

圖書系統-Python 的應用

作者:馮翊軒

指導老師:王璽崑

壹、緒論

一、研究動機

小時候，每周最期待的事就是假日玩電動，每次玩電動時都會萌生出一種感覺，開始想：「這些電動是怎麼做出來的？」、「一開始沒有程式語言時，電腦是怎麼出現的？」，不知不覺間，我迷上了程式並希望能讓自己有一項專長。做一個簡單的圖書系統，不會超過自己的能耐，也不會超過交件的期限，所以讓自己做一個實用且易操作的系統。

二、研究目的

(一)熟悉 Python 的實作以及在生活上的應用

(二)依照資優班的需求設計圖書借用系統

貳、文獻探討

一、python 介紹

根據維基百科，Python 是 Guido van Rossum Python 開發，他與團隊於 1991 年創建了 python 語言，於 1991/2 月時，發行 0.9.0 版本，並於後續增加功能。例如：1994 年 1 月推出 1.0 版本，這個發行版主要新特徵是包括了函式語言程式設計工具。目前 Python 為 3.9.4 版本。(我寫圖書系統的是 3.8.6 版本)

Python 是一種適合業餘使用者娛樂或興趣的程式語言，它很簡單，但很矛盾的又不怎麼簡單；是容易上手，但天花板很高的語言。他將不同功能融於一體，模板類似 C++，但是又不一樣。例如：用冒號取代大、中、小括號等等。

二、Python 的語法容易出錯的地方

(一)括號

Python 與其它的語言的語法最大的不同就是用冒號取代括弧，如 c++ 的括弧就分大中小三種，容易造成使用者的混淆。python 用冒號取代括號，僅使用小括號用來聲明指令。

(二)縮排

Python 有一個很麻煩的點就是縮排問題，Python 不會自動縮排且縮排錯誤可能導致程式運作錯誤或其他問題，需要使用其他軟體輔助才能減少錯誤的機會。

(三)打錯字

有些字如 if/else/or/foundation……，如果拼錯成其他函式，程式可能無法順利執行並顯示錯誤訊息，而執行錯誤報告中並無法明確顯示錯誤原因，以至於除錯步驟較為繁雜。

參、研究方法與步驟

一、研究方法

以專門為中學生寫的程式語言設計：強化邏輯思考力(2018)為參考書籍並挑選其中部分內容作為圖書借閱系統參考範例，理解書中內容後，選定 VSC(visual studio code)作為編譯器完成書中範例並執行。

二、研究步驟

1. 詢問老師實際的圖書借閱系統需求。
2. 增加擷取電腦時間、檢查書籍位置等功能。
3. 完成離線版。
4. 上傳至網路上並實測。

肆、製作歷程

一、目的討論

剛決定題目為程式語言時，我選擇幫資優班製作一個圖書系統，好和我家中的書籍相符合，也希望製作這次獨立研究能增長我對程式的了解。因此王老師和我一起討論圖書系統必須具備的功能，擬定了設計目標與安排實質的測試。

二、外部支援

去年 10 月、12 月和今年 2 月的時候，王老師幫我請 23 屆的學長來幫我解決問題，並幫我改良程式內容與所使用的應用程式，還能夠從學長和老師身上學到東西，讓我對程式有更進一步的了解。

三、問題討論

下表為在製作的過程中遇到的問題，以及解決方案，以供後人參考。

問題編號	問題說明	原因	解決方案
1.	借書迴圈重複執行無法執行輸入借書數量函式，導致數量錯誤。	Config_parser 函式的縮排沒有對齊	調整縮排範例：
2.	在 python 中執行程式時出現" the	函式庫名稱大小寫錯	函式庫名稱調整

	variable" 111-222" is not increased" 的錯誤訊息。	誤	
3.	讀不出正確的借書和還書數量	變數相同	調整縮排
4.	程式閱讀困難，不捲動視窗無法一次看完單一副程式	程式太過於冗長，縮排太多層，識讀不便。	把主程式拆解成數個副程式
5.	無法讀取電腦時間	找不出原因	由於時間規劃來不及完成，只好放棄本功能。
6.	不想匯入現成的圖書資料	由於無法識讀現成的程式的語言，無法更改內容。	自己嘗試重新寫一個系統。

伍、製作結果

一、作品介紹

這次作品的成果主要分三大功能：新增、查詢與借還書。使用者可由主介面新增書籍及使用者；查詢書籍及使用者，借、還書與離開系統。操作方法也很簡單。例：新增使用者只須先輸入"新增借閱證?"〈1〉，再依指令輸入相對應資料就可以新增。

```

-----MSPS Gifted automatic library system
歡迎來到資優圖書專區
<1>新增借閱證
<2>新增書籍
<3>查詢借閱證
<4>查詢書籍
<5>借書
<6>還書
<7>離開系統
請選擇欲使用功能:
1
請輸入您的學號
888
請輸入您的姓名:
王小明
請輸入您的信箱:
123456789@gmail.com
-----
新增成功，歡迎使用本系統
-----

```

二、作品截圖

<pre>def add_member(name,phone,email): config=setting() config.add_section(phone) config.set(phone,"name",name) config.set(phone,"email",email) f=open("userdata.ini","a") config.write(f)</pre>	<p>這是增加使用者的副程式，一開始先代入 configparser 函式庫，再依指定資料寫入組態設定。</p>
<pre>def search_member(phone): config=setting() config.read("userdata.ini") name=config.get(phone,"name") email=config.get(phone,"email") info={"姓名:":name,"信箱:":email}</pre>	<p>這是查詢使用者的副程式，先設定 config 函式，然後讀取資料並顯示。</p>
<pre>def manager(ISBN,num,rid,pattern): config_b=setting() config_b.read("book.ini") config_r=setting() config_r.read("records.ini") storage=int(config_b.get(ISBN,"storage")) if pattern == "borrow": left=storage-num if storage>0 and left>=0: x=search_section(rid,"records.ini") if x == "exist": amount=int(config_r.get(rid,"amount")) print(amount) amount=str(amount+num) config_r.set(rid,"amount",amount) else: config_r.add_section(rid) config_r.set(rid,"amount",str(num)) f1=open("records.ini","w") config_r.write(f1)</pre>	<p>設定一個借還書的副程式，並設定變數和寫入的位置。</p> <p>設定功能等於借書，剩下的書籍如果大於零，書籍總和也大於零，那麼先看紀錄中有沒有書被借過，再使用函式檢查借閱的書籍是否和先前輸入的資料相互應對，否則直接寫入資料。</p>

陸、心得與建議

先來分享我做完這份報告的經驗好了，我當時做完時覺得「我好生佩服自己！」今天能走到這一步，全是因為我的興趣吧！在這裡給學弟妹一個非～常重要的建議：沒有很喜歡或堅持，或耐力定力不夠，千萬不要做這份題目。很多人一定會問；為什麼？因為這份實在太吃你的耐力和消耗你的視力，所以我今天寧願選一份實驗製作或統計資料的題目，也不要做這份。我在這份題目中，從學長和老師身上學到東西，讓我對程式有更進一步的了解。

柒、資料來源

李家同(2018)。專門為中學生寫的程式語言設計：強化邏輯思考力。
維基百科-Python。(<https://zh.wikipedia.org/wiki/Python>)。
Python 官網。(<https://www.python.org>)。