

# iOS 程式設計

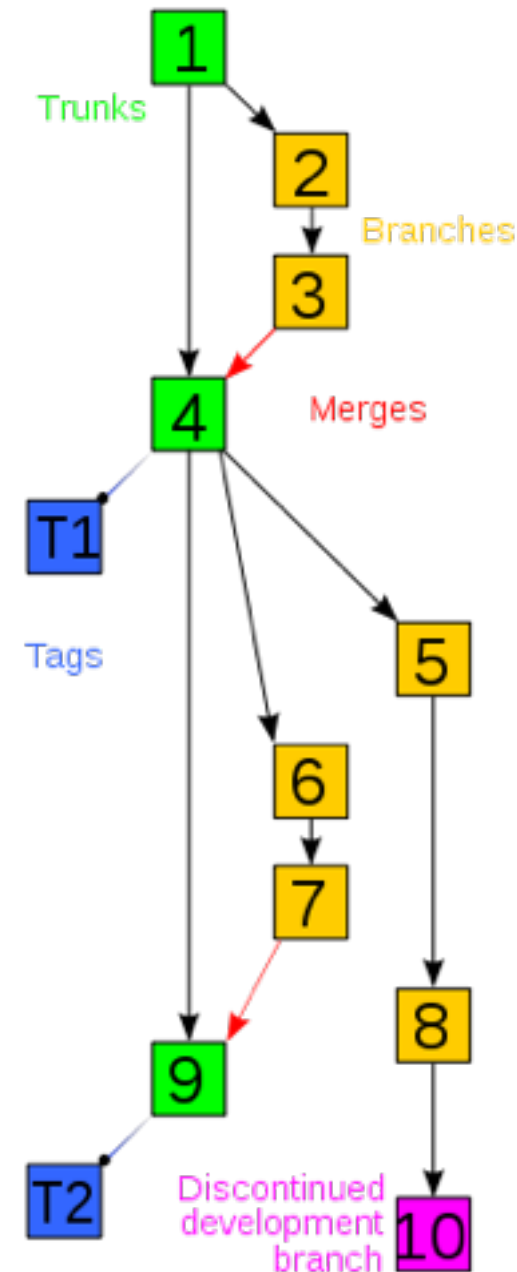
GitHub

王昱景 Brian Wang

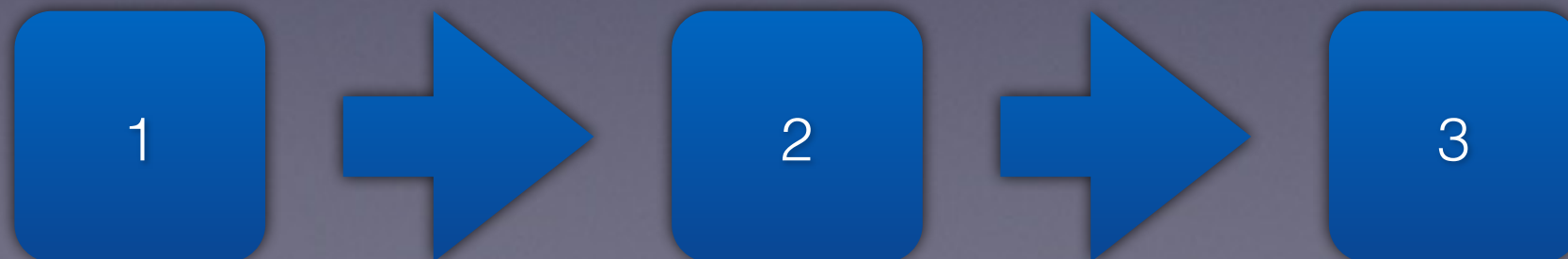
[brian.wang.frontline@gmail.com](mailto:brian.wang.frontline@gmail.com)

# 版本控制

- 版本控制 (Revision control) 是維護工程藍圖的標準作法，能追蹤工程藍圖從誕生一直到定案的過程
- 版本控制也是一種軟體工程技巧，藉此能在軟體開發的過程中，確保由不同人所編輯的同一程式檔案都得到同步



- 透過文件控制，能記錄任何工程專案內各個模組的改動歷程，並為每次改動編上序號
- 一種簡單的版本控制形式如下：
  - 賦給圖的初版一個版本等級「A」，當做了第一次改變後，版本等級改為「B」，以此類推
  - 最簡單的例子是，最初的版本指定為「1」，當做了改變之後，版本編號增加為「2」，以此類推



- 版本控制能提供專案的設計者，將設計回復到之前任一狀態的選擇權，這種選擇權在設計過程進入死胡同時特別重要
- 理論上所有的資訊記錄都可以加上版本控制，在過去的實務中，除了軟體開發的流程，其它的領域中很少有使用較複雜的版本控制技巧與工具
- 目前已有人開始用版本控制軟體來管理 CAD 電子檔案，電路板設計，來補足本來由人手工執行的傳統版本控制



# 軟體版本控制

- 軟體設計師常會利用版本控制來追蹤、維護源碼、檔案以及設定檔等等的改動，並且提供控制這些改動控制權的程式
- 在最簡單的情況下，軟體設計師可以自己保留一個程式的許多不同版本，並且為它們做適當的編號
- 這種簡單的方法已被用在很多大型的軟體專案中，該方法雖然可行，但不夠有效率
- 除了必須同時維護很多幾乎一樣的源碼備份外；而且極度依賴軟體設計師的自我修養與開發紀律，但這卻常是導致錯誤發生的原因

- 有時候，一個程式同時存有兩個以上的版本也有其必要性
- 例如：在一個為了部署的版本中程式錯誤已經被修正、但沒有加入新功能；在另一個開發版本則有新的功能正在開發、也有新的錯誤待解決，這使得同時間需要不同的版本並修改
- 此外，為了找出只存在於某一特定版本中（為了修正了某些問題、或新加功能所導致）的程式錯誤、或找出程式錯誤出現的版本，軟體除錯者也必須藉由比對不同版本的程式碼以找出問題的位置

# 軟體版本的控制方法

- 最簡單的版本控制就是保留軟體不同版本的數份copy，並且適當編號
- 許多大型開發案都是使用這種簡單技巧
- 雖然這種方法能用，但是很沒效率
- 一是因為儲存的數份 copy 幾乎完全一樣，也因為這種方法要高度依靠開發者的自我紀律，而常導致錯誤
- 因此，有人開發出了將部份或全部版本控制工作自動化的版本控制系統

- 差分編碼

- 大部份的版本控制軟體採用差分編碼：只保留檔案相繼版本之間的差異，這個方法可以更有效的儲存數個版本的檔案

- 檔案上鎖

- 檔案上鎖功能能對高難度的合併（例如大幅更改大檔案或檔案群的許多部份）提供一些保護，但其他開發者仍然可以繞過版本控制系統改變檔案（這本身就是很大的問題），所以檔案上鎖功能帶來的功效與副作用一直飽受爭議



- 中央式系統與分散式系統

- 大部分的軟體開發案，會有好幾個開發人員同時工作，如果兩個人員同時要改變同一個檔案，而沒有管理存取權限，很可能會覆寫彼此的工作
- 所以權限管理控制系統會在兩種方法中擇一解決：採用中央式系統，由中央權威管理存取權限；或是像分散式系統容許多個單位同時進行，包括同時更動同一檔案
- 傳統上版本控制系統都是採用中央式系統：所有版本控制的工作在一個伺服器進行，由中央權威管理存取權限「鎖上」檔案庫中的檔案，一次只讓一個開發者工作

- 2000 年後，TeamWare、BitKeeper、和 GNU 開始用分散式系統：開發者直接在各自的本地檔案庫工作，並容許多個開發者同時更動同一檔案，而各個檔案庫有另一個合併各個改變的功能
- 這個方式讓開發者能不靠網路也能繼續工作，也讓開發者有充分的版本控制能力，而不需經中央權威許可
- 分散式系統仍然可以有檔案上鎖功能
- 分散式系統 Linux 內核的發明人林納斯·托瓦茲就是分散式版本控制系統的支持者，他開發了目前被開源社群廣泛使用的分散式版本控制系統 Git

# 術語

- 基線 (Baseline) - 基線是軟體文件或源碼 (或其它產出物) 的一個穩定版本，它是進一步開發的基礎
- 檔案庫 (Repository)
  - 儲存檔案的新版本還有歷史資料的地方，通常是在伺服器上
  - 工作版本 (Working copy)：從檔案庫中取出一個本地端 (客戶端) 的複製，針對一個特定的時間或是版本
  - 所有在檔案庫中的檔案更動，都是從一個工作版本中修改而來的，這也是這名稱的由來，觀念上，這是一個沙盒

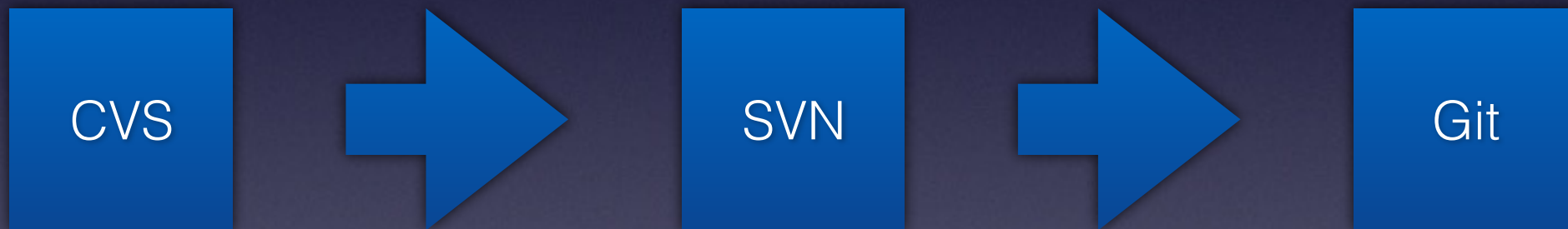


- 提交 (Commit) - 將本地端的修改送回檔案庫 (由版本控制軟體處理「跟上次更動相比，哪個檔案又被更動」的事)
- 變更 (Change) - 對一份檔案作的特定更動
- 變更記錄 (Change List)
- 取出 (Check-Out) - 從檔案庫取出檔案到本地端 (客戶端)
- 更新 (Update) - 將檔案庫的修改送到本地端 (與提交相反)



- 合併 (Merge / Integration) - 合併各個改變
- 版次 (Revision) - 一個 revision 或 version 指的是一系列版本變遷的其中之一
- 匯入 (Import)
- 匯出 (Export)
- 衝突 (Conflict) - 當兩方更動同一份檔案會發生衝突

# 版本控管演進



# CVS

- CVS (Concurrent Versions System) 代表協作版本系統或者並行版本系統，是一種版本控制系統，方便軟體的開發和使用者協同工作
- 這是一個將一組檔案放在層次目錄樹中以保持同步的系統，人們可以從 CVS 伺服器上更新他們的本地層次樹副本，並將修改的結果或新檔案發回；或者刪除舊檔案
- CVS 基於客戶端/伺服器結構的行為使得其可容納多使用者，構成網路也很方便

- 這一特性使得 CVS 成為位於不同地點的人同時處理資料檔案（特別是程式的原始碼）時的首選（現已被Git、SVN等逐漸替代）
- 限制
  - CVS 不支援檔案的複製和重新命名
  - 沒有原子性提交（Atomic commit）
  - CVS 只支援文字檔



# SVN

- SVN (Apache Subversion) ，一個開放原始碼的版本控制系統，相較於 RCS、CVS，它採用了分支管理系統，它的設計目標就是取代 CVS

- 特性

- 統一版本號

- CVS 是對每個檔案順序編排版本號，在某一時間各檔案的版本號各不相同
    - 而 Subversion 下，任何一次提交都會對所有檔案增加到同一個新版本號，即使是提交並不涉及的檔案
    - 所以，各檔案在某任意時間的版本號是相同的
    - 版本號相同的檔案構成軟體的一個版本

- 原子提交
  - 一次提交不管是單個還是多個檔案，都是作為一個整體提交的
  - 在這當中發生的意外例如傳輸中斷，不會引起資料庫的不完整和資料損壞
- 重新命名、複製、刪除檔案等動作都儲存在版本歷史記錄當中
- 對於二進制檔案，使用了節省空間的儲存方法（簡單的理解，就是只儲存和上一版本不同之處）

- 目錄也有版本歷史
  - 整個目錄樹可以被移動或者複製，操作很簡單，而且能夠保留全部版本記錄
- 分支的開銷非常小
- 最佳化過的資料庫存取，使得一些操作不必存取資料庫就可以做到，這樣減少了很多不必要的和資料庫主機之間的網路流量
- 支援後設資料（Metadata）管理，每個目錄或檔案都可以定義屬性（Property），它是一些隱藏的鍵值對，使用者可以自訂屬性內容，而且屬性和檔案內容一樣在版本控制範圍內



# Git

- Git 是一個分散式版本控制軟體，最初由林納斯·托瓦茲（Linus Torvalds）創作，於 2005 年以 GPL 釋出
- 最初目的是為更好地管理 Linux 內核開發而設計
- 與 CVS、Subversion 一類的集中式版本控制工具不同，它採用了分布式版本庫的作法，不需要伺服器端軟體，就可以運作版本控制，使得原始碼的釋出和交流極其方便



- Git 的速度很快，這對於諸如 Linux 內核這樣的大專案來說自然很重要
- Git 最為出色的是它的合併追蹤 (merge tracing) 能力
- Git 本身關心檔案的整體性是否有改變，但多數的 CVS 或 Subversion 系統則在乎檔案內容的差異
- 因此 Git 更像一個檔案系統，直接在本機上取得資料，不必連線到主機端取得資料

# 結構

- hooks：存儲鉤子的檔案夾
- logs：存儲日誌的檔案夾
- refs：存儲指向各個分支的指針（SHA-1標識） 檔案
- objects：存放 git 對象
- config：存放各種設置文件
- HEAD：指向當前所在分支的指針檔案路徑，一般指向 refs 下的某檔案

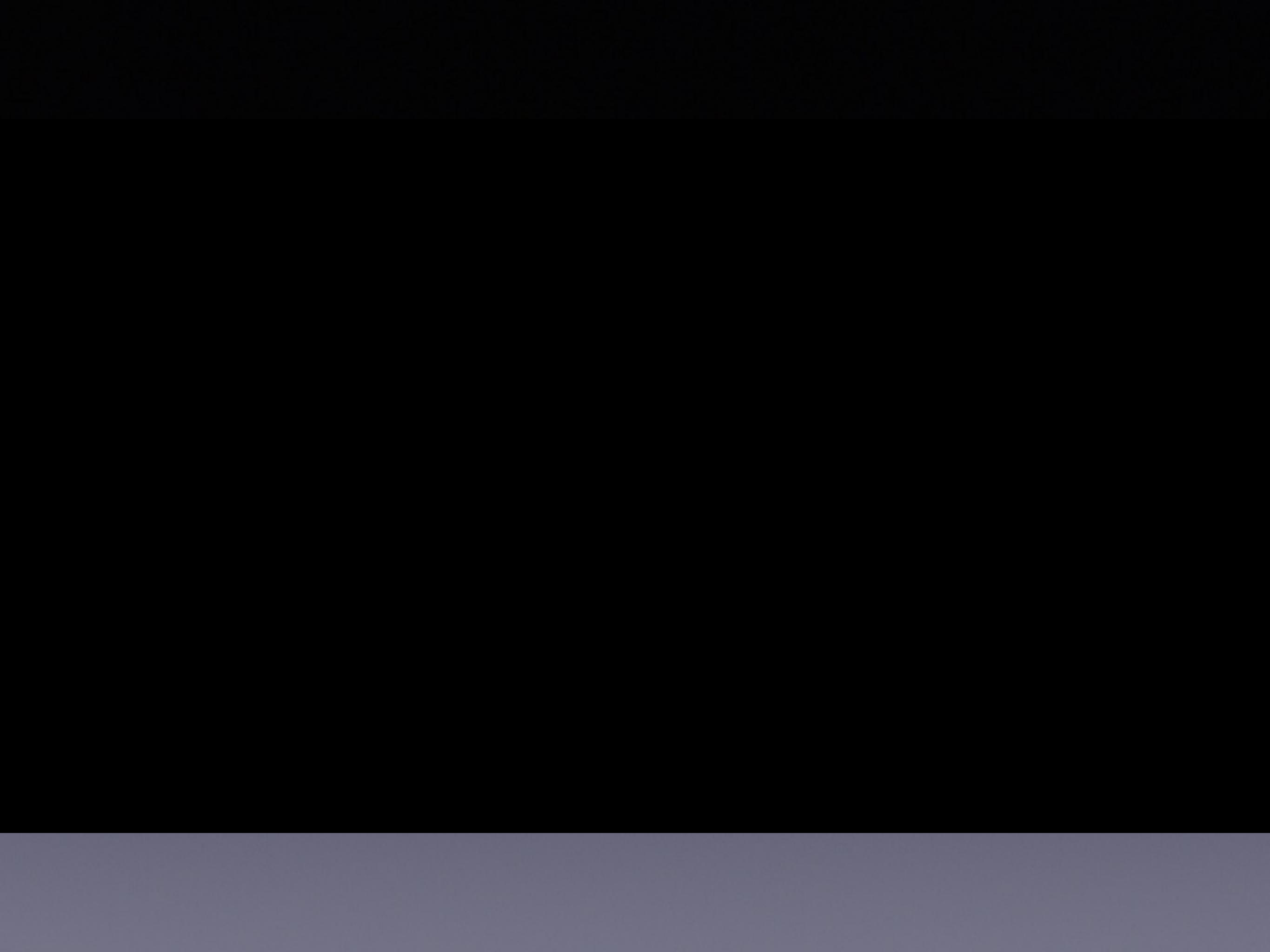
# GitHub

- GitHub 是一個共享虛擬主機服務，用於存放使用 Git 版本控制的軟體代碼和內容專案
- 由 GitHub 公司的開發者 Chris Wanstrath、PJ Hyett 和 Tom Preston-Werner 使用 Ruby on Rails 編寫而成
- GitHub 同時提供付費帳戶和為開源專案提供的免費帳戶





- 根據在 2009 年的 Git 使用者調查，GitHub 是最流行的 Git 存取站點
- 除了允許個人和組織建立和存取代碼庫以外，它也提供了一些方便社會化軟體開發的功能，包括允許使用者跟蹤其他使用者、組織、軟體庫的動態，對軟體代碼的改動和 bug 提出評論等
- GitHub 也提供了圖表功能，用於顯示開發者們怎樣在代碼庫上工作以及軟體的開發活躍程度
- GitHub 也提供一個貼上箱風格的站點 Gist，供軟體代碼庫使用的 Wiki，以及通過 git 版本庫進行編輯和管理的網頁代管功能



# Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Need private repositories?

Upgraded plans start at \$7/mo.

Use at least one lowercase letter, one numeral, and seven characters.

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.

## Want to use GitHub on your servers?

<https://github.com>



## Education from the experts

We love helping people just like you use GitHub and Git to collaborate and be more productive. No matter your level of experience, we've got something just for you.

### Online Training

No matter where you are in the world, we have live web-based classes that you can access from the convenience of your own computer.

[Learn from anywhere »](#)

### In-person Training

For face-to-face training, our highly experienced trainers can help increase your team's proficiency with a live, in-person class.

[Schedule a session »](#)

### Training Resources

We've open sourced all of our class outlines, slides, and cheat sheets so you can easily leverage these high-quality educational materials.

[Browse our resources »](#)

<https://training.github.com>





## GitHub & Git Foundations • Training

由GitHub Training & Guides建立 • 1/22 部影片



1 **Introduction • GitHub & Git Foundations**  
GitHub Training & Guides

2 **Setup • GitHub & Git Foundations**  
GitHub Training & Guides

3 **Config • GitHub & Git Foundations**  
GitHub Training & Guides

4 **Init • GitHub & Git Foundations**  
GitHub Training & Guides

5 **Commit • GitHub & Git Foundations**  
GitHub Training & Guides

## Introduction • GitHub & Git Foundations



GitHub Training & Guides



訂閱

18,042

19,924

+ 新增至



分享

... 更多

👍 82



2

發佈日期：2013年11月7日

We're excited to be refreshing the Foundations series. This is an intro to GitHub & Git, including basic



### Setup • GitHub & Git Foundations

由GitHub Training & Guides建立

觀看次數：11,227



### Config • GitHub & Git Foundations

由GitHub Training & Guides建立

觀看次數：9,478



### How to use GitHub for Beginners

由Chris Lam建立

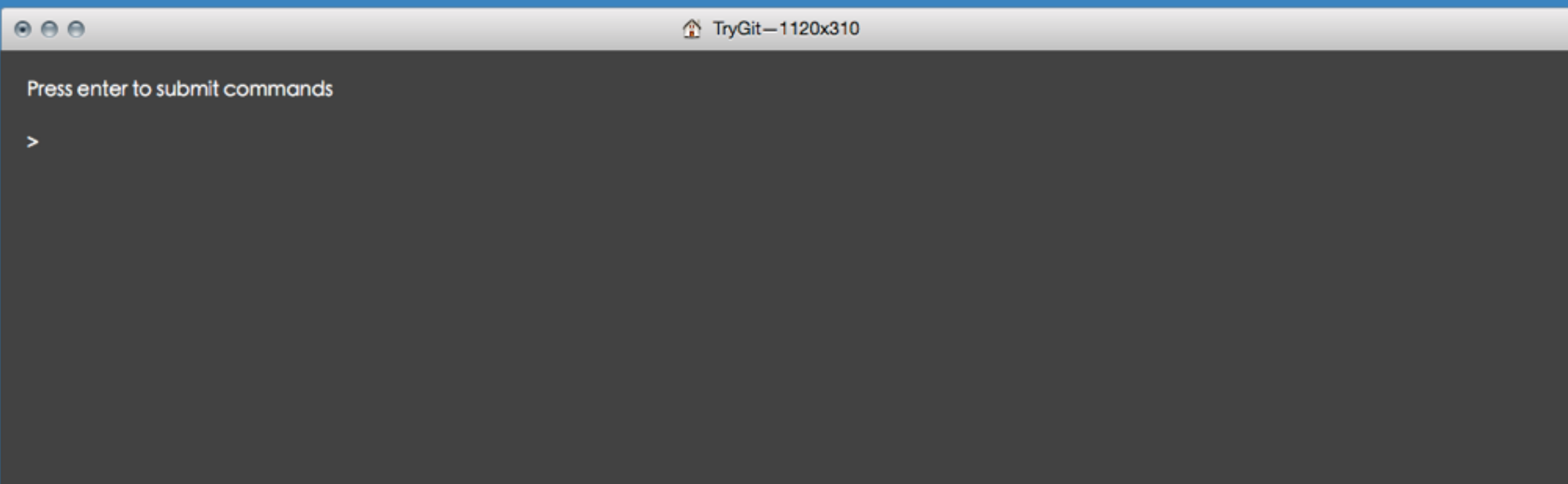


## 1.1 · Got 15 minutes and want to learn Git?

Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes. It's a distributed version control system.

Our terminal prompt below is currently in a directory we decided to name "octobox". To initialize a Git repository here, type the following command:

→ **git init**



<http://try.github.com/>



# 連猴子都能懂的Git入門指南

讓我們學習如何使用版本控制系統

推文

40

f 讚

2,842

g+1

291

目錄

入門篇

進階篇

Git命令快速參考

## 連猴子都能懂的Git入門指南

讓我們學習如何使用版本控制系統

歡迎來到超級簡單的Git入門指南，讓我們一起學習如何使用Git版本控制系統吧！

本指南分成三個部分，Git初學者請從「入門篇」開始，有使用過Git的朋友推薦直接從『進階篇』開始，當您遇到『哎呀～這要怎麼處理？！』及不確定如何使用Git的時候，可以試著到『Git命令快速參考』查詢一下。



Git入門的反饋

<http://backlogtool.com/git-guide/tw/>



# Student Developer Pack

The best developer tools, free for students



## Learn to ship software like a pro

 Like 28k

 Tweet 9,410

There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

[Get your pack](#)

### THE TOOLS



A hackable text editor for the 21st Century

[DETAILS](#) Open Source by GitHub, free for everyone

<https://education.github.com/pack>



# GitHub 學生開發者方案

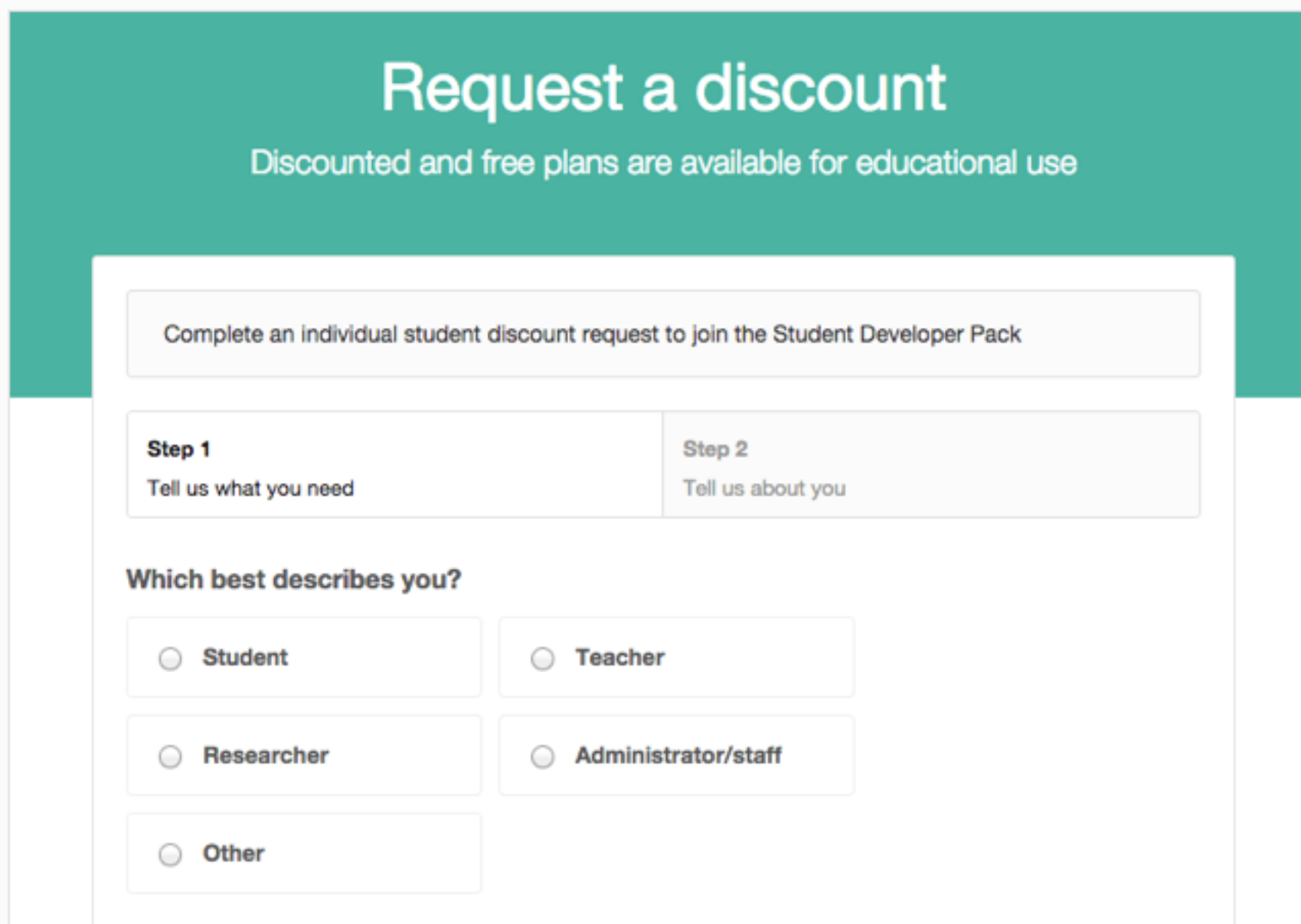
- Atom：由 GitHub 推出的程式碼編輯器，開放原始碼（免費）
- Bitnami：雲端應用平台，Business 3 Plan 一年（價值 \$49 美元/月）
- CrowdFlower：商業數據分析、處理平台，含 \$50 美元額度（價值 \$2,500 美元/月）
- DigitalOcean：雲端虛擬主機服務，含 \$100 美元額度
- DNSimple：網域名稱伺服器，Bronze 方案兩年（價值 \$3 美元/月）
- GitHub：免費升級為 Micro Account（價值 \$7 美元/月）
- Hackhands：24/7 即時程式設計協助平台，含 \$25 美元額度

- Namecheap：.ME 網域名稱一年、PositiveSSL 一年（價值 \$17.99 美元）
- Orchestrate：資料庫相關服務，Developer Account 一個（價值 \$49 美元/月）
- Screenhero：遠端連線工具，Individual Account 一個（價值 \$9.99 美元/月）
- SendGrid：Email 相關服務，學生方案每月可發送 15,000 封郵件
- Stripe：網頁及行動支付平台，首次 \$1,000 美元收益無需支付手續費
- Travis CI：針對開放原始碼或私人項目的集成平台，提供 Private Builds（價值 \$69 美元/月）
- Unreal Engine：針對 PC、行動裝置、網頁和 VR 所建置的遊戲開發平台，提供 Unreal Engine 帳戶權限（價值 \$19 美元/月）

## Step 1

首先，你必須要有一個 GitHub 帳戶，以及一個 **.edu.tw** 結尾的 **Email** 地址，以證明你是學生身份。如果沒有帳號，請先至 [GitHub 首頁](#) 註冊一個，填入你的使用者名稱、Email 和密碼即可註冊完成。

開啟 [GitHub Student Developer Pack](#) 網頁，點選上方的 Get Your Pack，然後登入你的 GitHub 帳號，登入後會看到如下表單，請依照你的實際情形作答。



The screenshot shows a web form titled "Request a discount" with a subtitle "Discounted and free plans are available for educational use". Below the title is a light green box with the text "Complete an individual student discount request to join the Student Developer Pack". The form is divided into two steps: "Step 1: Tell us what you need" and "Step 2: Tell us about you". Under Step 1, the question "Which best describes you?" is followed by five radio button options: "Student", "Teacher", "Researcher", "Administrator/staff", and "Other".

### Request a discount

Discounted and free plans are available for educational use

Complete an individual student discount request to join the Student Developer Pack

Step 1	Step 2
Tell us what you need	Tell us about you

**Which best describes you?**

- ☐ Student
- ☐ Teacher
- ☐ Researcher
- ☐ Administrator/staff
- ☐ Other

## Step 2

到第二個步驟時，請從 **Email** 裡選擇你學校提供給你的 **Email** 地址（**edu.tw** 或 **edu.cn** 結尾），如果你目前帳號使用的不是學校的 Email，點選下方的 Add and Verify It 來新增至你的 GitHub 帳號。

此外，還必須在頁面裡寫上你的學校名稱、以及申請學生開發者方案的用途，我是寫 coding 啦，你可以依照自己的情況填寫，不過可能要使用英文作答。

# Request a discount

Discounted and free plans are available for educational use

Step 1  
Tell us what you need

Step 2  
Tell us about you

**Name**

**Verify academic status**  
Select your school-issued email address:

If your school-issued email address isn't listed, please [add and verify it](#), then refresh this page.



### Step 3

在完成表單填寫後，GitHub 會需要一些時間審查，我大概等了五至十分鐘，回到 Student Developer Pack 頁面重新整理就拿到我的「學生開發者包」了，可以看到每個產品下方多出一個黃色框框，依照說明指示來索取代碼或鏈結。



Simple cloud hosting, built for developers

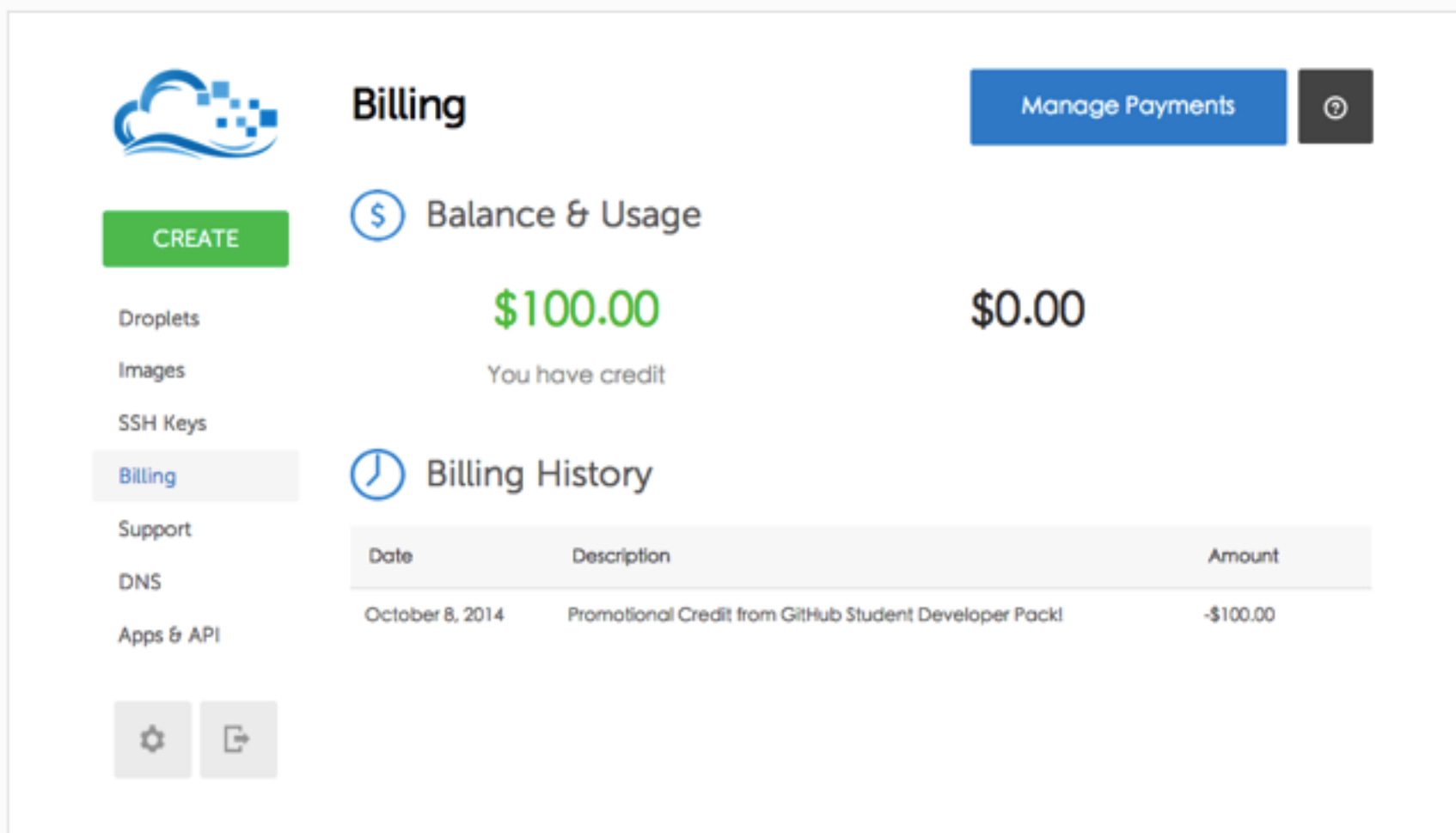
**DETAILS:** \$100 in platform credit

Use your offer code on the [DigitalOcean website](#)

Your code: **&I^\$%TKIUGAGITHUB-2a625790**

You can redeem your offer code using the link above, even if you have previously redeemed a different DigitalOcean promotional offer, credit card required for activation

我以 DigitalOcean 虛擬主機提供的 \$100 美元儲值額度為例，取得代碼後，在 DigitalOcean 網站兌換該代碼，帳戶內就會多一百元的可用餘額，這一百元能讓你開啟虛擬主機，練習如何架設網站。



The screenshot shows the DigitalOcean Billing page. On the left is a sidebar with navigation links: Droplets, Images, SSH Keys, Billing (highlighted), Support, DNS, and Apps & API. At the top of the main content area is the DigitalOcean logo, the word "Billing", a "Manage Payments" button, and a help icon. Below this is a "Balance & Usage" section showing a balance of \$100.00 (in green) and a usage of \$0.00. A note says "You have credit". Below that is a "Billing History" section with a table showing a promotional credit of \$100.00 on October 8, 2014.

Date	Description	Amount
October 8, 2014	Promotional Credit from GitHub Student Developer Pack!	-\$100.00

## Step 4

此外，這次的開發者方案裡還有 Namecheap 提供、可免費申請一年 .ME 網域名稱的優惠鏈結（每人只能申請一次），有需要的話不妨去註冊一個屬於自己的網址吧！

