| 9-26頁 | 2.當收盤價低於同時MA5,MA10,MA20賣出 | | | 2.當收盤價同時低於MA5,MA10,MA20賣出 |  |
| 10-8頁 | 前端表單網頁：設計一個可以讓使用者瀏覽存款試算的網頁，可以依第 2 章 E\_2\_2.html 為例。 | | | 前端表單網頁：設計一個可以讓使用者瀏覽存款試算的網頁，可以依第 2 章 E\_8\_1.html 為例。 |  |
| 10-9頁 | 關閉Anancoda相關程式，重新啟動。 | | | 關閉Anaconda相關程式，重新啟動。 |  |
| 10-19頁 | 樹狀圖(倒數第三行)staic/ | | | static/ |  |
| 並將要加入的靜態檔realdata.js | | | 並將要加入的靜態檔E\_10\_1.js |  |
| 10-20頁 | 範例10-1程式第6行  url(r'^go/$', views.pylinkweb),# 新增這一列 | | | url(r'^gogo/$', views.pylinkweb),# 新增這一列 |  |
| 10-21頁 | views.py程式第11行  fv=str(pv\*((1+i)\*\*n)) | | | fv=str(round(pv\*((1+i)\*\*n),2)) |  |
| E\_10\_1.html程式第7行新增 | | | <link rel="stylesheet" type="text/css" href="{% static "myapp/E\_10\_1.css" %}"> |  |
| 程式行號28到37行 | | | 改成8到17行 |  |
| 10-22頁 | 程式行號38到42行 | | | 改成18到22行 |  |
| E\_10\_1.html程式第20行  <p>本利和:<span id="result"></span></p> | | | <br><p>本利和:<span id="result"></span></p> |  |
| 第5列為載入靜態資料擋...  第6列為設定讀取的靜態擋... | | | 第5列為載入靜態資料檔...  第6列為設定讀取的靜態檔... |  |
| 10-23頁 | 新增E\_10\_1.css | | | 如下表 |  |

|  |  |
| --- | --- |
| 1 | input[type=text] { |
| 2 | width: 30%; |
| 3 | padding: 12px 20px; |
| 4 | margin: 8px 0; |
| 5 | box-sizing: border-box; |
| 6 | border: 2px solid Coral; |
| 7 | border-radius: 4px; |
| 8 | background-color:BlanchedAlmond; |
| 9 | color: BlueViolet; |
| 10 | } |
| 11 |  |
| 12 | button { |
| 13 | background-color: DarkSalmon; |
| 14 | border: none; |
| 15 | color: white; |
| 16 | padding: 15px 32px; |
| 17 | text-align: center; |
| 18 | text-decoration: none; |
| 19 | display: inline-block; |
| 20 | font-size: 16px; |
| 21 | } |
| 22 |  |
| 23 | p { |
| 24 | display: inline; |
| 25 | color: blue; |
| 26 | } |

|  |  |
| --- | --- |
| 10-23頁 | 執行結果圖片換成下圖 |



|  |  |  |  |
| --- | --- | --- | --- |
| 11-3頁 | 安全性：…並且需與資料庫管理師不同人以確保資料庫安全性 | 並且使用者與應用系統開發者需與資料庫管理師三個要不同人以確保資料庫安全性 |  |
| 11-6頁 | 外部層：…所以外部層的使用者通常被授權新增、修改以及刪除動作，只有查詢的權限。 | 外部層：…所以外部層的使用者通常不被授權新增、修改以及刪除動作，只有查詢的權限。 |  |
| 11-9頁 | 主從式網路架構見下圖是由主從式架構結合World Wide Web | 多層式網路架構見下圖是由主從式架構結合World Wide Web |  |
| 11-10頁 | (圖片文字)互動式網路架構 | (圖片文字) 多層式網路架構 |  |
| 11-12頁 | 分散式資料庫處理架構見下圖... | 分散式資料庫處理架構...(刪除紅字) |  |
| 11-14頁 | 張三買了華碩、台積電與台塑的股票，張三即為投資持有人；李四持有台積電與台塑的股票；趙二也買進台積電與台塑；王五則持有華碩、台塑兩檔股票。 | 張三買了華碩、台積電與台塑的股票，張三即為投資持有人；李四持有台積電與台塑的股票；趙二也買進台積電與台塑；王五則持有華碩、台塑兩檔股票。華碩公司也可以持有台塑公司的股票。 | 圖片更動如下 |
|  | | |
| 12-2頁 | 到此下載網址取得Server 端的安裝軟體。https://dev.mysql.com/downloads/ installer/， 點選「MySQL Community Server 5.7.18」 | 到此下載網址取得Server 端的安裝軟體。https://dev.mysql.com/downloads/ installer/，目前已經更新到MySQL 8.0版本，但仍有相容性之問題，故建議安裝MySQL 5.7版本，點選「MySQL Community Server 5.7.18」 |  |
| 12-22頁 | ...Tables下點選右鍵Create table... | ...Tables下點選右鍵Create Table... |  |
| 13-7頁 | 在連線MsSQL 之前，先測試建立一個連結(connection) | 在連線MySQL 之前，先測試建立一個連結(connection) |  |
| 13-19頁 | Views.py 程式碼更新 | 程式碼如下表 |  |

|  |  |
| --- | --- |
| 1 | from myapp.models import Company |
| 2 | from django.shortcuts import render\_to\_response |
| 3 | from django.shortcuts import render |
| 4 | from django.shortcuts import redirect |
| 5 |  |
| 6 | def company(request): |
| 7 | company = Company.objects.all() |
| 8 | stockid = Company.\_meta.get\_field('stockid').column |
| 9 | abbreviation = Company.\_meta.get\_field('abbreviation').column |
| 10 | url = Company.\_meta.get\_field('url').column |
| 11 | industryname = Company.\_meta.get\_field('industryname').column |
| 12 | return render\_to\_response('company.html',locals()) |

|  |  |  |  |
| --- | --- | --- | --- |
| 13-20頁 | 13-4-3節前加入一段說明 | 重要提醒: Views.py程式碼共有66列， 第13到66列分別於本章後續章節個別說明。 |  |
| 14章GetGoogleFinance.無法執行，改用GetYahooFinance方式如下 | | | |

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

範例：網路爬蟲股價資料程式，透過Yahoo finance取得股價資料。將上述範例google網路爬蟲的部分，更改成透過fix\_yahoo\_finance套件讀取股價資料。

程式碼檔名：class\_GetYahooFinance.py(放入同目錄下ProgramTrade資料夾下)

--------------------------程式開始-----------------------------------------

|  |  |
| --- | --- |
| 1 | # -\*- coding: utf-8 -\*- |
| 2 | """ |
| 3 | Created on Thu Jun 22 09:11:04 2017 |
| 4 | 1.自Yahoo finance抓取台股日資料(任意期間) |
| 5 | 2.存入EXCEL檔案 |
| 6 | @author: 林萍珍 |
| 7 | """ |
| 8 | import pandas as pd |
| 9 | import datetime |
| 10 | import os.path |
| 11 | import pymysql.cursors |
| 12 | import xlrd |
| 13 | from pandas\_datareader import data as pdr |
| 14 | import fix\_yahoo\_finance as yf |
| 15 |  |
| 16 | class GetYahooFinance(): |
| 17 | def \_\_init\_\_(self,stkno,startdate,enddate,refresh=True): |
| 18 | #string to datetime |
| 19 | if(isinstance(startdate, str) == True or isinstance(enddate, str) == True): |
| 20 | startdate=datetime.datetime.strptime(startdate,'%Y-%m-%d') |
| 21 | enddate=datetime.datetime.strptime(enddate,'%Y-%m-%d') |
| 22 | self.sdate=startdate |
| 23 | self.edate=enddate |
| 24 | #datetime to string |
| 25 | S\_date=datetime.datetime.strftime(startdate,'%Y%m%d') |
| 26 | E\_date=datetime.datetime.strftime(enddate,'%Y%m%d') |
| 27 | self.fn=str(stkno)+'\_'+S\_date+'\_'+E\_date+'.xlsx' |
| 28 | self.stkno=stkno |
| 29 | self.refresh=refresh |
| 30 |  |
| 31 | def savetoexcel(self,data,sheetname): |
| 32 | ''' |
| 33 | 若資料不存在，則不寫入檔案 |
| 34 | ''' |
| 35 | if len(data) != 0: |
| 36 | writer=pd.ExcelWriter(self.fn) |
| 37 | data.to\_excel(writer,sheetname,index=False) |
| 38 | writer.save() |
| 39 |  |
| 40 | def getstock(self,asc=False): |
| 41 | if not self.refresh and os.path.exists(self.fn): #不重置 &檔案已存在 |
| 42 | Data=pd.read\_excel(self.fn) |
| 43 | Data[['open','high','low','close','volume']]=\ |
| 44 | Data[['open','high','low','close','volume']].astype(float) |
| 45 | Data[['dates']]=Data[['dates']].astype(object) |
| 46 | else: |
| 47 | s\_yy=self.sdate.year |
| 48 | s\_mm='{:%b}'.format(self.sdate) |
| 49 | s\_dd=self.sdate.day |
| 50 | e\_yy=self.edate.year |
| 51 | e\_mm='{:%b}'.format(self.edate) |
| 52 | e\_dd=self.edate.day |
| 53 | try: |
| 54 | yf.pdr\_override() # <== that's all it takes :-) |
| 55 | Data = pdr.get\_data\_yahoo(self.stkno+'.tw',self.sdate,self.edate) |
| 56 | del Data['Adj Close'] |
| 57 | Data = Data.reset\_index() |
| 58 | print(Data) |
| 59 | except Exception as err: |
| 60 | print(err) |
| 61 | if len(Data) != 0: #有取得資料才進行轉換 |
| 62 | Data.columns=['dates','open','high','low','close','volume'] |
| 63 | Data[['open','high','low','close','volume']]=\ |
| 64 | Data[['open','high','low','close','volume']].astype(float) |
| 65 |  |
| 66 | if asc==True: |
| 67 | Data=Data.sort\_values(by=['dates'],ascending=True) |
| 68 | return Data |
| 69 |  |
| 70 | if \_\_name\_\_ == "\_\_main\_\_": |
| 71 | ''' |
| 72 | Get data from Yahoo finance |
| 73 | ''' |
| 74 | stkno='1102' |
| 75 | start = datetime.datetime(2015,9,1) |
| 76 | end = datetime.datetime(2017,10,1) |
| 77 | #增加參數，True則重新抓取資料， |
| 78 | # False則檢查檔案存在與否，若不存在再上網抓取資料。 |
| 79 | Gdata=GetYahooFinance(stkno,start,end,False) |
| 80 |  |
| 81 | Data =Gdata.getstock(True) |
| 82 | ''' |
| 83 | Save data to excel |
| 84 | ''' |
| 85 | if len(Data) != 0: |
| 86 | Gdata.savetoexcel(Data,stkno) |
| 87 | else: |
| 88 | print('{} 抓取的資料不存在!'.format(stkno)) |

--------------------------程式結束-----------------------------------------

程式碼說明：

大部分程式與上述GetGoogleFinance.py類似，主要變動的地方為第40行 getstock()中的部分。第54行的yf.pdr\_override()，是由於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。接著第54行以更改過後的get\_data\_yahoo()進行股價抓取。

重要提醒：若讀者想將E\_14\_2範例程式透過改以yahoo股價資訊抓取的話，則可將原本在E\_14\_2.py中引入的class\_GetGoogleFinance套件改成引入class\_GetYahooFinance套件，並將main中呼叫GetGoogleFinance函數改成GetYahooFinance函數即可完成！後續章節依此例更改。

|  |  |  |  |
| --- | --- | --- | --- |
| 15-23頁 | 第5 列依靠connection 方法找到所有Models 的名稱。 | 第5 列依靠connection 方法找到所有Tables 的名稱。 |  |
| 15-29頁 | 第2列引入numpy套件... | 第3列引入numpy套件... |  |
| 15-32頁 | 更新了E\_15\_2的範例程式，36至50頁 | 此列文字刪除 |  |
| 15-37頁 | 第6 列依靠connection 方法找到所有Models 的名稱。 | 第6 列依靠connection 方法找到所有Tables 的名稱。 |  |
| 15-41頁 | 在第74 列中也計算BBand 的上中下線，最後(df['volume'].iloc[i] > (df['volMA'].iloc[i] \* 2) 為此股的volume 要大於兩倍的volMA 值，而volMA 為SMA 的技術分析。 | 在第73 列中也計算BBand 的上中下線，最後(df['volume'].iloc[i] > (df['volMA'].iloc[i] \* 2) 為此股的volume 要大於兩倍的volMA 值，而volMA 為SMA 的技術分析。 |  |
| 16-20頁 | 第6 列依靠connection 方法找到所有Models 的名稱。 | 第6 列依靠connection 方法找到所有Tables 的名稱。 |  |
| 16-26頁 | 第21列至第32列... | 第21列至第31列... |  |
| 16-27頁 | 第43 列判斷為Levy 的下降趨勢。 | 第45 列判斷為Levy 的下降趨勢。 |  |
| 16-31頁 | 程式74行  if(trade == False): | if(trade == False) and ("TWS" in trade\_type or trade\_type == "all"): |  |
| 16-32頁 | 程式95行  if(trade == False): | if(trade == False) and ("TBC" in trade\_type or trade\_type == "all"): |  |
| 16-34頁 | 第74 列判斷當如果尚未符合其他的交易策略 | 第74 列判斷當如果使用者選擇TWS策略或不選擇特定策略 |  |
| 16-35頁 | 程式8行  if(trade == False): | if(trade == False) and ("TWS" in trade\_type or trade\_type == "all"): |  |
| 程式22行  if(trade == False): | if(trade == False) and ("TIU" in trade\_type or trade\_type == "all"): |  |
| 16-36頁 | 程式35行  if(trade == False): | if(trade == False) and ("TOU" in trade\_type or trade\_type == "all"): |  |
| 程式48行  if(trade == False): | if(trade == False) and ("MS" in trade\_type or trade\_type == "all"): |  |
| 16-37頁 | 程式68行  if(trade == False): | if(trade == False) and ("TBC" in trade\_type or trade\_type == "all"): |  |
| 程式81行  if(trade == False): | if(trade == False) and ("TID" in trade\_type or trade\_type == "all"): |  |
| 16-38頁 | 程式94行  if(trade == False): | if(trade == False) and ("TOD" in trade\_type or trade\_type == "all"): |  |
| 程式107行  if(trade == False): | if(trade == False) and ("ES" in trade\_type or trade\_type == "all"): |  |

程式之排版錯誤建議讀者直接去看Github之程式較為清楚

|  |  |  |
| --- | --- | --- |
| 排版錯誤 | | |
| 勘誤位置 | 發現問題 | 備註 |
| 1-6頁 | 表格的Javascript字體不一 |  |
| v頁 | 1,2,6,7,8,9,11行中間有不適當空格 |  |
| xii頁 | 3-8 JSON…J的字體不一 |  |
| vii頁 | 第七行’’FREE”左右符號不同 | (雙引號與雙逗點) |

|  |  |
| --- | --- |
| 14-6頁 | 程式27~29、33、38~47行  排版如下圖34~37、41、40~55行 |
|  | |
| 14-7頁 | 程式59、61、63、64行  排版如下圖67、69、71、72行 |
|  | |
| 14-9頁 | 程式27、29行  排版如下圖34~37行 |
|  | |
| 14-10頁 | 程式33、38~44行  排版如下圖41、45~52行 |
|  | |
| 14-13頁 | 程式45~58行  排版如下圖53~67行 |
|  | |
| 14-14頁 | 程式60、63~64行  排版如下圖65~76行 |
|  | |
| 14-18頁 | 程式排版28、48行  排版如下圖88、108行 |
|  | |
| 14-20頁 | 程式48、51、53行  排版如下圖124、127、129行 |
|  | |
| 15-23頁 | 程式7~10行  排版如下圖10~13行 |
|  | |
| 15-25頁 | 程式40、48行  排版如下圖39、47行 |
|  | |
| 15-28頁 | 程式136~148行  排版如下圖133~146行 |
|  | |
| 15-36頁 | 程式16~30行排版需縮排  排版如下圖19~34行 |
|  | |
| 15-36頁 | 程式32~38行排版需縮排  排版如下圖35~42行 |
|  | |
| 15-38頁 | 程式73行  排版如下圖73行 |
|  | |
| 15-38、15-39頁 | 程式85~105行  排版如下圖85~105行 |
|  | |
| 15-40頁 | 程式85~105行  排版如下圖85~105行 |
|  | |
| 16-20頁 | 18~21行  排版如下圖14~18行 |
|  | |
| 16-20頁 | 程式23~33行  排版如下圖28~38行 |
|  | |
| 16-22頁 | 程式34~38行  排版如下圖88~97行 |
|  | |
| 16-31頁 | 程式80行對齊如下圖233行 |
|  | |
| 16-32頁 | 程式101行對齊如下圖283行 |
|  | |
| 16-35頁 | 程式14、24行  排版如下圖223、233行 |
|  | |
| 16-36頁 | 程式26,37,39,50,52行應凸排  排版如下圖235、246、248、259、261行 |
|  | |
| 16-37頁 | 程式74、83行應凸排  排版如下圖283、292行 |
|  | |
| 16-38頁 | 程式85,96,98,109,111行應凸排  排版如下圖235、246、248、259、261行 |
|  | |