

系級： 四資工一 學號： B10715029 姓名： 陳彥瑋

期末 Python 程式：虛擬貨幣資產管理

學習重點：網頁資料爬取及檔案資料存取

一、設計此程式的動機

在前幾年的時候，加密貨幣非常的火熱，許多的加密貨幣如雨中春筍的產生，但市面上也少有以加密貨幣為主的記賬程式，於是我就想實際做一個類似的程式出來。

二、程式設計目標

撰寫 Python 程式，模擬一個虛擬貨幣的資產管理系統。

此管理系統共有 5 個功能，分別是：購入貨幣、售出貨幣、查詢貨幣價格、查詢交易紀錄、查詢盈餘。

首先先從查詢貨幣價格開始，這邊使用的是鉅亨網的虛擬貨幣專區，網址為「<https://crypto.cnyes.com/BTC/24h>」，其網站如圖所示。



這個網站上可以提供多個虛擬貨幣的價格查詢，只需更改網址列的貨幣代碼，便可以查到不同的虛擬貨幣價格，以下為以太幣的價格網頁(網址為 <https://crypto.cnyes.com/ETH/24h>)。



執行查詢的時候，會抓取左上角的價格標示，如果成功抓取，則回傳查詢的價格，否則顯示「連線異常，無法取得貨幣價格」的提示字樣。

購入及售出貨幣的部分則是詢問使用者欲購入/售出的貨幣，再抓取貨幣的現時價格，並詢問是否繼續交易，如果繼續，則詢問購入/售出的數量，否則結束交易。當使用者輸入數量之後，程式則更新資產資料並在將這次交易增添在交易紀錄中。

查詢交易紀錄則是查詢使用者的交易紀錄，交易記錄格式類似下圖：

	asset.txt	record.txt
1	2019-01-06 22:19:18	buy BTC 10000000.0 at \$3871.93
2	2019-01-06 22:19:33	buy ETH 10000000.0 at \$153.93
3	2019-01-06 22:20:20	sale BTC 10000000.0 at \$3871.09
4	2019-01-06 22:20:34	sale ETH 10000000.0 at \$153.82
5	2019-01-06 22:21:05	buy BTC 10000.0 at \$3871.93
6	2019-01-06 22:22:04	buy BTC 20000.0 at \$3870.61
7	2019-01-06 22:22:38	buy BTC 30000.0 at \$3870.61
8	2019-01-06 22:24:38	sale BTC 60000.0 at \$3875.04
9	2019-01-06 22:26:45	buy BTC 0.1 at \$3875.04
10	2019-01-06 22:26:59	buy BTC 100000.0 at \$3876.17
11	2019-01-06 22:27:32	sale BTC 0.1 at \$3876.17
12	2019-01-06 22:28:06	sale BTC 100000.0 at \$3878.55
13		

查詢盈餘部分則是查詢全部交易的利潤。

三、參考程式碼(共 205 行，141 行之後為主要程式)

列數	程式碼
1	# -*- coding: utf-8 -*-
2	import requests
3	from bs4 import BeautifulSoup
4	import time
5	
6	def testfile():
7	try:
8	asset=open(".\\asset.txt","r+",encoding="utf-8")
9	except FileNotFoundError:
10	print("建立資產表中...")
11	asset=open(".\\asset.txt","w")
12	for i in coins:
13	asset.writelines(i+": 0\n")
14	asset.writelines("balance: 0")
15	asset.close()
16	
17	def prompt():
18	print("輸入 1 購入貨幣")
19	print("輸入 2 售出貨幣")

```

20     print("輸入 3 查詢貨幣價格")
21     print("輸入 4 查詢交易紀錄")
22     print("輸入 5 查詢盈餘")
23     print("輸入 0 結束程式(記錄將保存)")
24
25 def coincode():
26     while(True):
27         print("目前支援的貨幣: ",end="")
28         for i in coins:
29             print(i,end=" ")
30         print()
31         code=input("輸入貨幣代碼(大小寫有別): ")
32         if(code in coins):
33             break
34         else:
35             print("目前不支援此貨幣，請重新輸入\n")
36     return code
37
38 def getamount():
39     while(True):
40         try:
41             amount=float(input("輸入數量: "))
42             if(amount>0):
43                 break
44             else:
45                 print("請輸入大於 0 的數量")
46         except ValueError:
47             print("請輸入數字")
48     return amount
49
50 class coin:
51     def buy(code,amount,value):
52         global asset
53         global records
54         try:
55             asset=open(".\\asset.txt","r",encoding="utf-8")
56             assetlist=asset.readlines()
57             asset.close()
58
59             asset=open(".\\asset.txt","w",encoding="utf-8")
60             index=coins.index(code)
61             add=float(assetlist[index].strip().replace(code+":",
62 ", ""))+amount
63             assetlist[index]=code+": "+str(add)+"\n"

```

63	balance=float(assetlist[-1].strip().replace("balance: ",""))- amount*value
64	assetlist[-1]="balance: "+str(balance)
65	asset.seek(0,0)
66	for i in assetlist:
67	asset.writelines(i)
68	
69	records=open("..\record.txt","a+",encoding="utf-8")
70	nowtime=time.strftime("%Y-%m-%d %H:%M:%S")
71	records.writelines(nowtime+" buy "+code+" "+str(amount)+" at \$" +str(value)+"\n")
72	print("交易完成\n")
73	asset.close()
74	records.close()
75	except FileNotFoundError:
76	print("檔案消失，請重新啟動程式\n")
77	def sale(code,amount,value):
78	global asset
79	global records
80	try:
81	asset=open("..\asset.txt","r+",encoding="utf-8")
82	assetlist=asset.readlines()
83	asset.close()
84	
85	asset=open("..\asset.txt","w",encoding="utf-8")
86	index=coins.index(code)
87	minus=float(assetlist[index].strip().replace(code+": ",""))- amount
88	if(minus<0):
89	print("數量不足，您只有%f 個 %s"%(minus+amount, code))
90	print("交易失敗\n")
91	else:
92	assetlist[index]=code+": "+str(minus)+"\n"
93	balance=float(assetlist[-1].strip().replace("balance: ", ""))+amount*value
94	assetlist[-1]="balance: "+str(balance)
95	asset.seek(0,0)
96	for i in assetlist:
97	asset.writelines(i)
98	records=open("..\record.txt","a+",encoding="utf-8")
99	nowtime=time.strftime("%Y-%m-%d %H:%M:%S")
100	records.writelines(nowtime+" sale "+code+" "+str(amount)+" at \$" +str(value)+"\n")

```

101         print("交易完成\n")
102         records.close()
103         asset.close()
104     except FileNotFoundError:
105         print("檔案消失，請重新啟動程式\n")
106     def getvalue(code):
107         url="https://crypto.cnyes.com/"+code+"/24h"
108         try:
109             res=requests.get(url)
110         except:
111             return -1.0
112         bs=BeautifulSoup(res.content,"html.parser")
113         lastprice=bs.find("div","last-price")
114         price=lastprice.find("span","big-
num").text+lastprice.find("span","small-num").text
115         price=price.replace(",","")
116         return float(price)
117
118     class account:
119         def record():
120             try:
121                 global records
122                 records=open("..\record.txt","r",encoding="utf-8")
123                 recordlist=records.readlines()
124                 if(len(recordlist)==0):
125                     print("無紀錄\n")
126                 else:
127                     print("紀錄如下：")
128                     for i in recordlist:
129                         print(i.strip())
130                     print()
131             except FileNotFoundError:
132                 print("無紀錄\n")
133             records.close()
134         def getmoney():
135             global asset
136             asset=open("..\asset.txt","r",encoding="utf-8")
137             assetlist=asset.readlines()
138             print("目前盈餘為： "+assetlist[-1].replace("balance: ","").strip()+
元\n")
139             asset.close()
140
141     coins=["BTC", "ETH", "XRP", "BCH", "EOS", "XLM", "LTC", "USDT", "BSV", "TRX"]

```

```

142 records=open(".\\record.txt","a+",encoding="utf-8")
143 records.close()
144 testfile()
145
146 while(True):
147     while(True):
148         prompt()
149         try:
150             choice=int(input("輸入: "))
151             break
152         except ValueError:
153             print("請輸入數字\n")
154
155     if(choice==0):
156         break
157     elif(choice==1):
158         code=coincode()
159
160         value=coin.getvalue(code)
161         if(value<0):
162             print("連線異常，無法取得貨幣價格")
163             buy="N"
164         else:
165             print("目前%s 的價格為%.3f 元"%(code,value))
166             buy=input("確認要購買(Y/N) ? ")
167             if(buy.upper()=="Y"):
168                 amount=getamount()
169                 coin.buy(code,amount,value)
170             else:
171                 print("交易取消\n")
172
173     elif(choice==2):
174         code=coincode()
175         value=coin.getvalue(code)
176         if(value<0):
177             print("連線異常，無法取得貨幣價格")
178             sale="N"
179         else:
180             print("目前%s 的價格為%.3f 元"%(code,value))
181             sale=input("確認要販賣(Y/N) ? ")
182             if(sale.upper()=="Y"):
183                 amount=getamount()
184                 coin.sale(code,amount,value)
185             else:

```

```

186             print("交易取消\n")
187
188         elif(choice==3):
189             code=coincode()
190             value=coin.getvalue(code)
191             if(value<0):
192                 print("連線異常，無法取得貨幣價格\n")
193             else:
194                 print("%s 的價格為%.3f 元\n"%(code,value))
195
196         elif(choice==4):
197             account.record()
198
199         elif(choice==5):
200             account.getmoney()
201
202         else:
203             print("輸入代碼錯誤\n")
204
205     print("程式即將關閉，歡迎再度使用")

```

四、程式碼解說

第 2~4 行：匯入各項所需的套件，包括：requests 套件、BeautifulSoup 套件、time 套件。

第 6~15 行：測試是否有 asset.txt，若無，則新建此資料表，並填入格式。

第 17~23 行：提示使用者輸入服務代碼的文字。

第 25~36 行：提供使用者輸入貨幣代碼的函式。

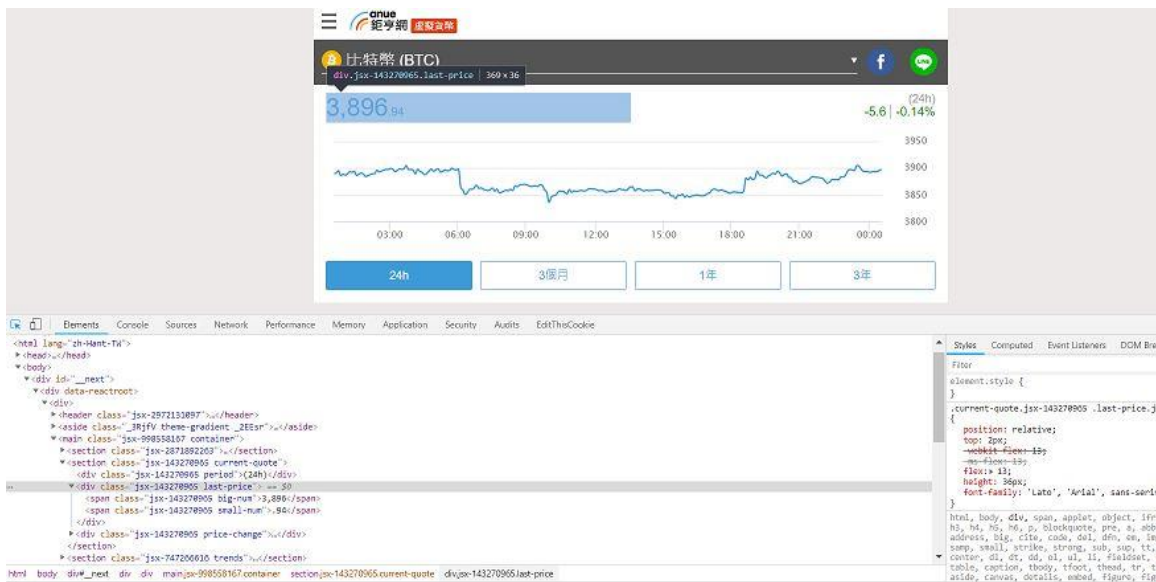
第 38~48 行：提供使用者輸入購入/售出數量的函式。

第 50~116 行：貨幣資料相關的 class。

其中的第 51~76 行：購買貨幣的函式，先把資產資料讀出並存入 assetlist 中(55~57)，再將欲修改的資料索引值讀出(60)，並把購入的數量加總並存回 assetlist 中(61,62)，再將餘額更新(63,64)，最後再將資料重新寫入資產表之中(65~67)。資產表更新完之後再將交易紀錄在交易紀錄表之中(69~71)，並在最後提示使用者交易完成(72)。

其中的第 77~105 行：賣出貨幣的函式，與購買的函式類似，唯獨其中的第 88 行會檢查是否有足夠的貨幣可以賣出，如果足夠才會繼續交易。

其中的第 106~116 行：獲得貨幣價格的函式，首先發出請求獲得網頁的 html，如果連線異常則回傳-1，否則繼續使用 BeautifulSoup 分析 html，其中的價格在 <div class="last-price"> 中的 (整數)和 (小數)之中，如圖所示：



第 118~139 行：使用者資料相關的 class。

其中的第 119~133 行：讀取交易紀錄的函式。

其中的第 134~139 行：讀取盈餘的函式。

第 141 行：儲存貨幣代碼的陣列

第 142~144 行：檔案的 initialize。

第 147~153 行：提供使用者輸入服務代碼。

第 155~156 行：服務代碼 0，結束程式。

第 157~171 行：服務代碼 1，購買貨幣，首先詢問使用者輸入貨幣代碼(158)，再來提示使用者貨幣價格(160~165)，並詢問使用者是否要購買(166)，如果是，則詢問使用者欲購入的數量(168)並進行購買(169)；如果否，提示交易取消(171)。

第 173~186 行：服務代碼 2，賣出貨幣，與代碼 1 類似。

第 188~194 行：服務代碼 3，查詢貨幣價格，如果回應價格小於 0(連線失敗)，提示使用者(192)，否則輸出價格(194)。

第 196~197 行：服務代碼 4，查詢交易紀錄。

第 199~200 行：服務代碼 5，查詢盈餘。

五、對於大數據與程式設計導論課程的學習心得

對於課程的前半段，也就是大數據理論的部分，我認為老師在這部份教的稍為有些無趣，聽完之後在這部分也沒有特別的感覺，也許是因為我比較不喜歡理論的部分吧，我認為可以再增加更多實際的使用情況，像是預想工作表的討論方式或是大數據分析工具包的使用方法之類的，或者是可以在上課時講一些笑話來提升大家的精神之類的方式吧。

而對於課程的後半段是我比較喜歡的部分，雖然很多的部分我已經都學過了，但有些的部分我都沒有了解過或是使用過，這些部份讓我學到了不少的獲益良多，但後面的部份上的速度稍微有些的快，希望老師能夠改善這個問題。