

國立臺灣科技大學資訊工程系

一百零六學年度第一學期專題研究

總報告

--

研究組員

胡安廷

聶誠漢

翁廷毅

許毓哲

指導教授：_____

中 華 民 國 107 年 1 月 5 日

目錄

第一章	緒論	4
1.1	研究動機	4
1.2	研究目的	4
1.3	研究簡介	4
1.4	報告組織	5
第二章	相關研究	6
2.1	Python	6
2.2	PHP	6
2.3	HTML	7
2.4	SQLite	8
2.5	網路爬蟲	8
2.6	Selenium	9
2.7	PhantomJS	9
2.8	BeautifulSoup	9
2.9	sqlite3	9
2.10	os	10
2.11	re	10
2.12	string	10
第三章	系統操作架構	11
3.1	執行流程	11
3.2	程式架構	12
3.2.1	後端搜尋及查找資料庫(演算法)	12
3.2.2	寄信與資料庫比對	13
3.2.3	網路爬蟲演算法	15
第四章	結果與討論	17
4.1.1	註冊頁面	17
4.1.2	登入頁面	17
4.1.3	使用者訂閱商品	18
4.1.4	歷史資料	18
4.1.5	現有需求	19
4.1.6	修改後的訂閱資料	19
4.1.7	已修改後的現有需求	20
4.1.8	使用者資料	20
4.1.9	寄信	21
4.1.10	資料庫(使用者需求)	21
4.1.11	資料庫(使用者資料)	22
4.2	結論	22

參考文獻.....	23
附錄.....	28
附錄 A.....	28
附錄 B.....	36
附錄 C.....	43
附錄 D.....	50
附錄 E.....	57
E.1 登入.....	57
E.2 註冊.....	60
E.3 搜尋.....	63
E.4 會員資料.....	67
E.5 修改會員資料.....	69
E.6 現有需求顯示.....	72
E.7 現有需求修改.....	75
E.8 顯示歷史資料.....	79

第一章 緒論

1.1 研究動機

線上購物已成為現代人購物的主要方式之一，電子商務已然改變我們的生活方式，精打細算的客戶穿梭在不同的平台上比較價格，卻不一定可以找到符合其需求的商品。潛在客戶可能會因為沒有找到商品而流失，但也許隔幾天就有符合需求的商品出現，而且客戶的需求可能不全是突發性的，而是有一段需求的區間，例如購買生日禮物、機票等。

現有的購物網站，使用者需要每天固定去搜尋，他才能確認是否有他所需要的商品，而我們提供他們以需求的方式搜尋商品，使用者僅需要設定下時間、商品、價格等相關資訊，我們即可為他提供定期搜尋，並且於需求符合時通知他。研究搜尋的執行時機、需求資訊與產品之間的比對方式、演算法的改進等，以期提供使用者更佳的購物經驗與方式，也是本計畫著眼的方向。

1.2 研究目的

開發出一個系統可以自動搜尋使用者所需的物品商品，現在人們普遍繁忙有時候會忘記自己上一秒前的想法或是沒有時間去慢慢搜尋自己要的東西。藉此這個系統來當個小助理幫忙這樣的人們。只要輸入所需的物品、價格、規格等等。系統會自動搜尋直到一有物件便會立刻通知使用者，使使用者能夠在最短最少的時間內也最精準的搜尋到他所要的物件。

時間：客戶為此商品設定一段期望購買到它的時間

價格上限：客戶的預算容許多少價格的商品

價格下限：客戶預期產品最低價，而不致使買到仿冒品等不符需求的商品

商品：客戶想購買的商品。例：手機 iPhone8、筆電 surface Pro4 256G i5

1.3 研究簡介

結合使用者操作面、資料庫系統、背後的爬蟲程式來完成我們的專題。後端的爬蟲程式會往各大拍賣網站或是購物網站爬下所有商品資料進我們的大資料庫裡，而使用者在登入網站後即可對網站下訂單也就是搜尋功能。而使用者的訂單會儲存在資料庫系統裡，而資料庫就會開始比對開始檢索是否有對應資料(商品)。如果符合使用者需求，便會寄信給使用者通知已搜尋到。

使用者操作面：也就是前端網頁製作，需要把使用者的資料與資料庫結合，並且設

計出使用者方便的介面。

資料爬蟲：也就是後端的資料來源，需要寫出爬蟲程式使其可以在各大拍賣網站或購物網站抓取資料，並且分類出名稱、樣式、價格、網址等等相關資料儲存進資料庫裡。

資料庫系統：結合了前端與後端的中樞橋梁，包括銜接程式運作與健全的資料庫。需要利用前端的使用者需求搜尋後端的爬蟲資料庫，進而把結果告知使用者，並且使用者能隨時修改需求時間及商品。資料庫要能隨時面對使用者不同的需求產生不同的結果。

1.4 報告組織

報告將針對相關研究、系統操作架構、結果與討論進行說明。第一章緒論，說明研究的動機目的與簡介此研究。第二章相關研究，簡述我們所採用的程式語言、框架。第三章系統操作架構部分，說明使用者之操作方式與程式內部執行之順序。第四章結果與討論，對於計畫最終所完成的結果進行說明與討論。並且於附錄附上程式碼。

第二章 相關研究

2.1 Python

是一種廣泛使用的高階程式語言，屬於通用型程式語言，由 Guido van Rossum 創造，第一版釋出於 1991 年。作為一種直譯式語言，Python 的設計哲學強調代碼的可讀性和簡潔的語法（尤其是使用空格縮排劃分代碼塊，而非使用大括弧或者關鍵詞）。相比 C++ 或 Java，Python 讓開發者能夠用更少的代碼表達想法。不管是小型還是大型程式，該語言都試圖讓程式的結構清晰明了。

與 Scheme、Ruby、Perl、Tcl 等動態型別程式語言一樣，Python 擁有動態型別系統和垃圾回收功能，能夠自動管理記憶體使用，並且支援多種編程範式，包括物件導向、命令式、函數式和程序式編程。其本身擁有一個巨大而廣泛的標準庫。

Python 虛擬機器本身幾乎可以在所有的作業系統中執行。Python 的官方直譯器 CPython，用 C 語言編寫，是一個由社群驅動的自由軟體，目前由 Python 軟體基金會管理。

2.2 PHP

PHP 用來製作動態網頁。當開啟網頁時，伺服器端便會處理 PHP 指令，然後把其結果送到網客的網頁瀏覽器上面。PHP 是開放原始碼（Open Source）而且是跨平台的。PHP 可以在 Windows NT 以及很多不同的 Unix 版本執行，它也可以被建成一個 Apache 模組，或者是一個 CGI 的二元檔案（binary）。

PHP 可以用來傳送 HTTP 表頭（header）。可以設定 cookies，做授權管理，以及將使用者重新導向至新的頁面（redirect users）。也提供了優良的資料庫連結功能（還有 ODBC 的連結功能），另外還有提供與各式各樣的外部檔案庫（library），可以用來做幾乎所有的事情，上至產生 PDF 文件，下至解析 XML。

PHP 程式碼就在網頁中，因此不必為它建立一個開發環境或整合性開發環境（IDE）。用 `<?php` 開始一段 PHP 程式，然後以 `?>` 做結束。（也可以將 PHP 設定成使用 ASP 形式的 `<% %>` tag 甚至是用 `<SCRIPT LANGUAGE="php"></SCRIPT>`。）PHP 引擎會處理 tag 之間的任何東西。

PHP 語言的語法跟 C 還有 Perl 很像。在使用變數前，不必先宣告他們。要建立陣列（array）以及雜湊（hash）或關聯性陣列也很快。PHP 甚至還有一些基本的物件導向功能，幫助組織以及封裝（encapsulate）程式。

PHP 語言的語法跟 C 還有 Perl 很像。在使用變數前，不必先宣告他們。要建立陣列（array）以及雜湊（hash）或關聯性陣列也很快。PHP 甚至還有一些基本的物件導向功能，幫助組織以及封裝（encapsulate）程式。

2.3 HTML

HTML 是一種標記語言；會告訴瀏覽器該如何呈現你的網頁——單純簡易或是極其複雜的頁面都沒問題。HTML 包含了一系列的元素（element），而元素包含了標籤（tag）與內容（content），我們用標籤來控制內容的呈現樣貌，例如字體大小、斜體粗體、在文字或圖片設置超連結等。

1. 起始標籤（The opening tag）：先打角括弧，也就是大於、小於的符號「<>」，裡面再放入元素名稱，如上面的例子「<p>」。起始標籤代表這個元素從這裡開始。
2. 結束標籤（The closing tag）：與起始標籤一樣，只是在元素名稱前面多了個斜線「/」。很容易理解地，內容的最後加上結束標籤，代表這個元素的尾端。
3. 內容（The content）：元素的內容。
4. 元素（The element）：由起始標籤、結束標籤、內容所組成。

元素還可以有「屬性（Attribute）」

屬性的組成包含：

1. 在元素名稱和屬性之間有一個空格（如果有多個屬性，屬性之間也需要有空格）
 2. 屬性名稱後面接著等於符號「=」
 3. 屬性包在起始標籤裡面
- <!DOCTYPE html> — 文件類型（doctype）。在 HTML 發展初期，約莫 1991/2 的年代，文件類型是用來連結一些應遵守的規則，有點像自動校正的功能。
 - <html></html> — <html> 元素，又被視為根元素（root element），包含了所有顯示在這個頁面上的內容。
 - <head></head> — <head> 元素，裡面放的是想涵括的重要資訊，但不會顯示於網頁瀏覽者眼前的。例如，顯示於搜尋結果的關鍵字、頁面說明、CSS、字元實體集...等。
 - <body></body> — <body> 元素，包含了所有會顯示於網頁瀏覽者眼前的內容。無論是文字、圖片、影片、互動遊戲...等。
 - <meta charset="utf-8"> — 這個元素指定了你的文件使用 utf-8 這種字元編碼，它會幫助你免去許多文字無法正確呈現的煩惱。
 - <title></title> — 呈現於網頁瀏覽者眼前的網頁標題。

2.4 SQLite

SQLite 與其他一般資料庫差異不大，一般的 SQL-92 語法都能夠使用，而且不需要建立一個資料庫系統，要使用的時候，只要在編譯程式的時候將 SQLite 程式庫一起編入就可以使用。另外，SQLite 的資料庫(database)都是以單一檔案的形式存於磁碟中，不需要再安裝資料庫伺服器軟體，所以要把資料庫複製或建立在你的電腦上是相單簡單快速。

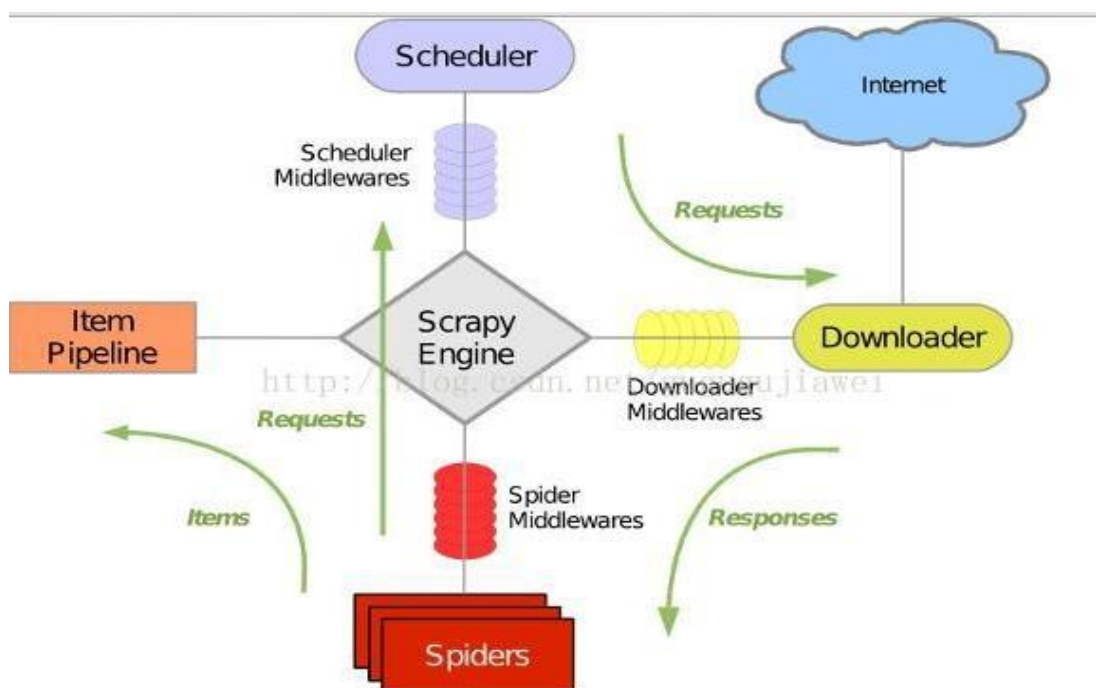
SQLite 是一個很小的 C 語言程式庫，且本身就完全包含資料庫引擎的功能，而且可以嵌入至其他程式中，完全不用額外的設定。

2.5 網路爬蟲

也叫網路蜘蛛 (spider)，是一種用來自動瀏覽全球資訊網的網路機器人。其目的一般為編纂網路索引。本專題採用基於 python 語言的 scrapy。

網路爬蟲始於一張被稱作種子的統一資源位址 (URL) 列表。當網路爬蟲存取這些統一資源定位器時，它們會甄別出頁面上所有的超連結，並將它們寫入一張「待訪列表」，所謂爬行疆域 (crawl frontier)。此疆域上的諸 URL 將被按照一套策略迴圈存取。

網際網路資源多如繁星，這意味著網路爬蟲在一定時間內只能下載有限數量的網頁，因此它需要最佳化它的下載方式。網際網路資源瞬息萬變，這也意味著網路爬蟲下載的網頁在使用前就已經被修改甚至是刪除了。這是網路爬蟲設計者們所面臨的兩個基本問題。



使用的 Library

2.6 Selenium

是自動化測試工具。它支持各種瀏覽器，包括 Chrome，Safari，Firefox 等主流界面式瀏覽器，如果你在這些瀏覽器裡面安裝一個 Selenium 的插件，那麼可以方便地完成對 Web 界面的測試。換句話說叫 Selenium 支援這些瀏覽器驅動。Selenium 支持多種語言開發，比如 Java，C，Ruby、Python。安裝時在 cmd 中輸入 `pip install selenium` 即可。

2.7 PhantomJS

PhantomJS 是一個基於 webkit 的 JavaScript API。它使用 QtWebKit 作為它核心瀏覽器的功能，使用 webkit 來編譯解釋執行 JavaScript 代碼。任何你可以在基於 webkit 瀏覽器做的事情，它都能做到。它不僅是個隱形的瀏覽器，提供了諸如 CSS 選擇器、支持 Web 標準、DOM 操作、JSON、HTML5、Canvas、SVG 等，同時也提供了處理文件 I/O 的操作，從而使你可以向操作系統讀寫文件等。PhantomJS 的用處可謂非常廣泛，諸如前端無界面自動化測試（需要結合 Jasmin）、網絡監測、網頁截屏等。

2.8 BeautifulSoup

Beautiful Soup 是一個 Python 包，功能包括解析 HTML、XML 文檔、修復含有未閉合標籤等錯誤的文檔（此種文檔常被稱為 tag soup）。這個擴展包為待解析的頁面建立一棵樹，以便提取其中的數據，這在網絡數據採集時非常有用。

2.9 sqlite3

python 對 sqlite 資料庫操作的 library，主要使用包含 connect、cursor、fetch、commit 等函式。

connect()：即是資料庫的連接，可以利用它來切換連接到不同的資料庫。

cursor()：類似操作介面，是用來操作 connect 後的回傳值經 cursor 呼叫 execute()即可使用 sqlite 的關鍵字。

fetch()：就是把 execute()後的資料抓取出來，存成 python 列表的形式，有 fetchone()、fetchall()兩種。

commit()：即許可，許可前面所有尚未被許可的操作但僅限於 cursor 所對應的資料庫。

2.10 os

os 提供了 python 對於系統資料的處理，像是檔案檢查與刪除等。

getcwd()：抓取現在執行的檔案所在目錄。

os.path：對於提供的 path 執行後面宣告的操作，像是刪除、確認等。

isfile()：檔案存在與否。

remove()：移除指定路徑檔案。

2.11 re

re 即是正規化，對於相似邏輯的字串操作很有幫助。

split()：字串切割，最多提供 4 個變數 input，第一個為切割符，二為要切割的字串，三為最多切割次數。切割符的 keyword 參考以下連結或查詢 python 提供之文件。

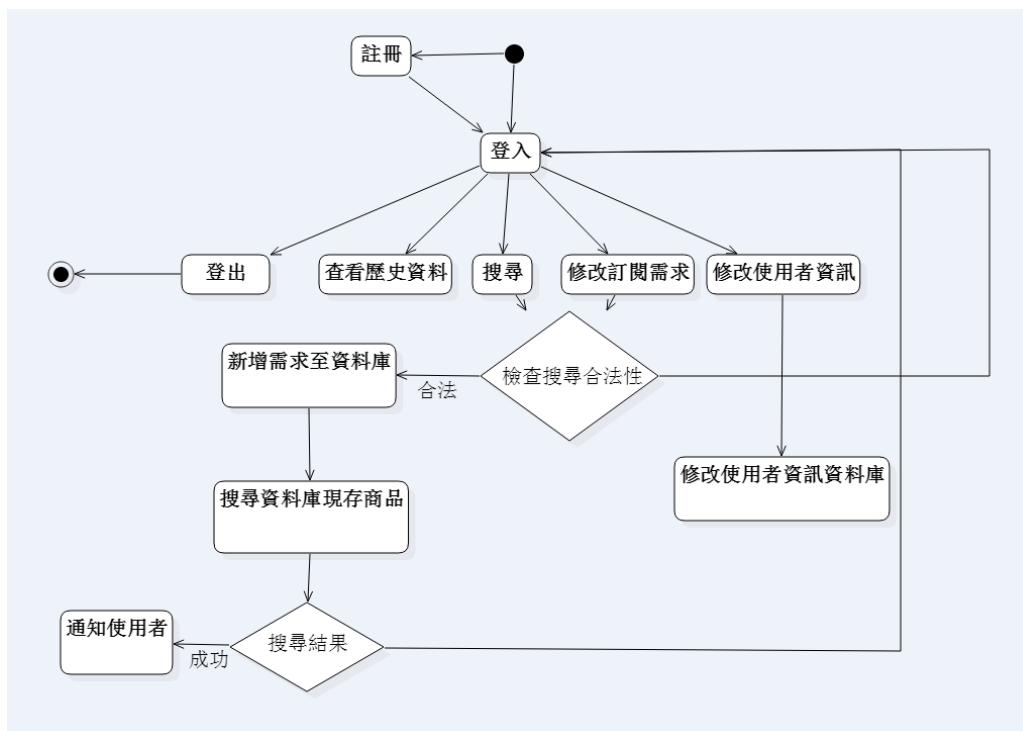
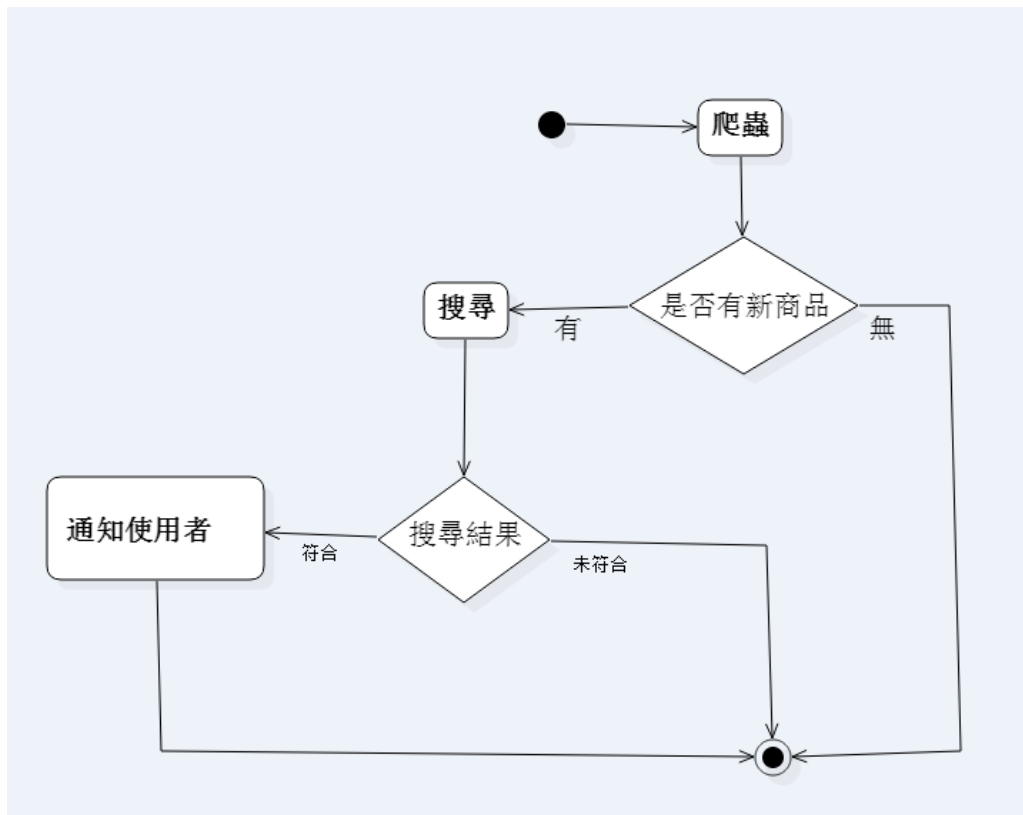
<http://blog.fantasy.codes/python/2013/07/02/py-re-module/>

2.11 string

string 提供了許多字串可用的擴充功能，與 re 的使用相輔相成，像是 punctuation 包含所有符號、digit 包含數字等、replace 替換字元，可以用來檢查字串的合法性。

第三章 系統操作架構

3.1 執行流程



3.2 程式架構

3.2.1 後端搜尋及查找資料庫(演算法)

a. 搜尋

1. 字串分析

目前運作的產品僅手機、電腦、衣服等，以 Iphone7plus 這組字串為例，使用者並未切割字串，我們可以將它切成

case1:Iphone、7plus

case2:Iphone、7、plus

case3:I、phone、7、plus

而且這類以數字為切割符的分割幾乎可以用在大多數的有流水編碼、大小寫區分的商品需求裡，像 Microsoft Surface Pro 4 等。

2. 邏輯組合

經由前面的分析結合邏輯組合成搜尋條件，再以 Iphone7plus 為例，若直接符合原字串則必然成立，用 and 相連，數字、大小寫分割出的子字串間用 and 相連，例：str has 7 and str has plus，而這組單元與其他的單元以”or”連接，則可適用於進階搜尋功能，幫助使用者增加更多細部需求如：iphone7plus128g 等...

b. 系統邏輯

1. 後端協作

main() 中執行順序為：

1. 呼叫爬蟲 process.start()
2. 檢查是否有新的商品 new_Product_In()
3. 確認是否有新商品符合已存在的需求 Request_fetch_and_check()
4. 符合需求者寄信 sending()

由於商品價格並非持續變動，而是有幾天或小時的間隔，因此此部分採取定期呼叫，利用工作排程執行。

附件一為後端程式碼。

2. 前端協作

main() 中執行順序為：

1. 測資處理
2. 新需求確認 new_Request_In()
3. 若有新需求，則處理 new_Request_search()
4. 修改需求確認 modify_Request_In()
5. 需求處理 new_Request_search()

由於使用者需求為不定期的，使用迴圈待命。

附件二為前端程式碼。

3.2.2 寄信與資料庫比對

1. 資料刪除(測試版)

```
3 def data_delete():
4     cursor.execute("DELETE FROM 'abc' WHERE cost=1000")
5     conn.commit()
```

利用使用者輸入欲取消名稱來配對資料庫裡刪除動作

2. 資料更新(測試版)

```
def data_update():
    cursor.execute("Update 'abc' SET cost=1500 WHERE title ='eee'")
    conn.commit()
```

使資料庫裡的東西更動更新，而不是建立新的資料

3. 到期時間

```
1 def time_check1():
2     print (time.strftime("%Y %m %d", time.localtime()) )
3     yearnow = time.strftime("%Y", time.localtime())
4     monthnow = time.strftime("%m", time.localtime())
5     daynow = time.strftime("%d", time.localtime())
6     cursor.execute("DELETE FROM 'Exist_Request_Table' WHERE year< ?", (yearnow, ))
7     cursor.execute("DELETE FROM 'Exist_Request_Table' WHERE year= ? AND mon< ?", (yearnow,monthnow,))
8     cursor.execute("DELETE FROM 'Exist_Request_Table' WHERE year= ? AND mon= ? AND date<=", (yearnow,monthnow,daynow))
9     conn.commit()
10    cursor.close()
11    conn.close()
```

Time.localtime()會抓出目前電腦時間

再去以資料庫語法去一一比對目前這個時間點

有哪些搜尋是超過時間的

如果超過時間就直接刪除，參照 1

4. 寄信通知

```
def sending(product, arr, user):
    print("sendStart")
    t=(user,)
    print(user)
    cursorUser.execute("SELECT * FROM 'UserData' WHERE username = ?",t)
    row = cursorUser.fetchone()

    gmail_user = 'SenderOfProject@gmail.com'
    gmail_password = '94879487'

    from_add = gmail_user
    to = row[2]##'這裡是寄到哪裡@gmail.com'
    # to = gmail_user
    subject = 'We found some ' + product + ' meet your request'

    things = "Hello " + user + ",\n\tLinks below are url of product that meet your request.\n\n"
    count = 1
    for url in arr:
        things = things + str(count) + ". " + url + "\n"
        count += 1

    msg=MIMEText(things)
    msg['Subject']=subject
    msg['From']= gmail_user
    msg['To']=to

    try:
        smtpObj = smtplib.SMTP('smtp.gmail.com', 587)
        smtpObj.ehlo()
        smtpObj.starttls()
        smtpObj.login(gmail_user, gmail_password)
        smtpObj.sendmail(from_add, to, msg.as_string())
        smtpObj.close()
        print ('Email sent')
    except:
        print ('Something went wrong')
```

此功能為以信件方式通知使用者愈搜尋物品已經出現
提醒使用者去看看物品等等
利用 GMAIL 的方式寄信，登入我們創的系統信箱
並且調理列出連結與名稱
方便使用者使用信箱系統

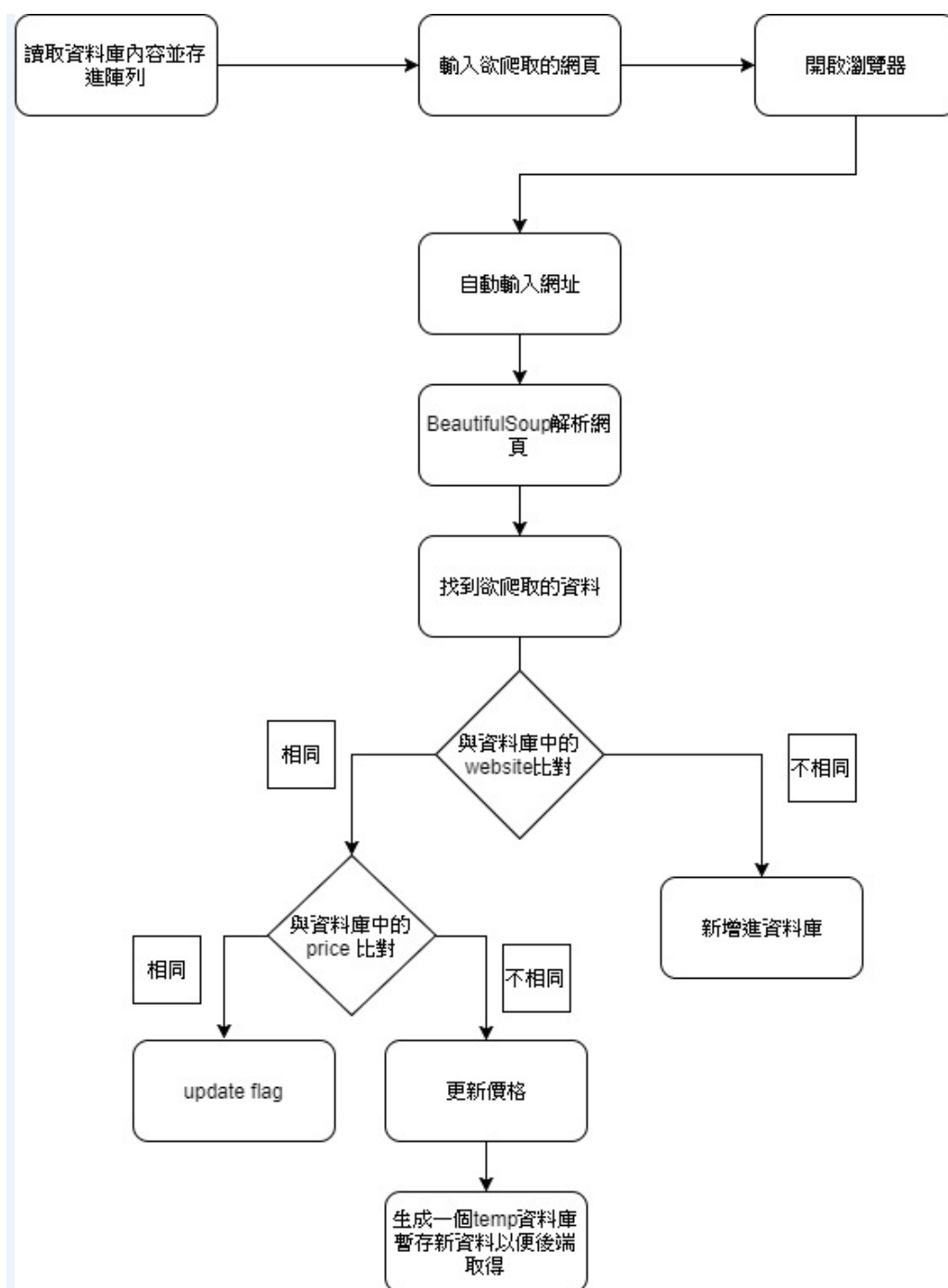
3.2.3 網路爬蟲演算法

A. 搜尋及比對資料庫

1. 在程式執行初期執行 `__init__` 這個函式，它會生成一個資料庫並預設更新 flag 為 False
2. 接著會執行 `start_requests()`，他會連結資料庫並把所有屬性的值丟進對應的陣列，例如：資料庫中的 title 屬性會被丟進程式中的陣列(程式中變數名稱為 `db_title`)，以此陣列去比對將要爬進資料庫的新資料。
3. 當爬蟲找到資料時，會去和 `db_website` (資料庫中的商品網址)做比對。假如爬到的資料和 `db_website` 相同，再去比對價格(`db_price`)有無相符。
4. 假如爬到的資料的 `price` 和 `db_price` 相同，代表商品沒有被更新價格。
5. 假如爬到的資料的 `price` 和 `db_price` 不相同，代表商品被更新價格，需要去更新資料庫中的價格
6. 假如爬到的資料和 `db_website` 沒有相同的，新增進資料庫。流程請參照附圖。

B. 爬蟲

1. `parse()` 會從 `start_urls` 這個陣列中取的欲爬取的網頁網址，並用 for 迴圈一個一個輸入進瀏覽器。
2. 瀏覽器取的網址後，進入到網頁，用 BeautifulSoup 此 python 套件去美化及排版取的的 html 程式碼，以方便我們去擷取需要的資訊。
3. 每個網站的頁面配置不盡相同，所以每當需要爬取一個新網站時，需要寫一個新的爬蟲去應對，本專題中爬取了露天拍賣和 pchome 兩家網路買賣平台。
4. 露天拍賣與 pchome 的爬蟲不同的地方在於，露天拍賣的瀏覽方式為換頁式，意思是網址列最後一個字為頁碼，只要將頁碼設定好一個範圍，便可以爬取那個範圍的資料。程式碼請參照附件 C。
5. pchome 為分類式的網站，例如他將手機、電腦、平板歸類為 3C 類，我們只需要從類別中依序將網址填入到 `start_urls` 中，便可以完成爬蟲。程式碼請參照附件 D。



第四章 結果與討論

4.1 前端介面

4.1.1 註冊頁面



註冊

帳號：

密碼：

再次輸入密碼：

信箱：

Pchome


yahoo拍賣


露天拍賣


4.1.2 登入頁面



登入

帳號：

密碼：

Pchome


yahoo拍賣


露天拍賣


4.1.3 使用者訂閱商品



[歷史資料](#)/[訂閱資料](#)/[會員資料](#)/[登出](#)

產品種類:

產品名稱:

價格區間: ~

結束日期:



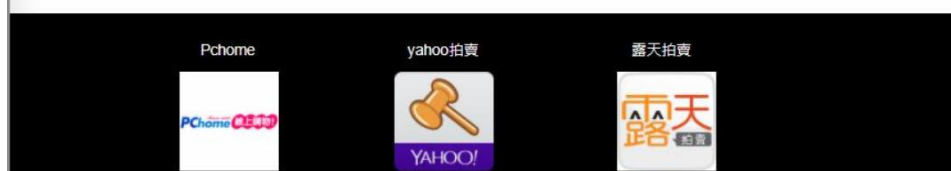
4.1.4 歷史資料



歷史資料

John用戶

	產品名稱
1	iphone7plus



4.1.5 現有需求



現有需求

John用戶

	產品名稱	價格下限	價格上限	截止日期
●	iphone7plus	1000	10000	2017-12-25

Pchome



yahoo拍賣



露天拍賣



4.1.6 修改後的訂閱資料



修改後的訂閱資料

價格區間： ~

結束日期：

Pchome



yahoo拍賣



露天拍賣



4.1.7 已修改後的現有需求



現有需求

John用戶

產品名稱	價格下限	價格上限	截止日期
●iphone7plus	1000	20000	2017-12-25

Pchome


yahoo拍賣


露天拍賣


4.1.8 使用者資料



John用戶

帳號:John

信箱:john171117@yahoo.com.tw

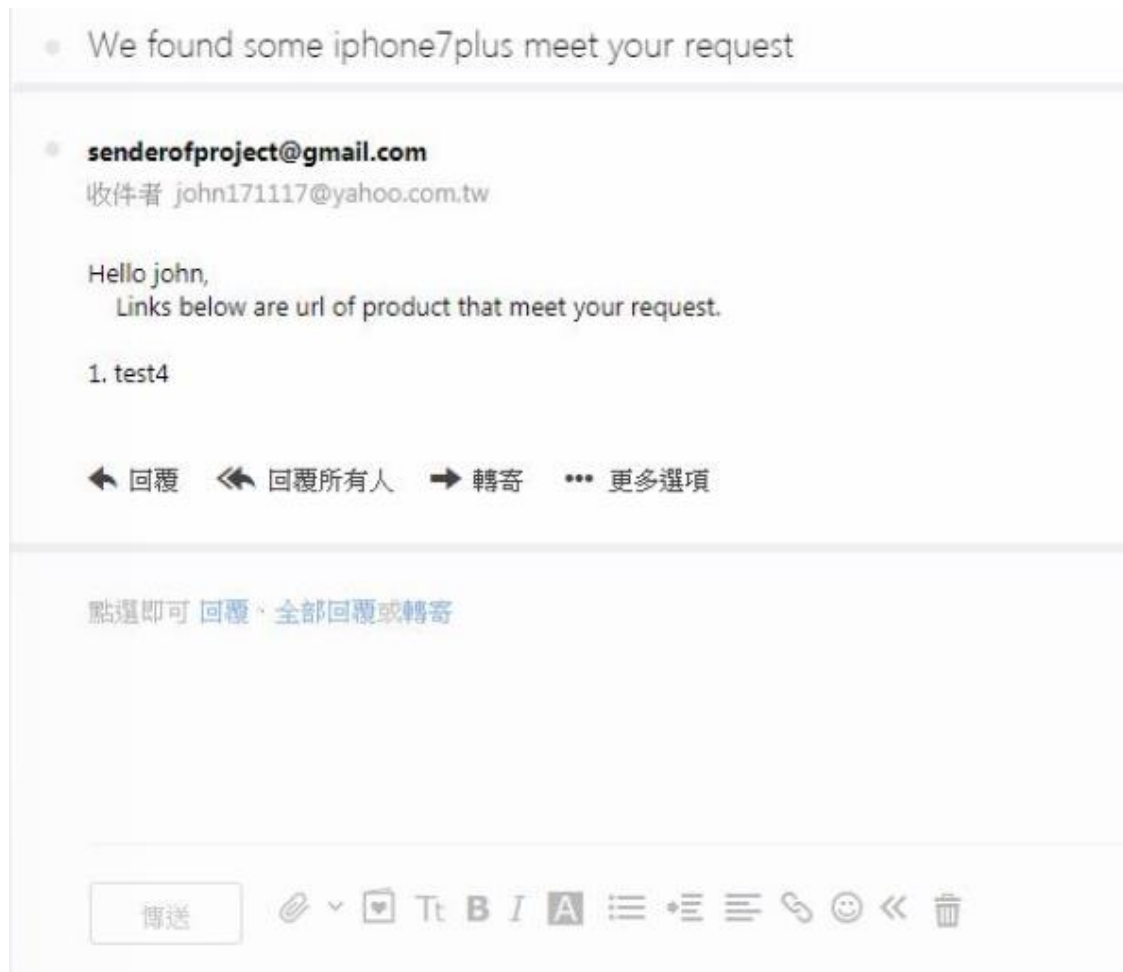
Pchome


yahoo拍賣


露天拍賣


4.1.9 寄信

搜尋到有商品符合需求，寄信給使用者



4.1.10 資料庫(使用者需求)

資料庫包含使用者姓名、訂閱商品內容、價格、到期日期。

SQLite Viewer
view sqlite file online

Drop file here to load content or click on this box to open file dialog.

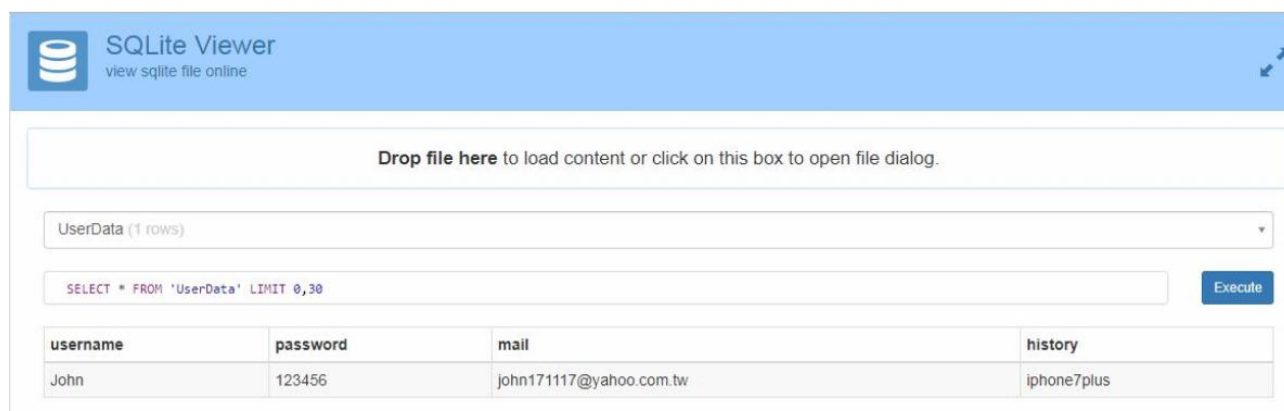
Exist_Request_Table (1 rows)

SELECT * FROM 'Exist_Request_Table' LIMIT 0,30 Execute

Type	username	product	lowerlimit	higherlimit	year	month	day	RequestID
cellphone	John	iphone7plus	1000	20000	2017	12	25	1

4.1.11 資料庫(使用者資料)

此資料庫包含使用者名稱、密碼、信箱以及訂閱歷史。



The screenshot shows the SQLite Viewer web interface. At the top, there's a blue header with the SQLite logo and the text 'SQLite Viewer view sqlite file online'. Below the header is a white box with the instruction 'Drop file here to load content or click on this box to open file dialog.'. Underneath, a dropdown menu shows 'UserData (1 rows)'. Below the dropdown is a text input field containing the SQL query 'SELECT * FROM 'UserData' LIMIT 0,30'. To the right of the input field is a blue 'Execute' button. Below the input field is a table with the following data:

username	password	mail	history
John	123456	john171117@yahoo.com.tw	iphone7plus

4.2 結論

由於現在購物環境已從現實逐漸轉移到在網路購物平台。人們除了在網路購物平台上購買商品，還會進行商品比價，以此來減輕荷包壓力。但是往往進行比價後的價格不是很滿意，並不符合當時消費者的預期價格，因此消費者會希望能有一個能時時監控他的需求商品是否能到達他的預期價格的軟體。本專題使用爬蟲並用 Windows 內建的工作排程進行監控商品的價格，並透過演算法去計算當時網頁中的價格是否有到達使用者的預期價格。由於速食文化的興起，人們已不像以往可以慢慢等待，現在就算已經在網路上訂好商品，也等不及商品配送到達的時間。也因此人們不會時時刻刻注意自己的需求商品是否已到達他的預期價格，本系統會自動搜尋直到需求商品到達使用者的預期並立即通知使用者，讓使用者節省很多時間不用每天反覆去搜尋比對價格，不論是時間上或是金錢方面的幫助都是一大助益。

參考文獻

- SQLite 命令：<http://www.runoob.com/sqlite/sqlite-alter-command.html>
- http://tw.gitbook.net/sqlite/sqlite_syntax.html
- 檢視 SQLite：<http://sqliteviewer.flowsoft7.com/>
- Python 概念語法：<http://djangogirlstaipei.herokuapp.com/tutorials/python/?os=windows>
http://itman2266.blogspot.tw/2013/04/python_22.html
- Python 日期時間：<http://www.runoob.com/python/python-date-time.html>
- Python 寄信：http://self.jxtsai.info/2016/09/python_22.html
- <http://sodahau.logdown.com/posts/141180-python-through-gmail-mail>
- <https://www.lightblue.asia/send-gmail-by-python/>
- <https://jamching.com/article/Python-to-Send-Email-1.html>
- PHP-維基百科：<https://zh.wikipedia.org/wiki/PHP>
- HTML 基礎-學習如何開發：https://developer.mozilla.org/zh-TW/docs/Learn/Getting_started_with_the_web/HTML_basics
- PHP 入門必讀：<http://web.cogsh.tp.edu.tw/department/ccr/skill/phpintro.htm>
- SQLite 簡介：<http://www.runoob.com/sqlite/sqlite-intro.html>
- SQLite 簡介：<http://mybeauty.pixnet.net/blog/post/26492636-sqlite%E7%B0%A1%E4%BB%8B>
- HTML Tutorial：<https://www.w3schools.com/html/>
- PHP 登入系統教學：<http://dreamtails.pixnet.net/blog/post/23583385-%5B%E6%95%99%E5%AD%B8%5Dphp%E6%9C%83%E5%93%A1%E7%99%BB%E5%85%A5%E6%A9%9F%E5%88%B6%EF%BC%8Csession%E7%9A%84%E4%BD%BF%E7%94%A8%EF%BC%8C%E7%B0%A1%E6%98%93%E5%9E%8B>
- PHP 教學：<http://james670818.pixnet.net/blog/post/2337519-%5Bphp%5D-%E8%A3%BD%E4%BD%9C%E7%99%BB%E5%85%A5%E9%A0%81%E9%9D%A2>
- PHP 會員系統教學：<http://johnnycomputer.blogspot.tw/2012/04/php.html>
- SQLite 使用教學：<http://gisanfu.pixnet.net/blog/post/7941810-%E8%BD%89%E8%B2%BC%EF%BC%9Asqlite%E4%BD%BF%E7%94%A8%E6%95%99%E5%AD%B8>
- 10 分鐘掌握 SQLite：<http://droidyue.com/blog/2014/09/08/learn-sqlite-in-a-very-fast-way/>

- 使用 php 操作 SQLite:
<https://gnehcic.azurewebsites.net/%E4%BD%BF%E7%94%A8php%E6%93%8D%E4%BD%9Csqlite/>
- php&SQLite: <http://hovertree.com/hvtart/bjae/pqma7k2h.htm>
- 輕型資料庫 SQLite 結合 PHP 開發:
- <http://q1696.pixnet.net/blog/post/180243169-%E8%BC%95%E5%9E%8B%E6%95%B8%E6%93%9A%E5%BA%ABsqlite%E7%B5%90%E5%90%88php%E7%9A%84%E9%96%8B%E7%99%BC>
- PHP 實現的簡單操作 SQLite 數據庫類與用法
<http://www.itread01.com/articles/1497980296.html>
- 爬蟲學習之一個簡單的網絡爬蟲 <https://kknews.cc/tech/2959p9.html>
- Scrapy 爬蟲框架：抓取淘寶天貓數據 <https://kknews.cc/tech/39orm2o.html>
- PTTcrawler <https://github.com/twtrubiks/PTTcrawler/blob/master/pttcrawler.py>
- Python 基本爬蟲程式 (crawler) 範例 - 以 Google 新聞為例
- <http://tekibrain.blogspot.tw/2015/12/python-crawler-google.html>
- python 實戰資料科學工作坊 <https://www.slideshare.net/ChiuYW/python-78119008>
- Scrapy (一):python 語言在爬網界的標準 <http://ithelp.ithome.com.tw/articles/10138348>
- Python 爬虫(六)--Scrapy 框架学习 <http://www.jianshu.com/p/078ad2067419>
- XPath 教程 <http://www.w3school.com.cn/xpath/>
- XPath 节点 http://www.w3school.com.cn/xpath/xpath_nodes.asp
- 範例 5：爬比價網的大同電鍋搜尋
[http://nickwarm.logdown.com/posts/891274-example-5-a-climbing-parity-net-tatung-rice-cooker-s-爬蟲實戰-using-Python3 \(part 1 / 3\)](http://nickwarm.logdown.com/posts/891274-example-5-a-climbing-parity-net-tatung-rice-cooker-s-爬蟲實戰-using-Python3-part1-3)
- [earches http://www.coder1024.net/crawler-using-python3-part1/#lv1](http://www.coder1024.net/crawler-using-python3-part1/#lv1)
- Scrapy Tutorial http://blog.sina.com.cn/s/blog_64d0b03c0101gvh4.html
- [Scrapy 爬蟲] 如何撰寫第一支 Scrapy 爬蟲以抓取蘋果即時新聞?
- <http://www.largitdata.com/course/68/>
- PTT 爬蟲圖片下載器 (使用 Scrapy) PttImageSpider
<https://github.com/twtrubiks/PttImageSpider>

- Python 网页信息采集：使用 PhantomJS 采集某宝某猫商品内容
- <http://www.jianshu.com/p/831bb2896f78>
- 超级易懂爬虫系列之使用 scrapy+webdriver 爬取动态网页：
- http://www.huqi.tk/index.php/2017/05/24/python_spider_scrapy_webdriver/
- [Python] 漫畫下載機器人 <http://riceeechang.blogspot.tw/2015/06/python.html>
- 為何大量網站不能抓取?爬蟲突破封禁的 6 種常見方法
- <https://kknews.cc/zh-tw/tech/mjqmg.html>
- Python 爬虫(七)--Scrapy 模拟登录 <http://www.jianshu.com/p/b7f41df6202d>
- 使用 Scrapy 爬取知乎网站
<http://blog.javachen.com/2014/06/08/using-scrapy-to-crawl-zhihu.html>
- 露天拍賣資料撈取 by python
- <http://bugcaptor.logdown.com/posts/181018-open-auction-data-backwaters-by-python>
- 20+個 Selenium 網站自動化測試程式技巧彙整 -以 Java 為範例
- <https://www.qa-knowhow.com/?p=2117>
github-login-selenium:https://github.com/stayhigh/github-login-selenium/blob/master/github_login.py
- [Python] Selenium 教學 <http://zwindr.blogspot.tw/2016/07/python-selenium.html>
- [译]Selenium Webdriver - 下载、安装稳定版本
<https://huilansame.github.io/huilansame.github.io/archivers/selenium-download-and-install-stable-versions>
- python 模拟登录，使用 cookie 自动领取下载豆(最新 2016)
- <http://blog.51cto.com/gaoming/1355293>
- [Python] Requests 教學
- <http://zwindr.blogspot.tw/2016/08/python-requests.html>
- Python + selenium 自动化测试 <https://www.zybuluo.com/king/note/194883>
- python3 的爬虫笔记 11——Selenium 和浏览器的一些设置

- <http://www.jianshu.com/p/19890d7af5dd>
- Python 的虛擬環境及多版本開發利器—Virtualenv 與 Pythonbrew
- <https://www.openfoundry.org/tw/tech-column/8516-pythons-virtual-environment-and-multi-version-programming-tools-virtualenv-and-pythonbrew>
- Scrapy 框架学习 笔记 2——Scrapy 与 Django 结合
<http://www.xefan.com/archives/83887.html>
- Scrapy 爬蟲庫快速入門 <https://www.zixundingzhi.com/search/?query=scrapy+>
- scrapy 同时运行多个爬虫 <http://www.jianshu.com/p/24842205c82b>
- 同时运行多个 scrapy 爬虫的几种方法（自定义 scrapy 项目命令）
<http://www.cnblogs.com/rwxwsblog/p/4578764.html>
- SCRAPY 同時運行多個爬蟲 <https://ddnews.me/world/fp1w103u.html>
- scrapy 多个 spider 怎么指定 pipeline
<https://segmentfault.com/q/1010000004863755>
- 通用爬虫(Broad Crawls)http://scrapy-chs.readthedocs.io/zh_CN/0.24/topics/broad-crawls.html
- scrapy-multi <https://github.com/yanceyblog/scrapy-multi>
- Scrapy 教程 10- 动态配置爬虫
http://scrapy-cookbook.readthedocs.io/zh_CN/latest/scrapy-10.html
- 为什么要学习爬虫 <https://piaosanlang.gitbooks.io/spiders/01day/section1.1.html>
- scrapy 笔记之探索数据流程 <http://snowdream1314.github.io/2016/02/19/scrapy-1/>
- Scrapy 笔记 06- Item Pipeline
<https://www.xncoding.com/2016/03/18/scrapy-06.html>
- 爬虫总结(二)-- scrapy
<http://www.shuang0420.com/2016/06/12/%E7%88%AC%E8%99%AB%E6%80%BB%E7%B%93-%E4%BA%8C-scrapy/>
- Python X 網路爬蟲一筆記(一)-如何開始(範例)

- https://dotblogs.com.tw/jason_wang/2016/12/18/python_crawler
- 什麼是網路爬蟲? <http://www.largitdata.com/course/1/>

附錄

A. 附件一

```
# -*- coding: utf-8 -*-
"""

Created on Thu Oct 26 00:34:45 2017

@author: Danny
"""

import sqlite3
import os
import re
import smtplib
from email.mime.text import MIMEText
from email.header import Header
from scrapy.utils.project import get_project_settings
from scrapy.crawler import CrawlerProcess
from sky.spiders import ComputerCrawler, CellphoneCrawler, BoyClothesCrawler

conn = sqlite3.connect('cellphone.sqlite')#連接商品資料庫，預設為cellphone.sqlite
cursor = conn.cursor()

connUser = sqlite3.connect('info.sqlite') #連接使用者資料庫
cursorUser = connUser.cursor()

def main():
    print('Crawler\n')
    process = CrawlerProcess(get_project_settings())#爬蟲操作設定

    process.crawl(CellphoneCrawler, cellphone())
    process.crawl(ComputerCrawler, computer())
    process.crawl(BoyClothesCrawler, boyclothes())
    process.start()#開始爬蟲

    print('go checking\n')

    a = new_Product_In()#檢查是否有新進產品
```

```

if a == '1':
    print('new coming check\n')
    Request_fetch_and_check()#檢查新產品是否符合需求
    a = '0'
else:
    print('nothing to check\n')

print('finish all work\n')

conn.commit()#許可操作並且關閉資料庫
cursor.close()
conn.close()

connUser.commit()
cursorUser.close()
connUser.close()

print('the End.')

#新產品存在確認
def new_Product_In():
    file_path = os.getcwd() + "/true.txt"#定義好以true.txt作為確認存在
    file_path_close = os.getcwd() + "/false.txt"#停用的協議，false.txt阻止後續程式

    if ( os.path.isfile(file_path)):
        print("new Arrival exists!")
        os.remove(file_path) #移除true.txt
        conn = sqlite3.connect('cellphone.sqlite')#連接cellphone.sqlite
        cursor = conn.cursor()
        #製作demo，把測資放入temp的資料庫裡
        cursor.execute("INSERT INTO temp VALUES(' iPhone X 8 7 Plus i7 6s i6 Plus SE 5s 5 ', 9, 'test0',
'2017/12/04/21:10:13', 1)")
        conn.commit()
        cursor.execute("INSERT INTO temp VALUES('iPhone X 8 7 Plus i7 6s i6 Plus SE 5s 5 ', 20000, 'test1',
'2017/12/04/21:10:13', 1)")
        conn.commit()
        cursor.execute("INSERT INTO temp VALUES('iPhone X 8 6s i6 Plus SE 5s 5', 9999, 'test2',
'2017/12/04/21:10:13', 1)")
        conn.commit()

```

```

        cursor.execute("INSERT INTO temp VALUES(' iPhone X 8 6s i6 Plus SE 5s 5 ', 99, 'test3',
'2017/12/04/21:10:13', 1)")
        conn.commit()
        cursor.execute("INSERT INTO temp VALUES(' iPhone X 8 7 Plus i7 6s i6 Plus SE 5s 5 ', 9999, 'test4',
'2017/12/04/21:10:13', 1)")
        conn.commit()
        return '1'

elif(os.path.isfile(file_path_close)):
    print("stop exec!")
    os.remove(file_path)
    return '-1'

else:
    print("File not found!")
    return '0'

#檢查現存需求
def Request_fetch_and_check():

    cursorUser.execute("SELECT * FROM 'Exist_Request_Table' WHERE Type = 'cellphone'")
    #確認request Table中有無'cellphone'類需求
    rowC = cursorUser.fetchone()

    #判別式 row is None時，就表示沒有此類需求
    if rowC is None:
        print('No Cellphone Request exist\n')

    else:
        print('Cellphone Request exist\n')

    conn = sqlite3.connect('cellphone.sqlite')#連接對應資料庫
    cursor = conn.cursor()

    cursor.execute("SELECT * FROM sqlite_master WHERE type = 'table' AND name = ?;",('temp',))#確認有無
    新的cellphone商品
    exists = bool(cursor.fetchone())
    conn.commit()

```

```

if exists:#有商品->search
    print('new Arrive\n')
    while rowC is not None:

        title = rowC[2]
        searchStr = CreateSearchStr(rowC[2]) #製作搜尋字串

        costU = rowC[4]
        costD = rowC[3]
        user = rowC[1]

        t=(costU,costD,)

        search_New_Product(user, title, t, searchStr) #搜尋與輸出
        rowC = cursorUser.fetchone()#下個此類需求
        cursor.execute("DROP TABLE temp")#刪除temp表
        conn.commit()

    else:
        print('no New\n')
        conn.commit()
print('End of check\n')

cursorUser.execute("SELECT * FROM 'Exist_Request_Table' WHERE Type = 'boyclothes'");
rowB = cursorUser.fetchone()

if rowB is None:
    print('No BoyClothes Request exist')

else:
    conn = sqlite3.connect('boyclothes.sqlite')
    cursor = conn.cursor()

    cursor.execute("SELECT * FROM sqlite_master WHERE type = 'table' AND name = ?;",('temp',))
    exists = bool(cursor.fetchone())
    conn.commit()

    if exists:
        while rowB is not None:

```

```

        title = rowB[2]
        searchStr = CreateSearchStr(rowB[2])
        costU = rowB[4]
        costD = rowB[3]
        user = rowB[1]
        t=(costU,costD,)

        search_New_Product(user, title, t, searchStr)
        rowB = cursorUser.fetchone()

    cursor.execute("DROP TABLE temp")
    conn.commit()

conn.commit()

cursorUser.execute("SELECT * FROM 'Exist_Request_Table' WHERE Type = 'computer'");
rowCom = cursorUser.fetchone()

if rowCom is None:
    print('No Computer Request exist')

else:
    conn = sqlite3.connect('boyclothes.sqlite')
    cursor = conn.cursor()

    cursor.execute("SELECT * FROM sqlite_master WHERE type = 'table' AND name = ?;",('temp',))
    exists = bool(cursor.fetchone())
    conn.commit()

    if exists:
        while rowCom is not None:
            title = rowCom[2]
            searchStr = CreateSearchStr(rowCom[2])
            costU = rowCom[4]
            costD = rowCom[3]
            user = rowCom[1]
            t=(costU,costD,)

            search_New_Product(user, title, t, searchStr)

```



```

        rowCom = cursorUser.fetchone()

        cursor.execute("DROP TABLE temp")
        conn.commit()

    conn.commit()

def search_New_Product(name, product, t, searchStr):
    costU = t[0]
    costD = t[1]
    cursor.execute(searchStr); #執行搜尋，抓出符合的產品之價格
    GotRow = cursor.fetchone()

    if GotRow is None:
        print(name, 'No product meet your request')

    else:
        hit = False #確認是否有符合需求之商品存在
        while GotRow is not None:
            price = int(GotRow[0])
            arr = []
            if ( price < costU ) and ( price > costD ):
                if not hit:
                    hit = True #首次符合需求，hit改為true

                arr.append(GotRow[1]) #將此商品之資訊記錄下來
                GotRow = cursor.fetchone()
            if hit:
                sending(product, arr, name) #若有符合(hit is true)，則通知使用者
            print('\n')

def CreateSearchStr(title):
    product = title
    searchStr = "SELECT price, website FROM temp WHERE title like '%" + product + "%' or"

    tmp = product.split() #初次以' '(blankspace)切割
    tmp = list(filter(None, tmp)) #移除空字串
    SpaceLen = len(tmp) #分割之數目

```

if SpaceLen > 0: #存在字串(非空)

#空白切割之首元素搜尋(預期完全未切割字串為重要字串組)

searchStr = searchStr + "(title LIKE '%" + tmp[0] + "%"

subStr = re.split('\\d+', tmp[0], 1) #以連續數字為切割符，且保留切割符，最多切1次

subStr = list(filter(**None**, subStr))

subLen = len(subStr)

if subLen == 3: #子字串有3個，為str、digit、str此種形式

name = subStr[0]

num = subStr[1]

nextWord = subStr[2]

numAndNextWord = subStr[1] + subStr[2]

searchStr = searchStr + " or (title LIKE '%" + name + "' and title like '%" + numAndNextWord + "%"

or (title like '%" + num + "' and title like '%" + nextWord + "%'))"

searchStr = searchStr + ")"

#空白切割之首字串以外的字串

if SpaceLen > 1:

#未切割的字串為重要字串組

searchStr = searchStr + " and ("

for countTmp **in** range(1, SpaceLen):

#未切割字串嵌入

if countTmp != 1:

searchStr = searchStr + " or "

searchStr = searchStr + "title LIKE '%" + tmp[countTmp] + "%"

subStr = re.split('\\d+', tmp[countTmp], 1)

subStr = list(filter(**None**, subStr))

subLen = len(subStr)

if subLen == 3:

name = subStr[0]

num = subStr[1]

nextWord = subStr[2]

numAndNextWord = subStr[1] + subStr[2]

```

        searchStr = searchStr + " or ( title LIKE '%" + name + "%' and title like '%" +
numAndNextWord + "%' or ( title like '%" + num + "%' and title like '%" + nextWord + "%'  ) )"
        searchStr = searchStr + ")"
    return searchStr

def sending(product,arr,user): #寄信
    t=('%'+user+'%')
    cursorUser.execute("SELECT * FROM 'UserData' WHERE username like ?",t) #抓出使用者訊息
    row = cursorUser.fetchone()

    gmail_user = 'SenderOfProject@gmail.com'#網站用信箱
    gmail_password = '94879487'

    from_add = gmail_user
    to = row[2]#使用者信箱
    subject = 'We found some ' + product + ' meet your request'

    things = "Hello " + user + ",\n\t Links below are url of product that meet your request.\n\n"
    count = 1
    for url in arr:
        things = things + str(count) + ". " + url + "\n"
        count += 1

    msg=MIMEText(things)
    msg['Subject']=subject
    msg['From']= gmail_user
    msg['To']=to

    try:
        smtpObj = smtplib.SMTP('smtp.gmail.com', 587)
        smtpObj.ehlo()
        smtpObj.starttls()
        smtpObj.login(gmail_user, gmail_password)
        smtpObj.sendmail(from_add, to, msg.as_string())
        smtpObj.close()
        print ('Email sent')
    except:
        print ('Something went wrong')

main()

```

B. 附件二

```
# -*- coding: utf-8 -*-
```

```
"""
```

Spyder Editor

This is a temporary script file.

```
"""
```

```
import sqlite3
```

```
import re
```

```
import string
```

```
import smtplib
```

```
from email.mime.text import MIMEText
```

```
conn = sqlite3.connect('cellphone.sqlite')
```

```
cursor = conn.cursor()
```

```
connUser = sqlite3.connect('info.sqlite')
```

```
cursorUser = connUser.cursor()
```

```
def main():
```

```
    a = '1'
```

```
    b = True
```

```
    while a == '1':
```

```
        if b:
```

```
            #測資當b == true -> nothing fund
```

```
            conn = sqlite3.connect('cellphone.sqlite')
```

```
            cursor = conn.cursor()
```

```
            cursor.execute("UPDATE cellphone SET price = 100000 WHERE title LIKE '%%'")
```

```
            conn.commit()
```

```
            b = not b#隔次對調
```

```
t = new_Request_In()#新進需求確認
```

```
if t != None:
```

```
    new_Request_search(t) #需求處理
```

```
t = modify_Request_In()#修改需求確認
```

```
if t != None:
```

```
new_Request_search(t)
```

```
conn.commit()
```

```
cursor.close()
```

```
conn.close()
```

```
connUser.commit()
```

```
cursorUser.close()
```

```
connUser.close()
```

```
print('the End.')
```

```
def modify_Request_In():
```

```
    cursorUser.execute("SELECT name FROM sqlite_master WHERE type = 'table' AND name = ?;",('Modify',)) #定義modify表為資料儲存處，檢查表格存在與否
```

```
    exists = bool(cursorUser.fetchone())
```

```
    if exists:
```

```
        #抓出需求並回傳
```

```
        cursorUser.execute("SELECT * from Modify")
```

```
        row = cursorUser.fetchone()
```

```
        cursorUser.execute("DROP TABLE Modify")
```

```
        connUser.commit()
```

```
        return row
```

```
    else:
```

```
        return None
```

```
def new_Request_In():
```

```
    cursorUser.execute("SELECT name FROM sqlite_master WHERE type = 'table' AND name = ?;",('temp',)) #確認有無新的需求
```

```
    exists = bool(cursorUser.fetchone())
```

```
    connUser.commit()
```

```
    if exists:
```

```
        cursorUser.execute("SELECT * from temp")
```

```
        row = cursorUser.fetchone()
```

```
        connUser.commit()
```

```
        title = row[2]
```

#新需求需要檢查合法性，先移除符號但保留空白

```
for char in string.punctuation:
```

```
    title = title.replace(char, ")
```

#避免需求非法，檢查移除符號後的字串是否非空

```
checkEmptyStr = title.split()
```

```
checkEmptyStr = list(filter(None, checkEmptyStr))
```

```
if len(checkEmptyStr) == 0:
```

```
    return None
```

```
cat = row[0]
```

```
name = row[1]
```

```
costD = int(row[3]) #價格型別轉換，前端傳入為char
```

```
costU = int(row[4])
```

```
if costU < costD: #確保上下界輸入正確
```

```
    tmpCost = costU
```

```
    costU = costD
```

```
    costD = tmpCost
```

```
y = row[5]
```

```
m = row[6]
```

```
d = row[7]
```

```
cursorUser.execute("DROP TABLE temp")#移除新需求temp表
```

```
connUser.commit()
```

#需求新增至Exist_Request_Table

```
rowCount = cursorUser.execute("SELECT COUNT(*) FROM Exist_Request_Table")
```

```
rowCount = rowCount.fetchone()[0] + 1#賦予新需求與其他需求互斥之ID
```

```
connUser.commit()
```

```
rQ = (cat, name, title, costD, costU, y, m, d, rowCount)
```

```
cursorUser.execute("INSERT INTO Exist_Request_Table VALUES (?,?,, ?, ?, ?, ?, ?)",rQ)
```

```
connUser.commit()
```

#更新歷史資料

```
ck = (name,)
```

```
cursorUser.execute("SELECT * FROM UserData WHERE username = ?",ck);
```

```

row = cursorUser.fetchone()[3]
connUser.commit()

if len(row) == 0: #新使用者新建歷史資料
    newStr = title
else:
    newStr = row + ';' + title
ss = (newStr, name)
cursorUser.execute("UPDATE 'UserData' SET history = ? WHERE username = ?", ss)
connUser.commit()

t = (cat, name, title, costD, costU,)
return t
else:
    return None

```

```
def new_Request_search(t):
```

```

    cat=t[0]
    user = t[1]
    product = t[2]

```

```

    costD = int(t[3])
    costU = int(t[4])

```

```

if costU < costD:
    tmpCost = costU
    costU = costD
    costD = tmpCost

```

```
print(t)
```

```
searchStr = "SELECT price, website FROM " + cat + " WHERE title like '%" + product + "%' or"
```

```

tmp = product.split()
tmp = list(filter(None, tmp))
SpaceLen = len(tmp)

```

```

if SpaceLen > 0:
    searchStr = searchStr + "( title LIKE '%" + tmp[0] + "%'"
    subStr = re.split('\d+', tmp[0], 1)

```

```

subStr = list(filter(None, subStr))
subLen = len(subStr)

if subLen == 3:
    name = subStr[0]
    num = subStr[1]
    nextWord = subStr[2]
    numAndNextWord = subStr[1] + subStr[2]
    searchStr = searchStr + " or ( title LIKE '%" + name + "%' and title like '%" + numAndNextWord + "%'
or ( title like '%" + num + "%' and title like '%" + nextWord + "%' )"

searchStr = searchStr + ")"

if SpaceLen > 1:
    searchStr = searchStr + " and ( "

    for countTmp in range(1, SpaceLen):
        if countTmp != 1:
            searchStr = searchStr + " or "

        searchStr = searchStr + "title LIKE '%" + tmp[countTmp] + "%"

        subStr = re.split('\\d+', tmp[countTmp], 1)
        subStr = list(filter(None, subStr))
        subLen = len(subStr)

        if subLen == 3:
            name = subStr[0]
            num = subStr[1]
            nextWord = subStr[2]
            numAndNextWord = subStr[1] + subStr[2]
            searchStr = searchStr + " or ( title LIKE '%" + name + "%' and title like '%" +
numAndNextWord + "%' or ( title like '%" + num + "%' and title like '%" + nextWord + "%' ) )"
            searchStr = searchStr + ")"

if cat == 'cellphone': #切換資料庫
    conn = sqlite3.connect('cellphone.sqlite')
    cursor = conn.cursor()

if cat == 'computer':

```



```

conn = sqlite3.connect('computer.sqlite')
cursor = conn.cursor()
if cat == 'boyclothes':
    conn = sqlite3.connect('boyclothes.sqlite')
    cursor = conn.cursor()

cursor.execute(searchStr) #搜尋
row = cursor.fetchone()
print(4)
#判別式 row is None時，表示沒找到
if row is None:
    print('Not Found')

else:
    hit = False
    while row is not None:
        price = int(row[0])
        arr = []
        if ( price < costU ) and ( price > costD ):
            if not hit:
                hit = True
                arr.append(row[1])

        row = cursor.fetchone()
    if hit:
        sending(product,arr,user)
conn.commit()

```

```

def sending(product,arr,user):
    print("sendStart")
    t=(user,)
    print(user)
    cursorUser.execute("SELECT * FROM 'UserData' WHERE username = ?",t)
    row = cursorUser.fetchone()

    gmail_user = 'SenderOfProject@gmail.com'
    gmail_password = '94879487'

    from_add = gmail_user

```

```

to = row[2]
subject = 'We found some ' + product + ' meet your request'

things = "Hello " + user + ",\n\tLinks below are url of product that meet your request.\n\n"
count = 1
for url in arr:
    things = things + str(count) + ". " + url + "\n"
    count += 1

msg=MIMEText(things)
msg['Subject']=subject
msg['From']= gmail_user
msg['To']=to

try:
    smtpObj = smtplib.SMTP('smtp.gmail.com', 587)
    smtpObj.ehlo()
    smtpObj.starttls()
    smtpObj.login(gmail_user, gmail_password)
    smtpObj.sendmail(from_add, to, msg.as_string())
    smtpObj.close()
    print ('Email sent')
except:
    print ('Something went wrong')

main()

```

C. 附件三

```
#-*- coding: UTF-8 -*-
```

```
from bs4 import BeautifulSoup
```

```
import scrapy
```

```
#from scrapy.spiders import CrawlSpider, Rule
```

```
#from scrapy.linkextractors import LinkExtractor
```

```
from selenium import webdriver
```

```
import time
```

```
import lxml
```

```
import sys
```

```
from scrapy.selector import Selector
```

```
import requests
```

```
import shutil
```

```
import sqlite3
```

```
from sky.items import CellphoneItem
```

```
import os
```

```
class cellphone(scrapy.Spider):
```

```
    name = 'cellphone'
```

```
    domain = ['http://www.ruten.com.tw/']
```

```
    #set the url you want to search
```

```
    start_urls = [
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210001&p=,#手機空機
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210003&p=,#手機門號
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210010&p=,#門號專案機
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210019&p=,#行動電源
```

```
        'http://class.ruten.com.tw/category/sub00.php?c=00210002&p=,#手機配件
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210016&p=,#電話卡、上網卡
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210017&p=,#租賃、出租
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210021&p=,#穿戴裝置
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210014&p=,#無線電設備
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210023&p=,#手機保養
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210022&p=,#貿易製造
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210018&p=,#代買、海外代購
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210012&p=,#其他
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210011&p=,#答錄機
```

```
        #http://class.ruten.com.tw/category/sub00.php?c=00210013&p=,#模型、樣品機
```

```
    ]
```

```
    urlpage = 0
```

```

db_title = []
db_price = []
db_website = []
db_date = []
db_flag = []
point = 0
count = 0
results = 0
temp_count = 0
db_count = 0
flag = 1
HEADER={
    #"Host": "www.ruten.com.tw/",
    #"Connection": "keep-alive",
    #"Cache-Control": "max-age=0",
    #"Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
    "User-Agent": "Mozilla/4.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/60.0.3112.90 Safari/537.36",
    #"Referer": "http://www.zhihu.com/people/raymond-wang",
    #"Accept-Encoding": "gzip,deflate,sdch",
    #"Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.6,en;q=0.4,zh-TW;q=0.2",
}

def __init__(self):
    #self.driver = webdriver.Chrome(executable_path=os.getcwd() +
r"\selenium_driver_chrome\chromedriver.exe') # chrome 瀏覽器
    #self.driver = webdriver.PhantomJS(executable_path=os.getcwd()
+r"\phantomjs-2.1.1-windows\bin\phantomjs.exe') #phantomjs
    self.conn = sqlite3.connect('cellphone.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists cellphone(title varchar(50) , price varchar(10) , website
varchar(100) , date varchar(100), flag bit);')
    print("init")
    try:
        self.cur.execute('UPDATE cellphone SET flag = 0;')
        self.conn.commit()
        print("update the flag.")
    except:
        print("not update the flag.")

```

```

        #self.conn.close()
        print("database close")
    def start_requests(self):
        #self.driver = webdriver.Chrome(executable_path=os.getcwd() +
        r"\selenium_driver_chrome\chromedriver.exe") # chrome 瀏覽器

        try:
            self.conn = sqlite3.connect('cellphone.sqlite')
            self.cur = self.conn.cursor()
            self.cur.execute('create table if not exists cellphone(title varchar(50) , price varchar(10) , website
            varchar(100) , date varchar(100), flag bit);')
            self.cur.execute('SELECT * FROM cellphone;')
            results = self.cur.fetchall()
            #print(count)
            #self.cur.execute('SELECT * FROM cellphone')
            for record in results:
                self.db_title.append(record[0])
                self.db_price.append(record[1])
                self.db_website.append(record[2])
                self.db_date.append(record[3])
            #print(len(self.db_website))
            #try:
            #    self.cur.execute('UPDATE cellphone SET flag = 0;')
            #    self.conn.commit()
            #    print("update the flag.")
            #except:
            #    print("not update the flag.")
            print("success")

        except:
            print("can't print website")
            self.conn.close()
            print("database close")

        self.driver = webdriver.PhantomJS(executable_path=os.getcwd()
        +r"\phantomjs-2.1.1-windows\bin\phantomjs.exe") #phantomjs
        self.driver.get(self.start_urls[self.urlpage])
        yield scrapy.Request(self.start_urls[self.urlpage] , callback = self.parse)
        self.urlpage = self.urlpage + 1

```

```

def parse(self , response):
    for start in self.start_urls:
        #self.driver.get(self.start_urls[urlpage])

    for page in range(0,1):#how many pages want spider
        url = start + str(page+1)
        self.driver.get(url)
        #time.sleep(1)
        #sp = []
        sp = BeautifulSoup(self.driver.page_source.encode(sys.stdin.encoding,
"replace").decode(sys.stdin.encoding) , "xml")
        #sp = BeautifulSoup(self.driver.page_source , "xml")
        page += 1
        #items = []
        items = sp.find_all('li',{'itemtype':'http://schema.org/Product'})

    for item in items: #catch the product's data
        title = item.find('a',{'class':'rt-goods-list-item-name-link'}).getText()
        price = item.find('strong',{'class':'rt-text-price'}).getText()
        website = item.find('a',{'class':'rt-goods-list-item-name-link'})['href']
        date = time.strftime("%Y/%m/%d/%H:%M:%S")
        #print(title + price)

    if len(self.db_website) == 0:
        #website no data , input data in database
        yield self.parse_detail(title ,price,website ,date,self.flag)
    else:
        for i in range(len(self.db_website)):
            if not (website == self.db_website[i]):
                #if new data not in the old database
                self.count = self.count + 1
                #find the different item
            else:
                #print("find the same item's website")
                #insert new price into the another table
                if not (price == self.db_price[i]):
                    #-----insert the new data in new table-----#
                    self.conn = sqlite3.connect('cellphone.sqlite')

```

```

        self.cur = self.conn.cursor()
        self.cur.execute('create table if not exists temp(title varchar(50) , price
varchar(10) , website varchar(100) , date varchar(100),flag bit);')
        self.cur.execute('insert into temp values (?, ?, ?, ?);',(title , price , website ,
date,self.flag))

        self.conn.commit()
        self.conn.close()

    if not os.path.exists('true.txt'):#close spider , write a true.txt
        open('true.txt', 'w').close()
    else:
        print("true exists")
        print("insert data success.")
        #-----end-----#
#-----update the new data into the old database-----#
        self.conn = sqlite3.connect('cellphone.sqlite')
        self.cur = self.conn.cursor()
        self.cur.execute('update cellphone set price = ? where website
= ?;',(price,website))

        self.cur.execute('update cellphone set flag = ? where website
= ?;',(self.flag,website))

        self.conn.commit()
        self.conn.close()

        print("update data success.")
        #-----end-----#
    else:#if find the same website and price same , update the flag
        self.conn = sqlite3.connect('cellphone.sqlite')
        self.cur = self.conn.cursor()
        self.cur.execute('update cellphone set flag = ? where website
= ?;',(self.flag,website))

        self.conn.commit()
        self.conn.close()

        print("update flag success.")
        #if not find the same item , insert the new item into the database
        if (self.count == len(self.db_website)):
            self.conn = sqlite3.connect('cellphone.sqlite')
            self.cur = self.conn.cursor()

```

```

        self.cur.execute('create table if not exists temp(title varchar(50) , price
varchar(10) , website varchar(100) , date varchar(100),flag bit);')
        self.cur.execute('insert into temp values (?, ?, ?, ?, ?);',(title , price , website ,
date,self.flag))

        self.conn.commit()
        self.conn.close()

        print("title = " + title)
        print("insert into database")
        if not os.path.exists('true.txt'):

#close spider , write a true.txt

            open('true.txt', 'w').close()
        else:
            print("true exists")
            yield self.parse_detail(title ,price,website ,date , self.flag)

        #print(self.count)
        self.count = 0
'''
#修改資料庫
try:
    self.conn = sqlite3.connect('cellphone.sqlite')
    self.cur = self.conn.cursor()
    #self.cur.execute('create table if not exists temp(title varchar(50) , price varchar(10) , website
varchar(100) , date varchar(100));')
    self.cur.execute('UPDATE cellphone SET price = 11111 where website =
"http://goods.ruten.com.tw/item/show?21533358755580";')
    self.cur.execute('UPDATE cellphone SET price = 11111 where website =
"http://goods.ruten.com.tw/item/show?21731843567052";')
    self.cur.execute('DELETE from cellphone where website =
"http://goods.ruten.com.tw/item/show?21642017617737";')
    self.conn.commit()
    self.conn.close()
    print("update data success.")
except:
    print("can't update data")
'''

def parse_detail(self,title ,price , website,date,flag):
    Item = CellphoneItem()

```



```
Item['title'] = title
Item['price'] = price
Item['website'] = website
Item['date'] = date
Item['flag'] = flag
return Item
```

```
def spider_closed(self, spider):
    self.driver.quit()
```

D. 附件四

```
##-*- coding: UTF-8 -*-
from bs4 import BeautifulSoup
import scrapy

#from scrapy.spiders import CrawlSpider , Rule
#from scrapy.linkextractors import LinkExtractor

from selenium import webdriver
import time
import lxml
import sys

from scrapy.selector import Selector
import requests
import shutil
import sqlite3

from sky.items import pchome_3cItem
import os

class pchome_3c(scrapy.Spider):
    name = 'pchome_3c'
    domain = ['http://24h.pchome.com.tw/']
    #set the url you want to search
    start_urls = [
        'http://24h.pchome.com.tw/region/DSAA',#桌 上 電 腦
        'http://24h.pchome.com.tw/region/DSAM',#商 用 電 腦
        'http://24h.pchome.com.tw/region/DSAU',#DIY 電腦
        'http://24h.pchome.com.tw/region/DSAB',#LCD 螢幕
        'http://24h.pchome.com.tw/region/DRAF',#網 路
        'http://24h.pchome.com.tw/region/DRAG',#網 路 硬 碟
        'http://24h.pchome.com.tw/region/DSAE',#電腦軟體
        'http://24h.pchome.com.tw/region/DCAS',#視訊監控
        'http://24h.pchome.com.tw/region/DCAR',#U P S
        'http://24h.pchome.com.tw/region/DRAB',#內接硬碟
        'http://24h.pchome.com.tw/region/DRAA',#外接硬碟
        'http://24h.pchome.com.tw/region/DRAH',#S S D
        'http://24h.pchome.com.tw/region/DGAG',#記 憶 卡
        'http://24h.pchome.com.tw/region/DGCD',#隨 身 碟
        'http://24h.pchome.com.tw/region/DRAI',#C P U
        'http://24h.pchome.com.tw/region/DRAC',#記憶體
        'http://24h.pchome.com.tw/region/DSAJ',#主機板
```

'http://24h.pchome.com.tw/region/DRAD',#顯示卡
'http://24h.pchome.com.tw/region/DRAE',#電腦機殼
'http://24h.pchome.com.tw/region/DSAZ',#電源供應器
'http://24h.pchome.com.tw/region/DCAX',#電腦擴充
'http://24h.pchome.com.tw/region/DSAX',#電 競 專 區
'http://24h.pchome.com.tw/region/DCAE',#印 表 機
'http://24h.pchome.com.tw/region/DCAF',#耗 材
'http://24h.pchome.com.tw/region/DCAC',#線 材
'http://24h.pchome.com.tw/region/DSBA',#D-Link
'http://24h.pchome.com.tw/region/DRAL',#Kingston
'http://24h.pchome.com.tw/region/DGCA',#SanDisk
'http://24h.pchome.com.tw/region/DRAM',#Seagate
'http://24h.pchome.com.tw/region/DGBN',#創 見
'http://24h.pchome.com.tw/region/DSAD',#Skype
'http://24h.pchome.com.tw/region/DSAW',#台 灣 精 品
'http://24h.pchome.com.tw/region/DSAF',#清 倉

]

urlpage = 0

db_title = []

db_price = []

db_website = []

db_date = []

db_flag = []

point = 0

count = 0

results = 0

temp_count = 0

db_count = 0

flag = 1

HEADER={

#"Host": "www.ruten.com.tw/",

#"Connection": "keep-alive",

#"Cache-Control": "max-age=0",

#"Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",

"User-Agent": "Mozilla/4.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/60.0.3112.90 Safari/537.36",

#"Referer": "http://www.zhihu.com/people/raymond-wang",

```

#"Accept-Encoding": "gzip,deflate,sdch",
#"Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.6,en;q=0.4,zh-TW;q=0.2",
}

```

```

def __init__(self):
    #self.driver = webdriver.Chrome(executable_path=os.getcwd() +
r"\selenium_driver_chrome\chromedriver.exe') # chrome 瀏覽器
    #self.driver = webdriver.PhantomJS(executable_path=os.getcwd()
+r"\phantomjs-2.1.1-windows\bin\phantomjs.exe') #phantomjs
    self.conn = sqlite3.connect('pchome_3c.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists pchome_3c(title varchar(50) , price varchar(10) , website
varchar(100) , date varchar(100), flag bit);')
    print("init")
    try:
        self.cur.execute('UPDATE pchome_3c SET flag = 0;')
        self.conn.commit()
        print("update the flag.")
    except:
        print("not update the flag.")
    #self.conn.close()
    print("database close")

def start_requests(self):
    #self.driver = webdriver.Chrome(executable_path=os.getcwd() +
r"\selenium_driver_chrome\chromedriver.exe') # chrome 瀏覽器
    try:
        self.conn = sqlite3.connect('pchome_3c.sqlite')
        self.cur = self.conn.cursor()
        self.cur.execute('create table if not exists pchome_3c(title varchar(50) , price varchar(10) , website
varchar(100) , date varchar(100), flag bit);')
        self.cur.execute('SELECT * FROM pchome_3c;')
        results = self.cur.fetchall()
        #print(count)
        #self.cur.execute('SELECT * FROM pchome_3c')
        for record in results:
            self.db_title.append(record[0])
            self.db_price.append(record[1])
            self.db_website.append(record[2])
            self.db_date.append(record[3])

```

```

# print(len(self.db_website))

# try:
#     self.cur.execute('UPDATE pchome_3c SET flag = 0;')
#     self.conn.commit()
#     print("update the flag.")
# except:
#     print("not update the flag.")
print("success")

```

except:

```

    print("can't print website")
self.conn.close()
print("database close")

```

```

self.driver = webdriver.PhantomJS(executable_path=os.getcwd()
+r"\phantomjs-2.1.1-windows\bin\phantomjs.exe') #phantomjs
self.driver.get(self.start_urls[self.urlpage])
yield scrapy.Request(self.start_urls[self.urlpage] , callback = self.parse)
self.urlpage = self.urlpage + 1

```

def parse(self , response):

```

    for start in self.start_urls:
        #self.driver.get(self.start_urls[urlpage])
        self.driver.get(start)

```

```

    sp = BeautifulSoup(self.driver.page_source.encode(sys.stdin.encoding,
"replace").decode(sys.stdin.encoding) , "lxml")
    #sp = BeautifulSoup(self.driver.page_source , "lxml")
    #items = []
    meta = sp.find('dl',{'id':'Block12Container50'})
    items = meta.find_all('dd',{'class':''})
    for item in items:
        title = item.find('h5').getText()
        price = item.find('span',{'class':'value'}).getText()
        website_com = item.find('a')['href']
        website = 'http://24h.pchome.com.tw' + website_com
        #print(title)
        date = time.strftime("%Y/%m/%d/%H:%M:%S")

```

```

        if len(self.db_website) == 0:    #website no data , input data in database
            yield self.parse_detail(title ,price,website ,date,self.flag)
        else:
            for i in range(len(self.db_website)):
                if not (website == self.db_website[i]):
                    #if new data not in the old database
                    self.count = self.count + 1
                    #find the different item
                else:
                    #print("find the same item's website")
                    #insert new price into the another table
                    if not (price == self.db_price[i]):
                        #-----insert the new data in new table-----#
                        self.conn = sqlite3.connect('pchome_3c.sqlite')
                        self.cur = self.conn.cursor()
                        self.cur.execute('create table if not exists temp(title varchar(50) , price
                        varchar(10) , website varchar(100) , date varchar(100),flag bit);')
                        self.cur.execute('insert into temp values (?, ?, ?, ?, ?);',(title , price , website ,
                        date,self.flag))

                        self.conn.commit()
                        self.conn.close()
                        if not os.path.exists('true.txt'):#close spider , write a true.txt
                            open('true.txt', 'w').close()
                        else:
                            print("true exists")
                            print("insert data success.")
                            #-----end-----#
                        #-----update the new data into the old database-----#
                        self.conn = sqlite3.connect('pchome_3c.sqlite')
                        self.cur = self.conn.cursor()
                        self.cur.execute('update pchome_3c set price = ? where website
                        = ?;',(price,website))

                        self.cur.execute('update pchome_3c set flag = ? where website
                        = ?;',(self.flag,website))

                        self.conn.commit()
                        self.conn.close()
                        print("update data success.")

```

```

#-----end-----#
else:#if find the same website and price same , update the flag
    self.conn = sqlite3.connect('pchome_3c.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('update pchome_3c set flag = ? where website
= ?',(self.flag,website))

    self.conn.commit()
    self.conn.close()
    print("update flag success.")

#if not find the same item , insert the new item into the database
if (self.count == len(self.db_website)):
    self.conn = sqlite3.connect('pchome_3c.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists temp(title varchar(50) , price varchar(10) ,
website varchar(100) , date varchar(100),flag bit);')
    self.cur.execute('insert into temp values (?,?,?,?,?);',(title , price , website ,
date,self.flag))

    self.conn.commit()
    self.conn.close()
    print("title = " + title)
    print("insert into database")
    yield self.parse_detail(title ,price,website ,date , self.flag)

#print(self.count)
self.count = 0

'''修改資料庫
try:
    self.conn = sqlite3.connect('pchome_3c.sqlite')
    self.cur = self.conn.cursor()
    #self.cur.execute('create table if not exists temp(title varchar(50) , price varchar(10) , website
varchar(100) , date varchar(100));')
    self.cur.execute('UPDATE pchome_3c SET price = 11111 where website =
"http://goods.ruten.com.tw/item/show?21309125836112";')
    self.conn.commit()
    self.conn.close()
    print("update data success.")
except:
    print("can't update data")
'''

```

```
def parse_detail(self,title ,price , website,date,flag):
```

```
    Item = pchome_3cItem()
```

```
    Item['title'] = title
```

```
    Item['price'] = price
```

```
    Item['website'] = website
```

```
    Item['date'] = date
```

```
    Item['flag'] = flag
```

```
    return Item
```

```
def spider_closed(self, spider):
```

```
    self.driver.quit()
```


E. 前端帳號登入與相關操作程式碼

E.1. 登入

```
<!DOCTYPE html>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>
<link href="layout.css" rel="stylesheet" type="text/css">
</head>

<body>
<!--頁首區-->
<div id="HEADER">
    <h1>OO 比價網</h1>
</div>

<!--內容區-->
<div id="CONTENT">
    <!--使用者登入頁面-->
    <form enctype="multipart/form-data" action="connect.php" method="post">
    <p style="font-size: 30px; padding-left: 45%; ">登入</p><br>
    <P style="font-size: 24px; padding-left: 35%">帳號：<input type="text" name="id" value="" size="20"
></P><br>
    <P style="font-size: 24px; padding-left: 35%">密碼：<input type="password" name="pw" value=""
size="20"></P><br><br>
    <button style="height: 30px; width: 60px; font-size: 20px; align-items: center; position: relative; left: 360px  "
type="submit" name="登入">登入</button>
    <button style="height: 30px; width: 60px; font-size: 20px; align-items: center; position: relative; left: 440px  "
type="button" name="註冊" onclick="self.location.href='register.php'">註冊</button>
    </form>

</div>
<!--頁尾區-->
<div id="FOOTER">
    <h2>資料主要來源</h2>
```

```

<pre style="font-size:15px">
<p style="color:#FFF;padding-left: 20%">Pchome
露天拍賣</p>
<a style="" href="http://shopping.pchome.com.tw/"></a>
href="https://tw.bid.yahoo.com/"></a>
<a href="http://www.ruten.com.tw/"></a>
</pre>
</div>

```

```
</body>
```

```
</html>
```

(顯示給使用者登入的頁面)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>無標題文件</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
/*連接資料庫*/
```

```
try{
```

```
    $db_conn = new PDO('sqlite:info.sqlite');
```

```
}catch(PDOException $e){
```

```
    echo 'Error';
```

```
}
```

```
@$id = $_POST['id'];/*使用者 ID*/
```

```
@$pw = $_POST['pw'];/*使用者密碼*/
```

```
$stmt=$db_conn->prepare( "SELECT * FROM UserData where username = '$id'");/*與資料庫做比對把與使用者輸入的 ID 相符的拿出來比較 有比對相符顯示登入成功*/
```

```
$stmt->execute();
```

```
$row=$stmt->fetch(PDO::FETCH_BOTH);
```

```

if($id != null && $pw != null && $row[0] == $id && $row[1] == $pw){
    session_start();
    $_SESSION['username'] = $id;
    ?>
    <script>
        alert("登入成功");
        location.replace('一般搜尋.php')
    </script>
    <?php
}
else
{
    ?>

    <script>
        alert("登入失敗，請確認帳密是否輸入正確");
        location.replace('登入.php')
    </script>
    <?php

}

?>
</body>
</html>

```

(使用者輸入帳號密碼之後與資料庫做比對，比對符合，登入成功，比對不符要求重新輸入)

E.2.註冊

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>
<link href="layout.css" rel="stylesheet" type="text/css">
</head>

<body>
<!--頁首區-->
<div id="HEADER">
    <h1>OO 比價網</h1>
</div>

<!--內容區-->
<div id="CONTENT">
    <form enctype="multipart/form-data" action="register_finish.php" method="post" >
    <p style="font-size: 30px; padding-left: 45%; ">註冊</p><br>
    <P style="font-size: 24px; padding-left: 30%">帳號：<input type="text" name="id" value="" size="20"
></P><br>
    <P style="font-size: 24px; padding-left: 30%">密碼：<input type="password" name="pw" value=""
size="20"></P><br>
    <P style="font-size: 24px; padding-left: 30%">再次輸入密碼：<input type="password" name="pw2"
value="" size="20"></P><br>
    <P style="font-size: 24px; padding-left: 30%">信箱：<input type="text" name="mail" value=""
size="20"><br><br>
    <input style="font-size: 20px; margin-top: 20px; margin-left: 15%;" type="Submit" value="確認">
    <input style="font-size: 20px; margin-top: 30px; margin-left: 60px;" type="button" value="取消"
    "onclick=location.replace('登入.php')>
    </p>
    </form>
</div>
<!--頁尾區-->
<div id="FOOTER">
```

<h2>資料主要來源</h2>

<pre style="font-size:15px">

<p style="color:#FFF;padding-left: 20%">Pchome

yahoo 拍賣

露天拍賣</p>

</pre>

</div>

</body>

</html>

(使用者註冊頁面)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>無標題文件</title>

</head>

<body>

<?php

try{

\$db_conn = new PDO('sqlite:info.sqlite');

}catch(PDOException \$e){

echo 'Error';

}

\$id = \$_POST['id'];/*使用者輸入的 ID*/

\$pw = \$_POST['pw'];/*使用者輸入的密碼*/

\$pw2 = \$_POST['pw2'];/*再次確認密碼*/

\$mail = \$_POST['mail'];/*使用者輸入的 mail*/

\$stmt="CREATE TABLE IF NOT EXISTS UserData(username,password,mail,history)";/*新增 UserData 資料庫*/

\$db_conn->exec(\$stmt);

\$stmt=\$db_conn->prepare("SELECT * FROM UserData where username='\$id'");

\$stmt->execute();

\$row = \$stmt->fetch(PDO::FETCH_BOTH);

/*將資料庫裡的 ID 與使用者輸入的 ID 比對一次若有相同請使用者重新輸入，若沒有人使用顯示登入成功*/

if (\$row[0] == \$id)

```

{
    ?>
    <script >
        alert("帳號已有人使用,請重新輸入");
        location.replace('register.php')
    </script>
<?php
}
else{

    if( $id != null && $pw != null && $pw2 != null && $pw == $pw2){
        $stmt = "insert into UserData (username, password,mail,history) values ('$id', '$pw', '$mail',')";
        $db_conn->exec($stmt);
        ?>
        <script >
            alert("註冊成功");
            location.replace('登入.php')
        </script>
        <?php
    }
    else
    {
        ?>
        <script>
            alert("確認密碼是否一致或空白");
            location.replace('register.php')
        </script>
        <?php
    }
}
?>
</body>
</html>

```

(使用者輸入後，與資料庫做比對看是否有人用過此 ID，若有重複，請使用者重新輸入 ID，若沒有重複，顯示註冊成功)

E.3. 搜尋

```
<!DOCTYPE html>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>
<link href="layout.css" rel="stylesheet" type="text/css">
</head>

<body>
<!--頁首區-->
<div id="HEADER">
    <h1>OO 比價網</h1>
</div>

<!--選單區-->
<div id="NAV">

<!--內容區-->
<div id="CONTENT">
    <pre style="font-size:24px">
<a
href="history.php">歷史資料</a><a></a><a href="data.php">訂閱資料</a><a></a><a href="member.php">
會員資料</a><a></a><a href="logout.php">登出</a>

    <form action="一般搜尋 finish.php" method="post" >
        產品種類:  <select style="height: 40PX;font-size: 20px" name="Type" >
            <option sytle=fontsize value="cellphone">手機</option>
            <option value="computer">電腦</option>
            <option value="boyclothes">男裝</option>
        </select>

        產品名稱: <input style="height: 24PX" type="text" name="產品名稱" size="30">

        價格區間: <input style="height: 24PX;" type="text" name="價格下限" size="15"> ~
<input style="height: 24PX" type="text" name="價格上限" size="15">
```

結束日期：</datalist> <input type="date" list="vacation-days" name="結束日期">

<datalist id="vacation-days">

<option label="Thanksgiving">2017-10-13</option>

<option label="Christmas Day">2017-12-25</option>

</datalist>

<button style="height: 40px; width: 60px; font-size: 20px; " type="submit">加入</button>

</form>

</pre>

</div>

<!--頁尾區-->

<div id="FOOTER">

<h2>資料主要來源</h2>

<pre style="font-size: 15px">

<p style="color: #FFF; padding-left: 20%">Pchome
露天拍賣</p>

</pre>

</div>

</body>

</html>

(使用者搜尋頁面)


```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>無標題文件</title>
</head>

<body>
<?php
try{
    $db_conn = new PDO('sqlite:info.sqlite');
} catch(PDOException $e){
    echo 'Error';
}
session_start();
$un=$_SESSION['username'];/*使用者 ID*/
$ty=$_POST['Type'];/*使用者欲訂閱商品種類*/
$pid=$_POST['產品名稱'];/*使用者欲訂閱商品名稱*/
$up = $_POST['價格上限'];/*價格上限*/
$dp = $_POST['價格下限'];/*價格下限*/
$ed = $_POST['結束日期'];/*結束日期*/

$stmt="CREATE TABLE IF NOT EXISTS temp(Type TEXT,username TEXT,product TEXT,lowerlimit
TEXT,higherlimit TEXT,year INTEGER,month INTEGER,day INTEGER)";/*新增一個 Temp 資料表*/
$db_conn->exec($stmt);
if($ty!= null && $pid != null && $up != null && $dp != null && $ed !=null){
    $a=explode('-', $ed);
    $sql = "insert into temp(Type,username,product,lowerlimit,higherlimit,year,month,day) values
('$ty','$un','$pid','$dp','$up','$a[0]','$a[1]','$a[2]')";
    $db_conn->exec($sql);

    ?>
<script >
    alert("加入訂閱成功");
    location.replace('一般搜尋.php')
</script>
<?php
}

```

```
else
{
    ?>
    <script>
        alert("確認是否有空白");
        location.replace('一般搜尋.php')
    </script>
    <?php
}
?>
```

```
</body>
```

```
</html>
```

(將使用者輸入的訂閱資料，加到 temp 資料表裡)

E.4.會員資料

```
<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>

<link href="layout.css" rel="stylesheet" type="text/css">

</head>

<body>

<!--頁首區-->

<div id="HEADER">
    <h1>OO 比價網</h1>
</div>

<!--內容區-->

<div id="CONTENT">

<?php
try{
    $db_conn = new PDO('sqlite:info.sqlite');
} catch(PDOException $e){
    echo 'Error';
}

session_start();
$id=$_SESSION['username'];
$stmt=$db_conn->prepare( "SELECT * FROM UserData where username = '$id'");
$stmt->execute();
$row=$stmt->fetch(PDO::FETCH_BOTH);
echo "<p style='font-size:30px;padding-left:45%'>".$_SESSION['username']. "用戶".<br><br>.</p>";
echo "<p style='font-size:30px;padding-left:35%'>". "帳號:". $row[0].<br><br>.</p>";
echo "<p style='font-size:30px;padding-left:35%'>". "信箱:". $row[2].<br><br>.</p>";
?>
```

```

<button style="height: 40px; width:60px ;font-size: 20px; align-items: center; position:relative;left:360px  "
type="button" name="修改" onclick="self.location.href='update.php'">修改</button>
<button style="height: 40px; width:60px ;font-size: 20px; align-items: center; position:relative;left:500px  "
type="button" name="修改" onclick="self.location.href='delete.php'">刪除</button>
</div>
<!--頁尾區-->
<div id="FOOTER">
    <h2>資料主要來源</h2>
    <pre style="font-size:15px">
        <p style="color:#FFF;padding-left: 20%">Pchome                yahoo 拍賣
        露天拍賣</p>
                                <a style="" href="http://shopping.pchome.com.tw/"></a>                                <a
href="https://tw.bid.yahoo.com/"></a>
<a href="http://www.ruten.com.tw/"></a>
    </pre>
</div>

</body>
</html>
(顯示會員資料)

```

E.5.修改會員資料

```
<!DOCTYPE html>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>
<link href="layout.css" rel="stylesheet" type="text/css">
</head>

<body>
<!--頁首區-->
<div id="HEADER">
    <h1>OO 比價網</h1>
</div>

<!--內容區-->
<div id="CONTENT">
    <form enctype="multipart/form-data" action="update_finish.php" method="post" >
    <p style="font-size: 30px; padding-left: 45%; ">修改會員資料</p><br>
    <P style="font-size: 24px; padding-left: 30 %">欲修改密碼：<input type="password" name="pw" value=""
size="20"></P><br>
    <P style="font-size: 24px; padding-left: 30 %">再次輸入密碼：<input type="password" name="pw2"
value="" size="20"></P><br>
    <P style="font-size: 24px; padding-left: 30 %">欲修改信箱：<input type="text" name="mail" value=""
size="20"><br><br>
    <pre>
<button style="height: 40px; width: 60px; font-size: 20px; align-items: center; " type="submit" name="register"
>確認</button></pre>

    </p>
    </form>
</div>
<!--頁尾區-->
<div id="FOOTER">
    <h2>資料主要來源</h2>
    <pre style="font-size: 15px">
```

```
<p style="color:#FFF;padding-left: 20%">Pchome  
露天拍賣</p>
```

yahoo 拍賣

```
<a style="" href="http://shopping.pchome.com.tw/"></a> <a  
href="https://tw.bid.yahoo.com/"></a>  
<a href="http://www.ruten.com.tw/"></a>  
</pre>  
</div>
```

```
</body>
```

```
</html>
```

(使用者修改會員資料頁面)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>無標題文件</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
try{
```

```
    $db_conn = new PDO('sqlite:info.sqlite');
```

```
}catch(PDOException $e){
```

```
    echo 'Error';
```

```
}
```

```
$pw = $_POST['pw'];/*使用者欲修改的密碼*/
```

```
$pw2 = $_POST['pw2'];/*再次確認的密碼*/
```

```
$mail = $_POST['mail'];/*欲修改的 mail*/
```

```
session_start();
```

```
/*檢查是否空白，若無空白將資料更新到資料庫*/
```

```
if( $pw != null && $pw2 != null && $mail!=null && $pw == $pw2){
```

```
    $id = $_SESSION['username'];
```

```
    $sql = "UPDATE UserData set password='$pw', mail='$mail' where username='$id'";
```

```
    $db_conn->exec($sql);
```

```
    ?>
```

```
<script>
```

```
        alert("修改成功");
        location.replace('一般搜尋.php');
    </script>
    <?php

}
else
{
    ?>
    <script>
        alert("修改失敗 請確認密碼是否一致或空白");
        location.replace("update.php");
    </script>
    <?php

}
?>

</body>
</html>
(將使用者欲修改的會員資料更新到資料庫)
```

E.6.現有需求顯示

```
<!DOCTYPE html>

<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>
<link href="layout.css" rel="stylesheet" type="text/css">
</head>

<body>
<!--頁首區-->
<div id="HEADER">
    <h1>OO 比價網</h1>
</div>

<!--選單區-->
<div id="NAV">

<!--內容區-->
<div id="CONTENT">
    </td>

<?php
try{
    $db_conn = new PDO('sqlite:info.sqlite');
}catch(PDOException $e){
    echo 'Error';
}

session_start();
$id=$_SESSION['username'];
$stmt=$db_conn->prepare("SELECT * FROM Exist_Request_Table where username='$id'");
$stmt->execute();
echo "<p style='font-size:50px;padding-left:40%;color:red>". "現有需求."<br><br>."</p>";
echo "<p style='font-size:30px;padding-left:45%>".$_SESSION['username']. "用戶."<br><br>."</p>";
?>
```



```

<form action="dataupdate.php" method="Post">
  <table style=" font-size: 24px; margin-left: 30%; border-width:2px; border-style:solid;
border-color:#000000;">
    <tr style=" border-width:2px; border-style:solid; border-color:#000000;">
      <th style=" border-width:2px; border-style:solid; border-color:#000000;"></th>
      <th style=" border-width:2px; border-style:solid; border-color:#000000;">產品名稱</th>
      <th style=" border-width:2px; border-style:solid; border-color:#000000;">價格下限</th>
      <th style=" border-width:2px; border-style:solid; border-color:#000000;">價格上限</th>
      <th style=" border-width:2px; border-style:solid; border-color:#000000;">截止日期</th>
    </tr>
    <?php
      while($row=$stmt->fetch(PDO::FETCH_BOTH))
      {
        ?>
        <tr style="text-align: center" style=" border-width:2px; border-style:solid; border-color:#000000;">
          <td style=" border-width:2px; border-style:solid; border-color:#000000;"><input type="Radio"
name="Idnumber" value="<?php echo $row[8]?>" ></td>
          <td style=" border-width:2px; border-style:solid; border-color:#000000;"><?=$row[2]?></td>
          <td style=" border-width:2px; border-style:solid; border-color:#000000;"><?=$row[3]?></td>
          <td style=" border-width:2px; border-style:solid; border-color:#000000;"><?=$row[4]?></td>
          <td style=" border-width:2px; border-style:solid;
border-color:#000000;"><?=$row[5]."-". $row[6]."-". $row[7]?></td>
        </tr>
      <?php
    }
  ?>
</table>
  <input style="font-size: 20px; margin-top: 30px; margin-left: 40%;" type="Submit" value="修改">
  <input style="font-size: 20px; margin-top: 30px; margin-left: 60px;" type="button" value="取消"
"onclick=location.replace('一般搜尋.php')>
</form>
</div>
<!--頁尾區-->
<div id="FOOTER">
  <h2>資料主要來源</h2>
  <pre style="font-size:15px">
  <p style="color:#FFF;padding-left: 20%">Pchome
露天拍賣
                                yahoo 拍賣

```

```

<a style="" href="http://shopping.pchome.com.tw/"></a>
<a
href="https://tw.bid.yahoo.com/"></a>
<a href="http://www.ruten.com.tw/"></a>
</pre>
</div>

</body>
</html>

```

(用表格方式程限使用者現有需求)

E.7.現有需求修改

```
<!DOCTYPE html>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>
<link href="layout.css" rel="stylesheet" type="text/css">
</head>

<body>
<!--頁首區-->
<div id="HEADER">
    <h1>OO 比價網</h1>
</div>

<!--選單區-->
<div id="NAV">

<!--內容區-->
<div id="CONTENT">
    <pre style="font-size:24px">

        <form action="dataupdatefinish.php" method="post" >
        <input type="hidden" name="Idnumber" value="<?=$_POST['Idnumber']?>">
            <a style="font-size:40px">修改後的訂閱資料</a>

            價格區間：<input style="height: 24PX;" type="text" name="價格下限" size="15"> ~
            <input style="height: 24PX" type="text" name="價格上限" size="15">

            結束日期：</datalist> <input type="date" list="vacation-days" name="結束日期">

            <datalist id="vacation-days">

            <option label="Thanksgiving">2017-10-13</option>
```

```
<option label="Christmas Day">2017-12-25</option>
```

```
</datalist>
```

```
                <button style="height: 40px; width:60px ;font-size:
20px; " type="Submit">確認</button>      <input type="button" style="height: 40px; width:60px ;font-size:
20px; " value="取消" onclick="location.replace('data.php')">
```

```
    </form>
```

```
</pre>
```

```
</div>
```

```
<!--頁尾區-->
```

```
<div id="FOOTER">
```

```
    <h2>資料主要來源</h2>
```

```
    <pre style="font-size:15px">
```

```
    <p style="color:#FFF;padding-left: 20%">Pchome                                yahoo 拍賣
    露天拍賣</p>
```

```
                                <a style="" href="http://shopping.pchome.com.tw/"></a>                                <a
```

```
href="https://tw.bid.yahoo.com/"></a>
```

```
<a href="http://www.ruten.com.tw/"></a>
```

```
</pre>
```

```
</div>
```

```
</body>
```

```
</html>
```

(使用者修改需求頁面)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>無標題文件</title>
```

```

</head>

<body>
<?php
try{
    $db_conn = new PDO('sqlite:info.sqlite');
} catch(PDOException $e){
    echo 'Error';
}
session_start();
$un=$_SESSION['username'];
$idsn = $_POST['Idnumber'];
$dp = $_POST['價格下限'];
$up = $_POST['價格上限'];
$ed = $_POST['結束日期'];

if($dp != null && $up != null && $ed != null){
    $a=explode('-', $ed);
    $sql = "UPDATE Exist_Request_Table set
lowerlimit='$dp',higherlimit='$up',year='$a[0]',month='$a[1]',day='$a[2]' where RequestID=$idsn";
    $db_conn->exec($sql);
    $stmt="CREATE TABLE IF NOT EXISTS Modify(Type TEXT,username TEXT,product TEXT,lowerlimit
TEXT,higherlimit TEXT,year INTEGER,month INTEGER,day INTEGER)";
    $db_conn->exec($stmt);
    $stmt=$db_conn->prepare("SELECT * FROM Exist_Request_Table where RequestID=$idsn");
    $stmt->execute();
    $row=$stmt->fetch(PDO::FETCH_BOTH);
    $stmt = "insert into Modify(Type,username,product,lowerlimit,higherlimit,year,month,day) values
('$row[0]','$row[1]','$row[2]','$row[3]','$row[4]','$row[5]','$row[6]','$row[7]')";
    $db_conn->exec($stmt);

    ?>
    <script >
        alert("修改成功");
        location.replace('data.php')
    </script>
    <?php
}

```

```
else
{
    ?>
    <script>
        alert("確認是否有欄位空白");
        location.replace('dataupdate.php')
    </script>
    <?php
}

?>
</body>
</html>
```

(將使用者修改的需求，更新到資料庫)

E.8.顯示歷史資料

```
<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>專題網站</title>

<link href="layout.css" rel="stylesheet" type="text/css">

</head>

<body>

<!--頁首區-->

<div id="HEADER">

    <h1>OO 比價網</h1>

</div>

<!--選單區-->

<div id="NAV">

<!--內容區-->

<div id="CONTENT">

    </td>

<?php
try{
    $db_conn = new PDO('sqlite:info.sqlite');
}catch(PDOException $e){
    echo 'Error';
}

session_start();
$id=$_SESSION['username'];
$stmt=$db_conn->prepare("SELECT * FROM UserData where username='$id'");
$stmt->execute();
echo "<p style='font-size:50px;padding-left:40%;color:red>". "歷史資料". "<br><br>". "</p>";
echo "<p style='font-size:30px;padding-left:45%>". $_SESSION['username']. "用戶 ". "<br><br>". "</p>";
?>
```

```

<form action="dataupdate.php" method="Post">
<table style=" font-size: 24px; margin-left: 40%; border-width:2px; border-style:solid;
border-color:#000000;">
<tr style=" border-width:2px; border-style:solid; border-color:#000000;">
<th style=" border-width:2px; border-style:solid; border-color:#000000;"></th>
<th style=" border-width:2px; border-style:solid; border-color:#000000;">產品名稱</th>
</tr>
<?php
while($row=$stmt->fetch(PDO::FETCH_BOTH))
{
    $a=explode(',',$row[3]);
}
?>

<?php
for($i=0;$i<count($a);$i++){
?>
<tr style="text-align: center" style=" border-width:2px; border-style:solid; border-color:#000000;">
<td style=" padding-left:15px;padding-right: 15px; font-size:20px;border-width:2px;
border-style:solid; border-color:#000000;"><?php echo $i+1?></td>
<td style=" border-width:2px; border-style:solid; border-color:#000000;"><?php echo $a[$i]?></td>
</tr>
<?php
}
?>

</table>
<input style="font-size: 20px; margin-top: 30px; margin-left:450px;" type="button" value="取消"
onclick=location.replace('一般搜尋.php')>
</form>
</div>
<!--頁尾區-->
<div id="FOOTER">
<h2>資料主要來源</h2>
<pre style="font-size:15px">
<p style="color:#FFF;padding-left: 20%">Pchome
yahoo 拍賣
露天拍賣</p>

<a style="" href="http://shopping.pchome.com.tw/"></a>
<a

```



```
href="https://tw.bid.yahoo.com/"></a>
<a href="http://www.ruten.com.tw/"></a>
</pre>
</div>
```

```
</body>
```

```
</html>
```

(顯示使用者訂閱過的產品名稱)

E.9.登出帳戶

```
<?php
```

```
unset($_SESSION['username']);
```

```
?>
```

```
<script>
```

```
alert("登出成功")
```

```
location.replace("登入.php")
```

```
</script>
```

(登出)

E.10.刪除帳號

```
<?php
```

```
try{
```

```
    $db_conn = new PDO('sqlite:info.sqlite');
```

```
}catch(PDOException $e){
```

```
    echo 'Error';
```

```
}
```

```
session_start();
```

```
$id=$_SESSION['username'];
```

```
$sql="delete from UserData where username='$id'";
```

```
$db_conn->exec($sql);
```

```
?>
```

```
<script>
```

```
alert("帳號成功刪除");
```

```
location.replace('登入.php')
```

```
</script>
```

(將使用者資料從資料庫刪除)