**報名信**

1. **自我介紹**

　　 我叫做蘇裕翔，24歲，畢業於輔仁大學物理學系，在別人眼中我是一個很難被說服的人，若自己沒用盡方式去嘗試，不會輕易相信別人的方法比較好。個性上也比較容易鑽牛角尖，處理問題前會習慣先以想到的第一種解決方式來解決，即便大家會認為第二種方法可以比較省時間，我還是希望用我自己的方式解決問題。因為出生背景比別人差，我從小就很擔心別人看不起自己，不論是親戚朋友或同學，我希望自己能從事大家認為高尚且能轉更多錢的工作。

我之前有在宏碁擔任輔導工程師，聽起來好像很厲害是個工程師的身份，但是其實我什麼程式都不會。我以為這份職位能讓我學習到程式的部分，但後來發現這份職位只是了解客戶端的問題，再把程式問題轉給工程師處理。這份職位其實跟客服很像。因為我完全看不懂系統，所以全部的問題都沒辦法自己解決，只能PASS給工程師。由於其他輔導工程師都具有資訊背景，因此他們在面對程式問題，可以自行替客戶解決，但是我沒有程式基礎，總是需要其他人協助處理相關問題，導致自己有被其他輔導排擠的情況。我也因而感到很痛苦，痛苦的來源不是因為被同事排擠，而是自己什麼都不會，工作了半年的時間，我頂多比較能聽的懂客戶在反映什麼問題以及如何跟工程師溝通。

但我認知到這不是我想要做的事，想真的成為一位工程師，我必須從零開始學習，從頭把基礎學好，於是我毅然決然辭去了工作。因為不知道該從哪裡開始著手，所以開始在網路上搜尋資料，在 ＰＴＴ這篇文章「從教學者的角度看無經驗轉職以及課程選擇」<https://www.ptt.cc/bbs/Soft_Job/M.1546655647.A.807.html>，看到線上課程跟實體培訓班，於是我取找到了作者開的線上課程，在 Lidemy 開始學習城市基礎。

1. **報名這個計畫的理由**

報名這個計畫的動機是，前面自我介紹提到的 PTT 文章，文章中提到有些學生會切版會寫 JavaScript 會用 jQuery 可是寫不出九九乘法表。告訴我們要應該掌握哪些核心能力，看到問題以後如何拆解、解完如何組裝、該從哪裡去測試自己寫得程式、判斷哪邊有問題，這些都是最基本的能力。由於前一份工作自己就陷入了缺乏專業能力，所以對這段敘述特別有共鳴。再加上我自今年3 月份開始有親自學習過老師的課程，發現老師很用心站在新手的角度思考。會利用生動有趣的故事來幫助我們理解核心概念。例如：傳紙條故事、拉麵販賣機、一日圖書管管理員······等。

另外，作業像在闖關玩遊戲！讓我瞭解某個概念實際上能夠這樣應用。比如說朋友找我寫一段可以得到遊戲直播觀看排行的程式，我就要去直播網站上找到網站提供的 API 文件，知道在現實生活中如何去應用，瞬間了解這個概念對於開發來說有多重要，重要性在哪，為什麼要會。

在練習串接API 時，遇到了同步、異步的相關問題，查詢許多的說明，比較所有網友提供的資料，還是只看得懂 Huli 老師做的說明。老師利用排隊點餐的故事說明堵塞的原因以及背後的原理。再加上拿出 node 官方的文件告訴我們正確的訊息，利用佐證資料的方式會讓我們覺得更具說服力。

從今年 3 月開始到現在，都在 Lidemy 的網站上學習，在過程中發覺自己的邏輯與老師很類似，所以吸收的很良好，於是打算下定決心跟著老師學習，老師的影片跟其他人的差別就在於一刀未剪！即使遇到問題，也是現場示範解決。而且每一堂開放課程都會說明這堂課適合哪些人上課。先前在做 LIOJ 的作業時，因為解法上面遇到困難，在 slack 上提問問題，Huli 老師回答留言板問題的速度很快，我可以馬上了解自己的哪裡出在哪裡。

在上 [alg101] 先別急著寫 leetcode 這門課程時，好幾次寫完 LIOJ 的題目再去看老師的解答時，發現到自己寫出的程式可以與老師的答案幾乎一模一樣，甚至連變數的名稱都設的一樣。讓我覺得非常開心，感覺自己成功運用老師所教育的程式思維，能與老師有相同的思考邏輯，特別有成就感。

我相信跟著 Huli 老師學習，我會成為一位具備專業能力的工程師。會這麼說是因為，老師打破了我對學程式的迷思。一開始我以為程式就是指令，背起來拿來用。像 HTML 的標籤一樣，我要標題就用標題的標籤，要表單就用表單的標籤，覺得自己已經算是會很多了。後來，一堆讓人摸不著頭緒的概念出現。為了解決一個問題，卻出現了千千萬萬新的問題。原本只是想用 node 串接 API 取得前十本書的資料，才知道原來 ajax 非同步會使 log 出來的順序亂掉。要解決非同步的問題，就得去了解 callback function 跟遲延的概念與使用方式。我才領悟到寫出程式碼不能成為工程師，必須要會解決問題的能力。

看了老師提拱的學習課綱後，我的學習目標已經不再是寫出指令碼，而是學習基本功。Huli 老師讓我這個外行人理解了專業人士眼中的基礎是什麼！

為了未來能跟上老師課程的腳步，我現在已經利用第一期實驗計畫的課程影片做預習。即使會看不懂內容，但讓自己提前有概念知道我需要學會什麼。

好不容易看到了老師開放了第四期課程了，於是就趕快來報名了。

但除了觀看影片之外，我想要的是一個能夠發問、並驗證自己的學習狀況的學習環境。我知道報名這門課程，能讓我擁有一個很棒的學習平台，也可以與同學之間彼此交流。

1. **若成功錄取之後，你的時間安排為何**

目前待業中，故每天都有 8 小時可以學習程式，目前計畫上午會先花兩個鐘頭將這週的重點觀念複習一遍，下午利用兩個鐘頭來寫作業或閱讀參考補充資料，中間休息一個鐘頭。再回來用一個鐘頭看其他同學問的問題與參考其他同學寫得作業。晚上再花兩個鐘頭，學新的影片或觀看直播作業檢討影片。

短期計畫，跟著上完六個月的課程後，找尋前端工程師的相關工作。並希望能在2~3年內逐步增進自己後端程式技術，提升自己成為一名全端工程師。就前端的部分，會隨著實驗計畫的學習進度來學習，後端的部分我會先思考要往哪個領域發展再來決定要專修哪一種後端語言。

1. **你未來想做什麼？**

我的第一志願：希望能成為 Huli 老師的助教，能跟隨老師在身邊，協助老師從事教育工作。因為這能讓我持續吸取老師的實驗精神，接受挑戰並學習成長。也許 N 年之後，也可以成為一位優秀的程式老師。

第二志願：希望能先進小公司的前端工程師，因為我覺得小公司能學東西，而大公司能學的是制度。想先進小公司累積自己處理問題的經驗。

夢想工作，希望能成為閒閒的月入百萬工程師，彈性的工時，可以在咖啡廳上班，事情處理完就下班。

1. **平時有寫東西的習慣嗎？**

偶爾會在 FB 或 IG 上PO文分享心情。

1. **Scratch 作品連結 & 做完 Scratch 作品的心得**

連結：https://scratch.mit.edu/projects/392744604

第一次看老師教我們使用 Scratch 的影片，我覺得很興奮，不需要會任何語言，單純用拼圖做出自己的小程式。想說這應該是一件簡單的事情。但實際製作遊戲的時候，卻出現意料之外的錯誤。像要做出走路的效果，測試的時後，角色會先站在原地做出走路的動作，做完後才會真的改變位置。摸索了一陣子才想起老師說過程式是一行一行執行，所以走路的動作執行完後，才會執行改變角色的位置。那就產生新的問題了，如何讓兩段程式可以同步執行？想著想著才找到了解決方式，讓走路的動作跟移動位置分成兩段獨立的程式，再以相同的事件作為監聽的對象，才解決了走路效果的問題。

我覺得 Scratch 真的蘊含許多程式的概念，但又不需要擔心自己完全不會使用指令。難怪老師這麼推薦在學習程式前先學著使用 Scratch。既能實際練習運用程式邏輯，也讓整個學習變得好玩起來。提升自己對程式學習的興趣。

我做出的這份作品，跟自己預期的樣子差很多。我選擇的是逃生主題的小遊戲，會想這樣設計是因為我希望這個遊戲是有教育意義的，能讓玩家了解火場自救的重要觀念。在劇本寫出來後，我開始琢磨著如何用 Scratch 呈現。原先希望會是一個動畫感十足，讓人能身入其境的遊戲，畢竟每一個選擇都會攸關性命，若是在現實生活中遇到或許可以救自己一命，真的製作時才發現有夠難的 ! 不要說動畫感十足了，要讓角色跟背景互動就已經燒掉一半的腦細胞了。當加入的素材越來越多時，bug 也越來越多，例如：我換個背景，該消失的素材卻不會消失，或者明明觸發新的事件了，素材還在執行原本的動作。後來才知道，在每一次新增新的素材時，可以先把預設條件跟結束條件都訂好，能省去後面不少麻煩。該它出現之前把它預設為隱藏，該有事件變更時就設定成結束。素材們就會乖乖的在該出來的時候出來，該躲起來的時候乖乖躲好。

我覺得用 Scratch 做遊戲，就好比在導一齣經典大戲。每個素材都有自己的劇本。要讓整齣戲可以看起來流暢，需要每個素材之間配合，這同時也是最難的部分。還好程式不會脫稿演出，也不會在表演時忘記台詞。雖然最後的成果不是我預期的樣子，不過能在第一次製作就將我需要的功能通通以程式呈現出來，這點讓我蠻有成就感的。

1. **Codewar 帳號 & 解 Codewar 題目的心得**

帳號：krebikshaw

所有題目寫完後，我想分享我認為最有趣的三道題目:

1. Take a Ten Minute Walk

2. A Man and his Umbrellas

3. Check if two words are isomorphic to each other

這三題詳細的解題心得如下:

第一題: Take a Ten Minute Walk

要判斷有一個人往東西南北不同方向走完 10 步後會不會回到原點，我第一個想到的解決方式就是利用兩個 temp 分別記錄東西與南北兩個方向。Temp 先預設為 0，往某個方向就 +1 另一個方向則 -1。這樣一來結束之後我就可以判斷這兩個 temp 是不是依然都為 0。接下來便開始利 pseudo code 來打草稿，利用迴圈來處理每一步要做什麼，設定條件判斷不同方向的處理方式。一行一行的確認是否能得到我想要的答案，最後才把它轉換成程式碼。當我 Attempt 之後發現有出現錯誤。未滿 10 步的情況也要算失敗 ! 但是我沒有把未滿 10 步的情況列入考慮。可見一開始題目的要求我就沒有看仔細了，真的是不應該 ! 下次得多多注意。

第二題: A Man and his Umbrellas

這題真的很有趣，題目給我接下來幾天的天氣預報，我要計算我一共會買幾把傘。我想到設定兩個變數來記錄家裡跟公司有傘還是沒傘。若同時滿足沒傘且下雨，則需要買一把新的傘。判斷方式大概寫成下面這個樣子:

result 代表目前買了幾把傘

temp1 = 0 代表家裡沒傘 = 1 代表有傘

temp2 = 0 代表公司沒傘 = 1 代表有傘

if (weather[0] = rainy ) result ++ temp2 ++

if (temp2 == 0 && weather[1] = rainy ) result ++ temp1 ++

if (temp1 == 0 && weather[2] = rainy ) result ++ temp2 ++

利用這個概念我開始撰寫我的 pseudo code，寫完後一行一行確認沒有問題在轉換成程式碼。最後執行發現自己的答案有誤，才發現自己設定的判斷方式有個問題，當家裡或公司的傘超過 1 把時，就不滿足判斷的條件了。我才把有傘的條件從 = 1 轉換成 != 0。

第三題: Check if two words are isomorphic to each other

判斷兩個字串是否為同構，我發覺關鍵就是字母的位置。於是我先將問題拆解開來，變成如何找出一個字串中是否有重複字母，如果有就把位置記錄下來。由於我是個陣列愛好者，很喜歡用 push 的方式把東西都 push 到陣列裡。

像是下面這樣 : 字串ABCAAC A字母有重複，位置為 [0,3,4]， C字母有重複，位置為 [2,5] 所以字串的陣列會長這樣 [ [0, 3, 4] , [2, 5] ] 。我把程式設計成，第一個字串若有重複字母，則在陣列中紀錄字母的位置，再看第二個字串是否有重複的字母，記錄到第二個陣列裡面，再比較兩個陣列是不是長的一樣。結果跟預料的不一樣， [ [0, 3, 4] , [2, 5] ] 不等於 [ [0, 3, 4] , [2, 5] ]。回去翻翻筆記看看哪裡搞錯了，果然這問題老師早就講過了，是新手一定會弄錯了概念，陣列用等號來判斷不會是裡面的內容，而是記憶體位置。

老師出題真的是挑重點出的，用題目來提醒我們避免犯大家容易犯的錯誤。後來我將陣列內容改成字串，就可以成功通關啦。這題也再一次提醒我把這個容易忘記的觀念記在心裡。

我每一道題目平均大概花了 20 分鐘左右，會想選擇這三道題目的原因是，認真把老師教會我們的技能練習過，真的就可以解出大部分的基本題型。其實每次看到 codewar 分享的最佳解答，長的總是特別厲害，反觀自己寫的看起來就相對冗長很多。就會很崇拜這些高人能寫出厲害的程式碼，雖然有想試著每次都用厲害的寫法來寫程式，不過想起 Huli 老師說過，新手先求有再求好，按照老師所教的思維來解題，一步一步踩穩把問題拆開來處理。才發現簡化問題真的很重要，把複雜的問題，適時的拆解開來，才不會讓自己陷於慌亂之中。

因為上過 [alg101] 先別急著寫 leetcode 之後真的讓我信心大增，心想無論如何一定要把老師用來篩選的題庫通通解完，讓自己不要落後在起跑點。看懂問題，寫出 pseudo code，轉換成程式碼，一行一行檢視程式執行的步驟，這些方法真的很有效而且非常簡單，就像是玩馬力歐，開局直接拿到無敵星星一樣，關卡雖然有難度，但是 Huli 老師的無敵星星可以讓我一路衝到終點的感覺。雖然有時候若沒注意到陷阱，可能掉進洞裡失敗，不過試個兩三次多半就能通關了。

**8. 任何你想跟我分享的東西**

之前有在老師的文章看到老師用「 跑跑卡丁車 」來做說明。想必老師也很喜歡玩這個遊戲，所以我想跟老師分享我考到的「 跑跑卡丁車手遊 」 L2 駕照！

如果老師也有玩可以加我好友唷＾＾



**9.其他**

關於我前面所提，我個性上比較鑽牛角尖，處理問題習慣先以想到的第一種解決方式來解決，即便大家會認為第二種方法可以比較省時間，在這個習慣方面想請教老師，進入職場後會不會讓主管或同事覺得困擾？