書籍勘誤資訊表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 書號 | MP31814 | 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 |
| 1-4頁 | 1-1-2　CCS3 | | | 1-1-2　CSS3 | |  |
| 3-22頁 | 新增程式碼  str1 = "500";  document.write(" 字串轉成數值 = " + Number(str1) + "</br>"); | | | 如下圖12、13行 | | 示範Number()函數 |
|  | | | | | |
| 4-9頁 | 合計的應繳稅額為 3,645,000 | | | | 合計的應繳稅額為 3,604,900 |  |
| 5-14頁 | 有關二部曲「Python網頁程式交易APP實作：Web + MySQL + Django（第二版）」第五章14頁, E\_5\_[4.ht](http://4.ht/)ml的程式碼, 小部份更新幫忙讀者理解, 不影響執行結果。  並補充說明如下:  JavaScript 除物件資料型別是call by reference以外，其餘都是call by value (整數, 浮點數, 布林, 字串). 此程式第17列是用JSON格式(物件) var z = {sex: true};  與第7列用字串陣列(物件)b.push(" Funny!"); 兩者都是物件資料型別，因此參數傳遞採用call by reference. 當函數內對此參數做內容值更改時，因為屬同一塊記憶體，函數外的原始值也會被更改。因此，執行結果字串陣列因用.push方法新增了Funny內容值，而sex的原來內容值也由true改為false.   程式碼: E\_5\_[4.ht](http://4.ht/)ml  <!DOCTYPE html>  <html>  <head>  <script>  function f(a,b,c) {      a = 3;      b.push(" Funny!");      [c.se](http://c.se/" \t "_blank)x = false;  }  </script>  </head>  <body>  <h3>參數傳遞方式</h3>  <script>  var x = 4;  var y = ["I like ", "JavaScript"];  var z = {sex: true};  f(x,y,z);  document.write(x + "</br>" + y + "</br>" + [z.se](http://z.se/" \t "_blank)x);  </script>  </body>  </html>   執行結果  參數傳遞方式  4  I like ,JavaScript, Funny!  false | | | | |  |
| 8-18頁 | 在自己創的 app 裡，建立一個 Templates 資料夾( 只能在這個路徑、取這個名字) | | | | 在自己創的 myapp 裡，建立一個 Templates 資料夾( 只能在這個路徑、取這個名字) |  |
| 9-21頁開始 | 程式更動   |  |  | | --- | --- | | 14 | var n = 5; var m =10 ;var p=20; | | 26  (新增) | //計算20日動平均  NumAry = MovingAverage(NumAry, p, NumAry.length); | | 96 | if ((NumAry[5][i] < NumAry[4][i]) && (NumAry[6][i] < NumAry[4][i]) && (NumAry[7][i] < NumAry[4][i]) && flag == false) { | | 155  (新增) | var trace4 = {  type: 'scatter',  x: NumAry[0],  y: NumAry[7],  mode: 'lines',  name: '20日均線(MA20)',  line: {  color: 'orange',  width: 1  }  }; | | 185 | var data = [trace1, trace2, trace3, trace4, trace5, trace6]; | | | | | | 完整程式碼詳細請至Github查看 |
| 9-25頁 |  | | | | | 結果圖修正 |
| 13-2頁 | 設定文字太小，可參閱此圖 | | | |  |  |
|  | | | | | | |
| 15-27頁 | 117、118行  if loss == 0:  loss = win | | | | if loss == 0:  loss = 0.0001  if win == 0:  win = 0.0001 | 防止沒有資料0除以0錯誤，給他一個很小的數字 |
| 16-26頁 | 程式第40行  total\_change += df['close'].iloc[j]/df['open'].iloc[j] | | | | total\_change += df['close'].iloc[j] - df['open'].iloc[j] | 運算符號邏輯錯誤修正 |

錯誤處理

|  |
| --- |
| 第13章  匯入company.txt時，中文欄位出現亂碼的原因 |
| 匯入company.txt到資料表時，中文欄位出現亂碼，解決辦法建議: Mysql5.7在建立SCHEMAS(databased)的時候，需要將charset設定為utf8 或utf8mb4，若採用內定值，將會發生上述問題。若已出現此問題，則自行更改該SCHEMAS的charset即可。 |
| 第13章  若您在匯入company.txt的時候，遇到錯誤碼1148的問題時 |
| 若您在匯入company.txt的時候，遇到錯誤碼1148的問題時，請遵循下列步驟處理:   1. 進入command window，切換路徑到MySQL的指令路徑下   (參考路徑C:\Program Files\MySQL\MySQL Server 8.0\bin)     1. 以下列指令登入MySQL   mysql -u (您的使用者名稱) -p --local-infile (您的資料庫名稱)  (參考指令 mysql -u root -p --local-infile mydatabase)    輸入您的密碼後登入MySQL   1. 輸入SET GLOBAL local\_infile = 1;   將匯入本地資料的權限打開     1. 輸入匯入檔案的SQL程式碼，可以分行輸入，最後用分號結尾表示語法的結束(檔案路徑要配合您的路徑，且僅能用”/”，不能用”\”)   (參考語法)  LOAD DATA local INFILE 'C:/Users/user/Desktop/company.txt'  INTO TABLE company  CHARACTER SET utf8  FIELDS TERMINATED BY '\t'  LINES TERMINATED BY '\n'  IGNORE 1 LINES;     1. 回到workbench的query介面輸入   SELECT \* FROM mydatabase.company;  查詢，有結果即完成！ |