

# 高一專題統整

郭勝威

2022 年 7 月 27 日

# 目錄

<b>1</b>	<b>序言</b>	<b>1</b>
<b>2</b>	<b>爬蟲</b>	<b>2</b>
2.1	實作流程 . . . . .	2
2.2	抓取解題網站的資料 . . . . .	3
2.3	自動機器人 . . . . .	6
2.4	密碼破解 . . . . .	9
2.5	校內廣播 . . . . .	11
2.6	排名爬取 . . . . .	14
2.7	學測單字爬取 (一整年的最終成果) . . . . .	17
2.7.1	序言 . . . . .	17
2.7.2	實作動機 . . . . .	17
2.7.3	遇到的困難和解決方式 . . . . .	17
2.7.4	學到的事情 . . . . .	19
2.7.5	可以改進的方向 . . . . .	19
<b>3</b>	<b>遊戲</b>	<b>20</b>
3.1	貪吃蛇 (python) . . . . .	20
3.2	貪吃蛇 (c++) . . . . .	21
<b>4</b>	<b>網頁製作</b>	<b>22</b>
4.1	乘法練習網站 . . . . .	22
4.2	決策器製作 . . . . .	23
4.3	數學公式製作 . . . . .	24

<b>5</b>	<b>應用程式製作</b>	<b>25</b>
5.1	搜尋單字 (半完成) . . . . .	25
<b>6</b>	<b>manim 動畫製作</b>	<b>26</b>
6.1	餘弦證明動畫 . . . . .	26
6.2	圓周弦與弧的性質 . . . . .	27
<b>7</b>	<b>人工智慧</b>	<b>28</b>
7.1	MINST手寫辨識 . . . . .	28
7.2	個人思考 . . . . .	29

# 1 序言

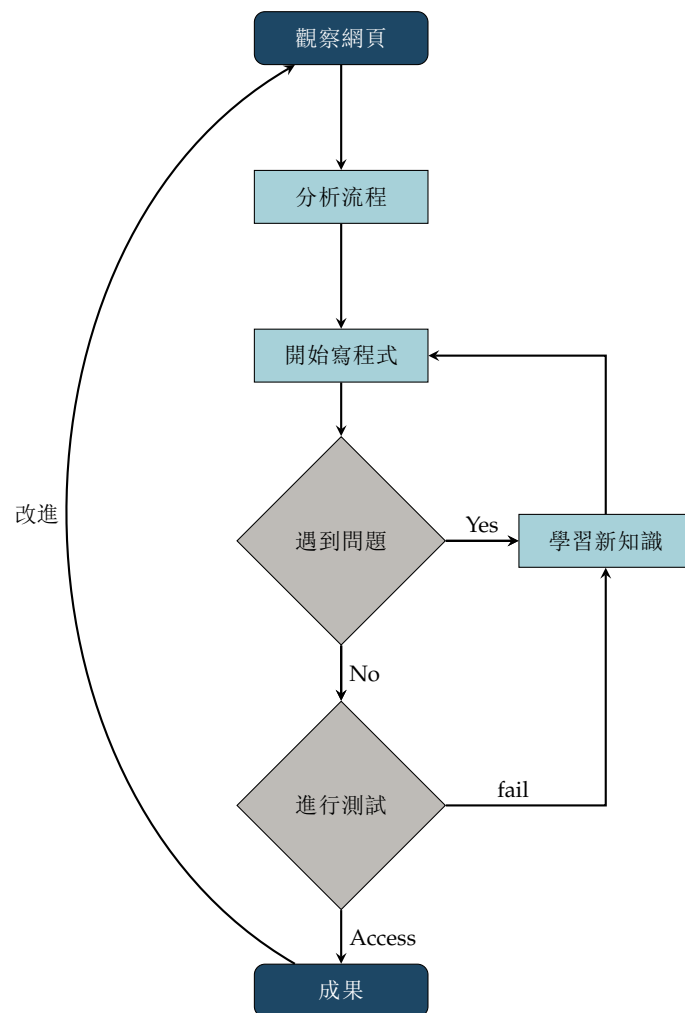
高一轉瞬即逝，當我回顧一年來的收穫，不知不覺中就做了許多小專案，每一個專案都是心血來潮才開始的，實作時，總會遇到許多問題，不過我會投入許多的時間去思考不同的解決方案，抑或從網路上尋找解答，我相信這一點一滴了累積，即使不起眼，但仍代表我的進步

我把這一年所做的專案分成六個部分，大部分都是由`python`實作的，網頁則是用`html`, `css`, `javascript`實作，爬蟲的大部分都是為了讓自己可以輕鬆一點，其他的地方則是可以因為自己的好奇心而去研究的，每一個專案都至少花了**10hr**以上，不過卻都沒有很完整，都還有許多空間可以改進，不過許多都因為自己的能力未到而尚須時間，就先把這些託付給未來的我了。

## 2 爬蟲

### 2.1 實作流程

實作流程都相差無幾，可以使用下面這張圖來代表



## 2.2 抓取解題網站的資料

### 實作動機

因為我有擔任程式設計的教學，所以常常需要出題，而解題網站 [CSES](#) 有許多很好的題目，所以我有時會把裡面的題目翻譯然後搬到 [DDJ](#) 上，但是 [CSES](#) 裡面的測資是 .txt 的，還要一個一個下載下來，而 [DDJ](#) 要求的是 .in, .out 檔，所以我決定寫個程式把檔案下載下來然後改成 .in, .out 檔

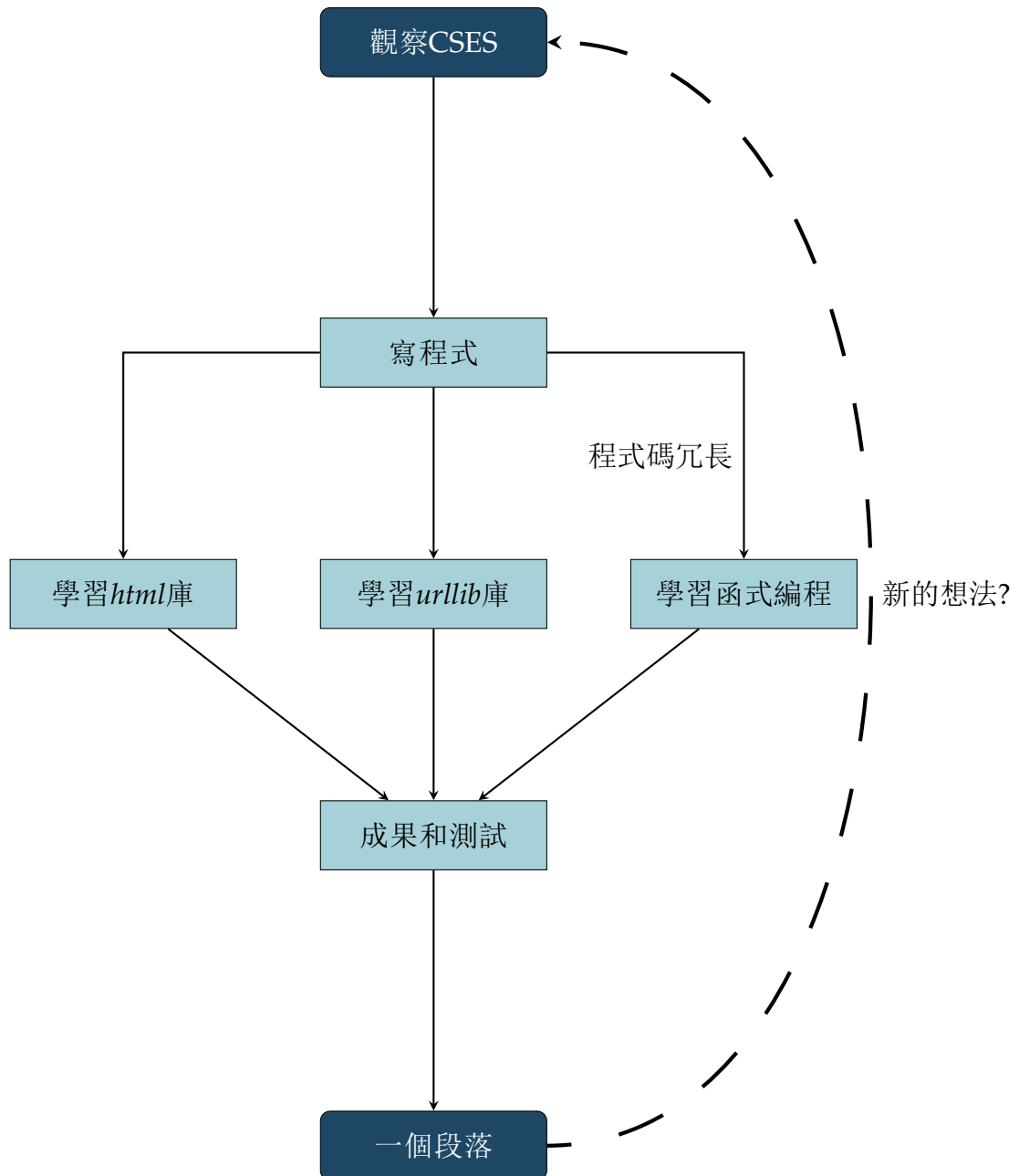
### 遇到的困難和解決方式

一開始的時候，因為才剛開始寫 *python*，所以程式碼十分冗長，我自己也看不太懂，所以之後就開始嘗試把許多東西變成函式，意外的發現挺方便的。

### 學到的事情

1. 開始嘗試用 *python* 拼出想要的功能
2. 學會使用 *urllib* 庫，
3. 學會使用 *with...open...* 來開啟和寫入檔案
4. 如何觀察 *html* 文檔

過程



## 改進空間

1. 因為我實作登陸是用`cookie`，每一次都需要複製網站上的`cookie`，所以可以使用`session`之類的庫把他改成輸入帳號跟密碼就可以登錄。
2. 通用的改進: 我想要把他弄出圖形化介面

## 心得

這一次實作的功能十分簡單，只有使用`python`爬取檔案，然後寫入而已。不過仍花了我許多時間，



## 2.3 自動機器人

### 實作動機

之前因為疫情所以開始了線上上課，但是每一次上課的時候都須找尋找 google meet 的連結，十分的不方便，因此我就萌生了一個想法: 可不可以在每節下課的時候發出下一節上課的meet，讓他自動把連結送到聊天室，這樣除了我自己可以方便的進教室外，其他同學也一樣可以方便的使用。

### 遇到的困難

message api:

1. 實作message api的時候發現要伺服器  $\implies$  使用[heroku](#)
2. 需要一個群組才能開始傳送訊息  $\implies$  這時候我決定換一個做法

selenium庫:

1. 網頁十分複雜，不知道哪一個地方代表按鈕  $\implies$  研究了很久之後發現他們是直接

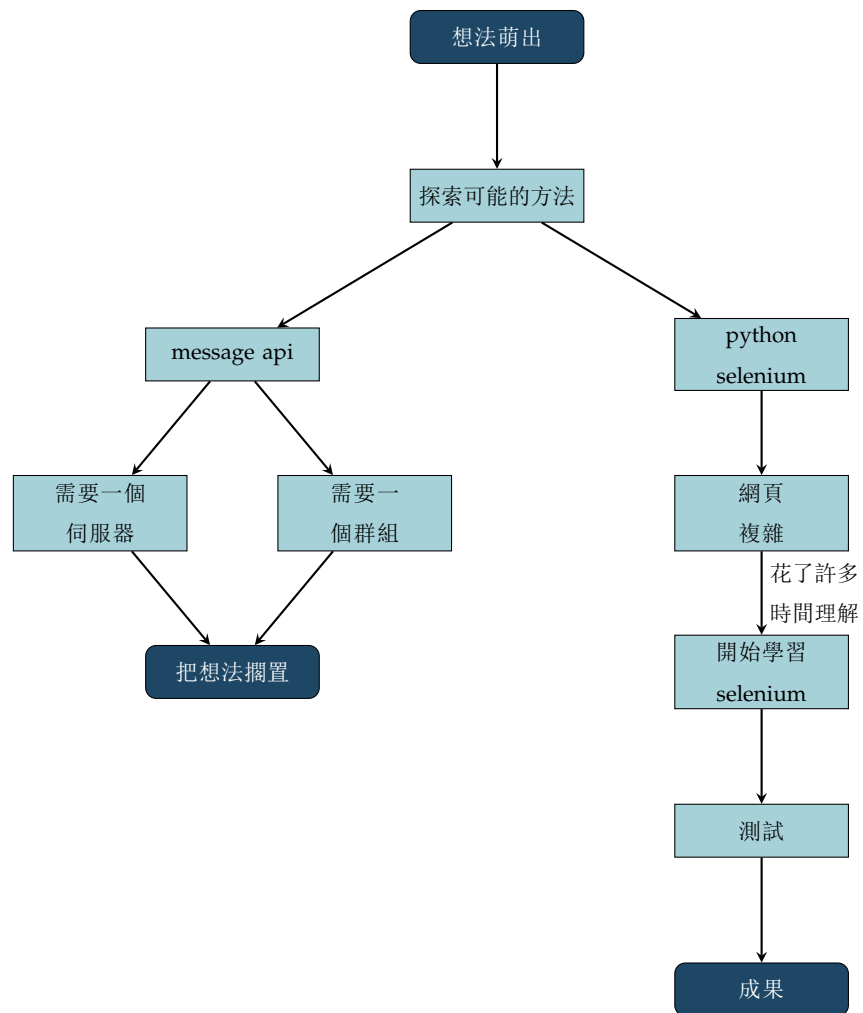
div

來表示按鈕

### 學到的事情

1. 學會了如何分析更複雜的網頁
2. 學會了使用selenium庫

## 過程



## 心得

這個專案是我第一次研究大型公司的網頁，我發現他們的網頁都十分的複雜，無論是元素的命名，還是他作用的方式，都大幅的讓我了解到自己的不足，對網頁的了解還只是皮毛，更加深了我對網頁的興趣。

## 改進

1. 現在的程式仍需要每天清自去點開來執行，有些許的不方便
2. 在修改課程的時候有寫麻煩，在使用的最後幾天因為調課頻繁的原因出了許多差錯，應該可以在多做一個程式來專門修改課程
3. 圖形化介面

## 2.4 密碼破解

### 實作動機

在學了許多的知識之後，有時候總會冒出一些奇妙的想法，在學校的時後，我發現有些幹部他們登陸時候的密碼十分的簡單，許多都只有四位數，於是我就萌生了一個想法，是不是可以透過程式來破解密碼，稍微計算了一下，如果每秒鐘請求十次的話，對於四位數的密碼共需要請求約1000秒，也就是16分鐘，是在可以接受的範圍，於是我就開始打程式了

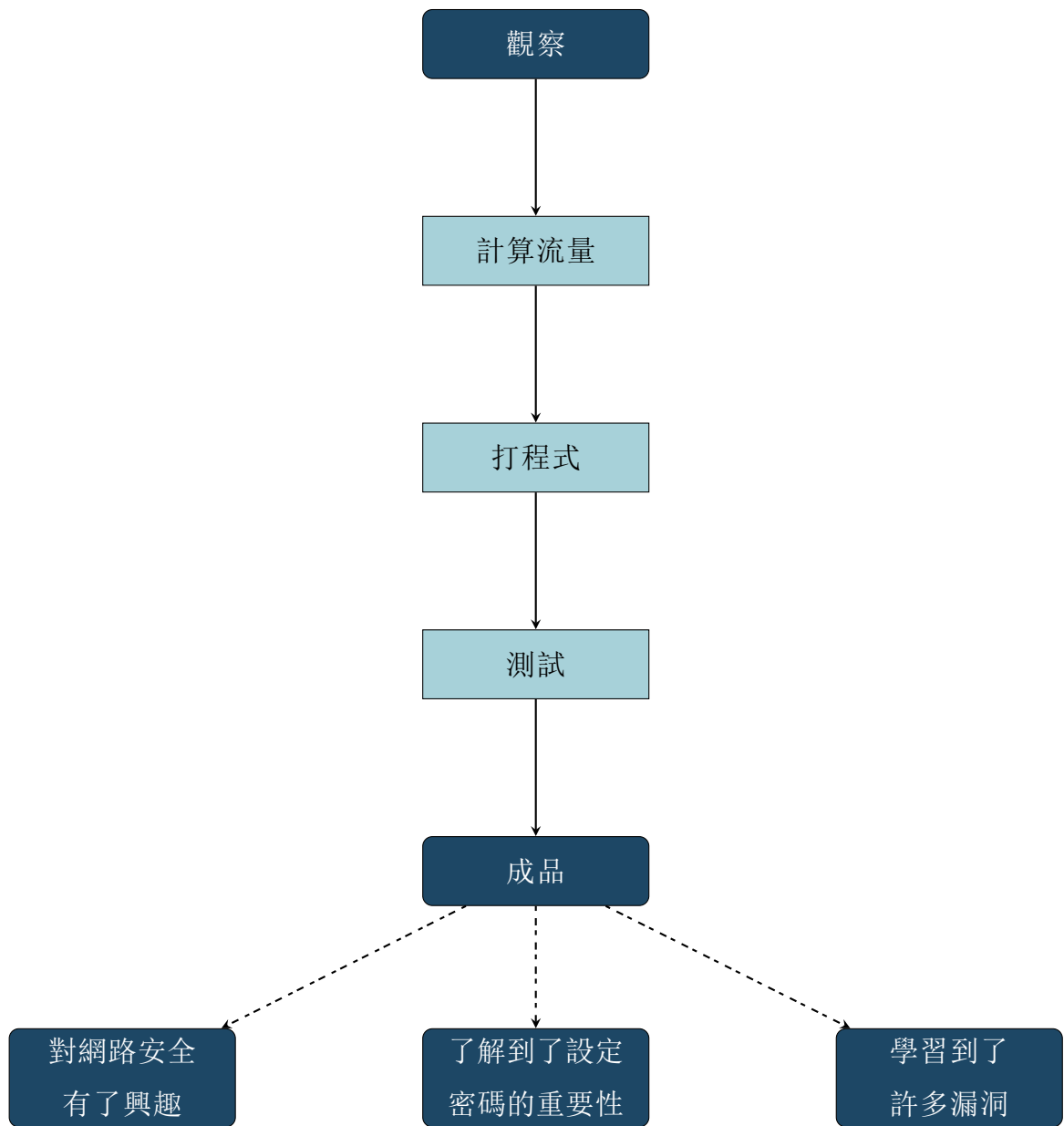
### 遇到的困難和解決方式

這次因為只是簡單的嘗試所以沒有遇到什麼困難，不過為了確保不會讓學校網站當掉所以有壓低了請求速率。

### 學到的事情

在實作過後我學到了密碼一定要好好設，如果只有數字而且只有四位數的話，雖然看起來很難被猜到，不過對於電腦來說只是幾分鐘的事情，同時也因為了這次嘗試，我開始對網路安全有了些許興趣，也有開始去學習一些漏洞向觀的知識(譬如: xss, sql注入..., ddos)

過程



## 2.5 校內廣播

### 實作動機

因為我是資訊股長，所以要負責看廣播，但有時候會漏看許多的廣播，這時候腦袋內又出現了一個想法，是不是可以透過`python`來進行廣播的確認。

### 遇到的困難和解決方式

一開始我一直在煩惱要怎麼樣才可以解決登入的問題，在經過了一段時間的尋找之後，我發現`session`這個函式庫可以很好的解決我的問題。

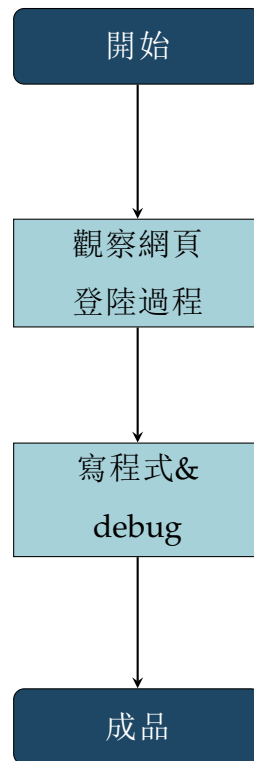
### 學到的事情

1. 學會使用`session`庫
2. 透過觀察網頁了解到學校登陸功能的實作  $\implies$  透過承載資訊直接傳給網頁，

### 意外的發現

在實作的過程中，使用控制台觀察資料傳遞的過程的時候，發現學校網頁在傳遞的過程中沒有把承載的資料進行收回，雖然不知道會不會造成安全性的問題，不過也是有趣的收穫。

## 過程



### 改進的方向

1. 希望可以在把資訊爬取下來之後把得到的資訊整理過後再重新放到網頁上面讓班級同學查看
2. 透過增加一些程式讓使用程式的時候可以更加方便的更改想要資訊，目前是希望透過一個`setting.json`來進行實作，
3. 圖形化介面



## 2.6 排名爬取

### 實作動機

由於我是進階助教，所以每次考試之後都需要統計他們的分數，每一次都要逐一的去登記，由於我覺得他十分麻煩，不如直接寫程式把他們的成績爬取下來。

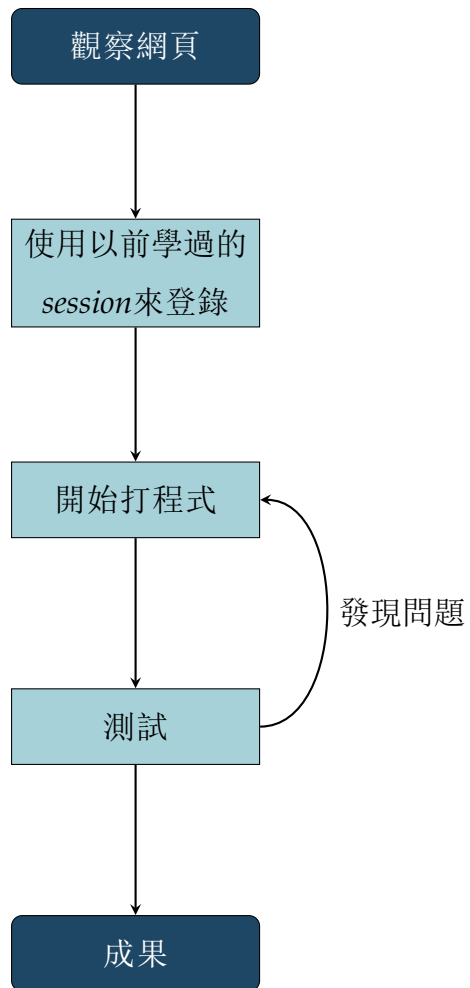
### 遇到的困難和解決方式

因為登陸的問題在前面已經解決了，所以這次沒有發生什麼特別的困難，只是對於改進沒有什麼想法

### 學到的事情

1. 對於python的實作更加的了解

## 過程



### 可改進的方向

1. 這次實作的只有獲得總分，因為如果要獲取每一題的分數的話難度會增加很多，因為比賽時如果拿到滿分的話，顯示出來的是圖片，所以要做的事情就需要多許多。
2. 圖形化介面
3. 改善操作使用，因為現在需要把網址一個一個貼上才可以，希望以後可以直接使用比賽的命稱來直接獲得分數

## 2.7 學測單字爬取 (一整年的最終成果)

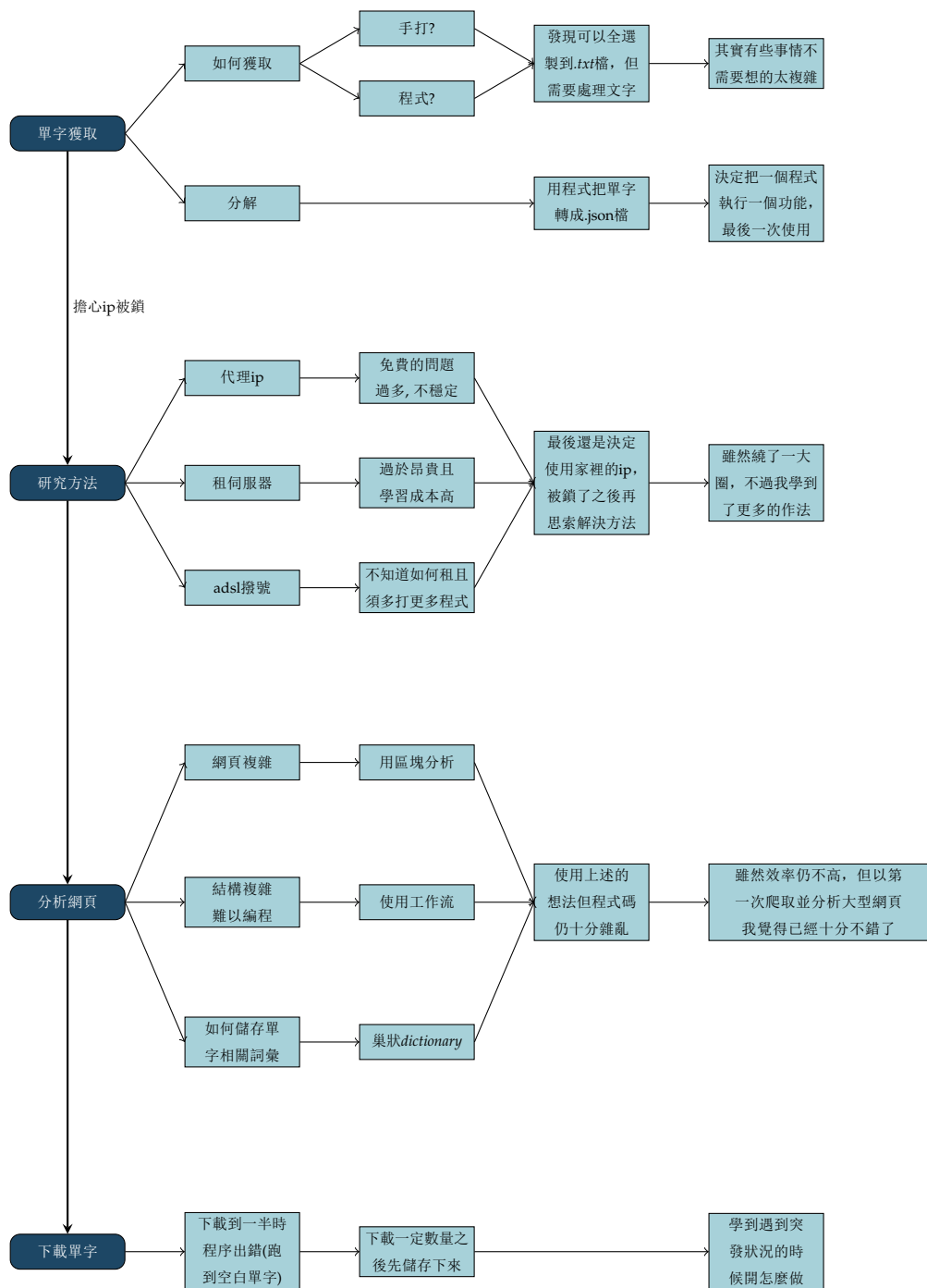
### 2.7.1 序言

這一個專案是我目前做過最大的一個專案，程式碼超過了500行，也是我第一次把我學到的幾乎所有東西都用上了，也學到了更多的新事物，雖然這個專案可能沒有如此的龐大，但對於我在探索程式的道路上還是有了深遠的影響。

### 2.7.2 實作動機

一切的起因是因為同學一句：「學測英文是不是沒有範圍阿?」，這引起了我的好奇心，所以我就很認真的去查了學測的範圍，發現其實每一科都有很明確的範圍的，我也發現到英文科其實有特別明定7000單的範圍，所以我就萌生了想法，我是不是可以先把7000單下載下來，然後透過劍橋字典，把每個單字都找出他的用法、片語、詞性、同義字....，存下來，之後就看要做網頁、應用程式、單純的執行檔，都可以，於是我就開始了盡情揮灑創意的實作了。

### 2.7.3 遇到的困難和解決方式



#### 2.7.4 學到的事情

1. 透過把程式碼變成許多函式+檔案來讓自己可以快速的找到使用的函式
2. 學到了許多關於ip的事情
3. 對於中小型項目的檔案分布跟程式碼實作方式有了基本的了解
4. 開始嘗試使用許多小小的區塊，透過工作流的方式，把小小的功能集結成一個完整的項目。

#### 2.7.5 可以改進的方向

1. 因為對於一些名詞不太熟悉的緣故，命名命的不太好，只後應該可以把函式命名簡化配合工作流(譬如: `work_flow_1`, `work_flow_2`, `work_flow_3`)
2. 之後可以使用應用程式、網頁、執行檔來把單字使用上去。
3. 對於檔案名稱和資料夾的名稱可以再加強，
4. 在實作專案之前，可以先把整個檔案的執行過程使用流程圖表示出來，然後把檔案的名稱分配好，讓實作可以變得更加的容易。讓整個檔案可以變得更加的易讀

## 3 遊戲

### 3.1 貪吃蛇 (python)

## 3.2 貪吃蛇 (c++)



## 4 網頁製作

### 4.1 乘法練習網站

## 4.2 決策器製作

### 4.3 數學公式製作

## 5 應用程式製作

### 5.1 搜尋單字 (半完成)

## 6 manim 動畫製作

### 6.1 餘弦證明動畫

## 6.2 圓周弦與弧的性質

## 7 人工智慧

### 7.1 MINST手寫辨識

## 7.2 個人思考