



第一堂课

编程并不会很困难

困难的是

大多数人容易退却或放弃...

▶ 情绪挫折

- ▶ 情绪挫折
- ▶ 陌生环境

- ▶ 情绪挫折
- ▶ 陌生环境
- ▶ 觉得学习是登山

我们会带你走出去

编程事实上是“解决问题”的方法

- ▶ 用电脑解决问题的思维
- ▶ 把大问题拆解成可以实作的小问题
- ▶ 在实作过程中找到资源

越来越“熟练”

不要想着“错了”、“没有头绪”

第一招

如何不掉入情绪陷阱

你编程时遇到的最大挫折是什么？

- 解决不了自己的问题
- 面对更大的问题慌张
- 怕被人耻笑（比隔壁差）

“心理建设”

孤独与无助是很正常的

- ▶ 你不是“学不会”，只是“还不熟练”
- ▶ 大家很愿意帮你，但要先学着救“自己”

大脑乱了的话，这样做

- ▶ 我遇到了什么问题？
- ▶ 你想要怎么解决
- ▶ 我希望别人怎么帮助我
- ▶ 如果别人现在无法帮助我的话，我的解决方案
- ▶ 这个问题卡住的话，会让我接下来无法继续下去吗？

第二招

如何不求助无门

伸手党

不小心成为，不小心被别人以为

为什么会被当成“伸手党”

- 太急着发问，没回去重看教程
- 太急着发问，而不试着自己看“非常明显”的答案
- 太急着解决，而没意识到现在这个任务，不重要

急

慢

發問指南

Yi-Ting Cheng edited this page 5 days ago · 8 revisions

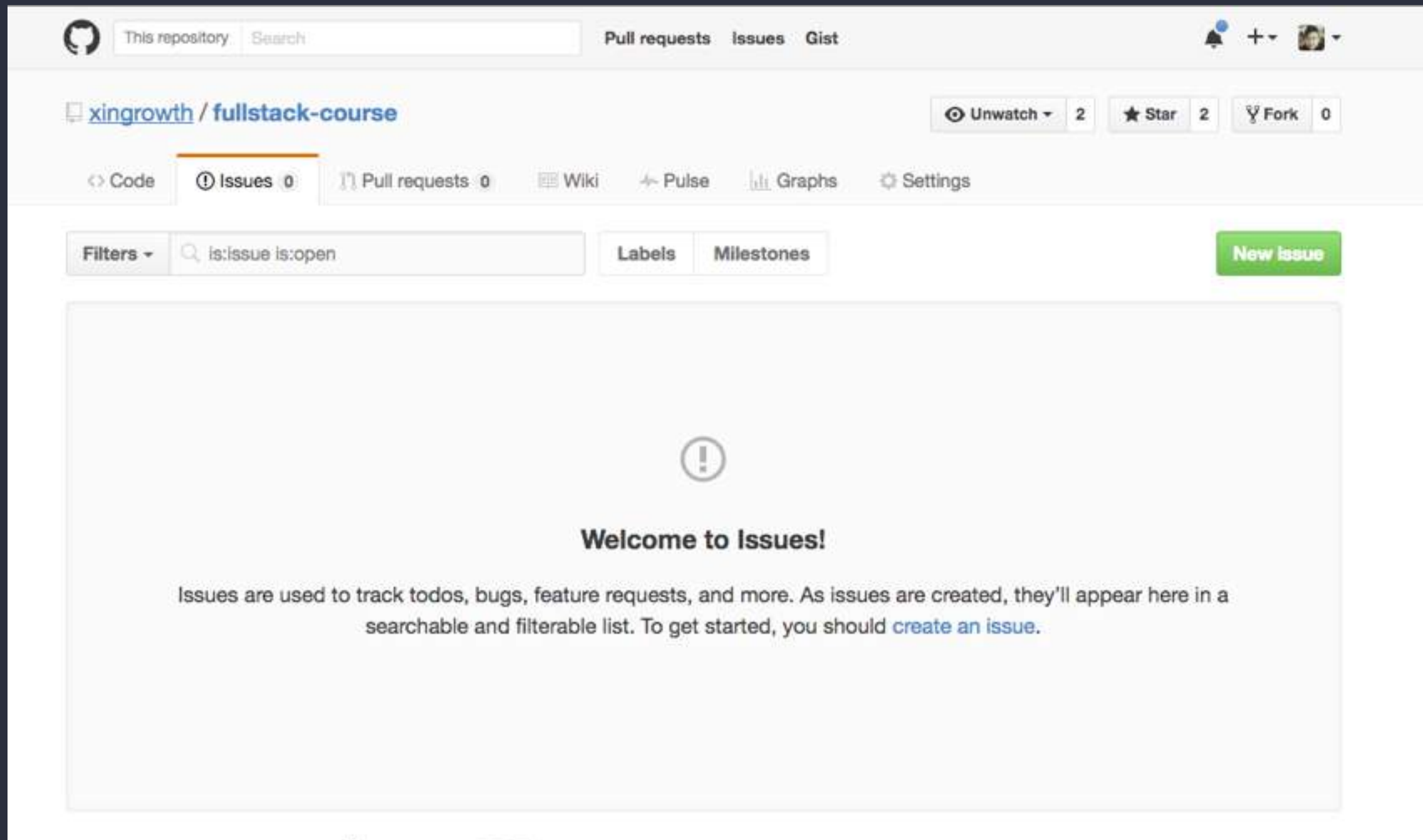
如何正確的在頻道上發問：

- 我的 程式碼（公開權限）的放置網址：<http://github.com/my-account-name/rails101>
- 正在執行什麼樣的動作：我目前在「第幾章的」「實作檔案上傳」。我在「傳檔時」遇到錯誤。
- 用的作業系統與 Ruby/Rails 版本：ruby 是用用「rvm 上的的 2.3.1」
- 目前瀏覽器的錯誤訊息截圖：（使用 [droplr](#) 上傳）
- 目前 rails s 最新的 log：（至少貼 100 行，且貼在 <http://gist.github.com> 上）

然後再貼到頻道上來

為什麼要這樣做

1. 當你說你「做了什麼」，得到「不如預期的結果」。旁人不知道「你做了什麼」，得到「什麼結果」。
 - 所以你必須提供「輸入」（輸入就是你的 程式碼，給「原始碼」是最好的了，所以需要你的 github 上傳最新結果）。
 - 接著你必須提供「輸出」（輸出就是你的 瀏覽器截圖，以及你的 rails s log 資料）
 - log 有時候很長，為了怕 log 洗到大家的版，請先把 log 貼到 Gist 然後再貼到頻道



<https://github.com/xingrowth/fullstack-course>

开发时遇到问题怎么办

- 去 issues 写下来
- 下午 15:30, 互助时间
- 尽量别打断别人编程, 你也不希望别人打断
你是吧!

第三招

如何不被自己绑架

- ▶ 饿了就去吃饭，不要想着没关系，饿了绝对解不出 bug

- ▶ 饿了就去吃饭, 不要想着没关系, 饿了绝对解不出 bug
- ▶ 累了就去睡觉, 不要想着没关系, 累了还是学不会 bug

- ▶ 饿了就去吃饭，不要想着没关系，饿了绝对解不出 bug
- ▶ 累了就去睡觉，不要想着没关系，累了还是学不会 bug
- ▶ 不会没关系，搞不懂也不要勉强搞懂它。想着三个礼拜后我就会自动懂。