

Ontrack

答應我兩件事

各位同學，在繼續進行下去之前，拜託答應我兩件事。就當被我騙也好：

請先去讀完我最近寫的這兩篇文章：

讀完了，再回來看底下的指南。

學程式沒你想像的可怕：

這兩篇文章呢？是想告訴大家一個學程式，非常簡單的道理：

- 我知道大家在學習時都會害怕，因為你會踢到很多鐵板，遇到很多不懂的地方。
- 越害怕，當你做書上的練習時，你就會緊張，想要去查其他的資料。聽從第三方的意見。
- 然後，你就會越看越多參考書。書是看多了，最後你就搞矇了。

這是一般人學習程式失敗的最大原因。

其實...你知道嗎？初學程式最快最好的方法，是「跟著打」，「不懂的地方背起來」。

然後，跟著這份教材打三遍。然後你就會突然間「懂」很多東西了。解題越來越快，越來越知道自己在幹什麼。最後有辦法獨立開發程式。

我知道這段話，聽起來很唬人，但我跟你保證是真的。我教過 200 個以上的 Rails 學生。我可以跟你保證，每個人上課前都覺得我在虎爛，畢業心得卻充滿震驚。

我知道你來上這門課前，可能已經放棄好幾次了。但是，既然這一次你已經交了這麼多的學費了，我希望這一次你可以：

我要求的只有兩件事

1. 我設計的指南，每一步都有意思。仔細看，照著打，不要跳過上面的步驟，隨意照自己的意思胡改。我帶過超過 200 個學生，我知道大家亂改，會掉入什麼坑。掉進這些坑，你就會覺得程式好難，我一定學不會。那就浪費你交這麼多學費來上這個班的用意了。
2. 這份作業，上課前，請重複做個三遍。我沒有開玩笑。你沒有做個三遍，就是對自己的學費開玩笑。

放下心虛、放下害怕，先信我一次。

這份作業要怎麼寫？

1. 第一次做這份作業，你就跟著貼就行了。大致瞭解貼什麼 code 會產生什麼結果。（底下的解說，你忍不住時，再看一眼就行了，別停下來看太久，也別太糾纏一定要問助教問到懂，我不希望你挫折滿點）
2. 第二次，你不要貼 code，跟著上面一個字一個字「打」。這次就不要貼了。打錯字是很正常的，遇到錯誤畫面也是很正常的。我希望你看到「錯誤畫面」先不要慌，這些錯誤畫面也「只是英文字」，深呼吸一下，仔細看上面在說什麼，真的無法排除，上 slack 找助教教你。slack 上面有至少 3 個台灣助教 埋伏，他們都

對新手很友善，儘管問。

3. 第三次，我希望你還是跟著打，但這次去看底下的「解說」，了解自己在幹什麼。「運作」不懂的地方，問明白。

記得，第一遍沒做完，不要去「查」，放下你的慌張.....

沒搞懂不會死人的，學第一遍，沒人會笑你，別把自己大腦搞糊塗了，那別人真救不了你。

筆記要怎麼做？

1. 我希望在第二次做時，你開始做筆記。筆記呢，你可以到 <http://logdown.com> 申請一個 blog 帳號，開始寫你的筆記。筆記內容寫什麼呢？
 - 我學到什麼？用法？
 - 我遇到什麼錯誤了？解法是什麼？
2. 很多人執著於「紀錄正確的資訊」。但我認為「紀錄錯誤的資訊」同等重要，把你當初的錯誤紀錄寫下來，這可以保證一件事，就是下次你絕對不會「再遇到同樣的錯誤了」（很神奇，但我沒匡你）。當你把該犯的錯誤全犯一遍了，你就能自己獨立 debug 了。
3. 開學前我希望看到你的十篇筆記。

總結

說了這麼多，究竟開學前你要做哪些事呢？

- 練這份中級指南 3 遍。（可以練 5 遍，但別少於 3 遍）
- 交了這份指南上指定的作業
- 部落格 10 篇。（不限制字數）
- 把這篇指南印出來，放在你的電腦旁，遇到「自我懷疑」時再看一下。

