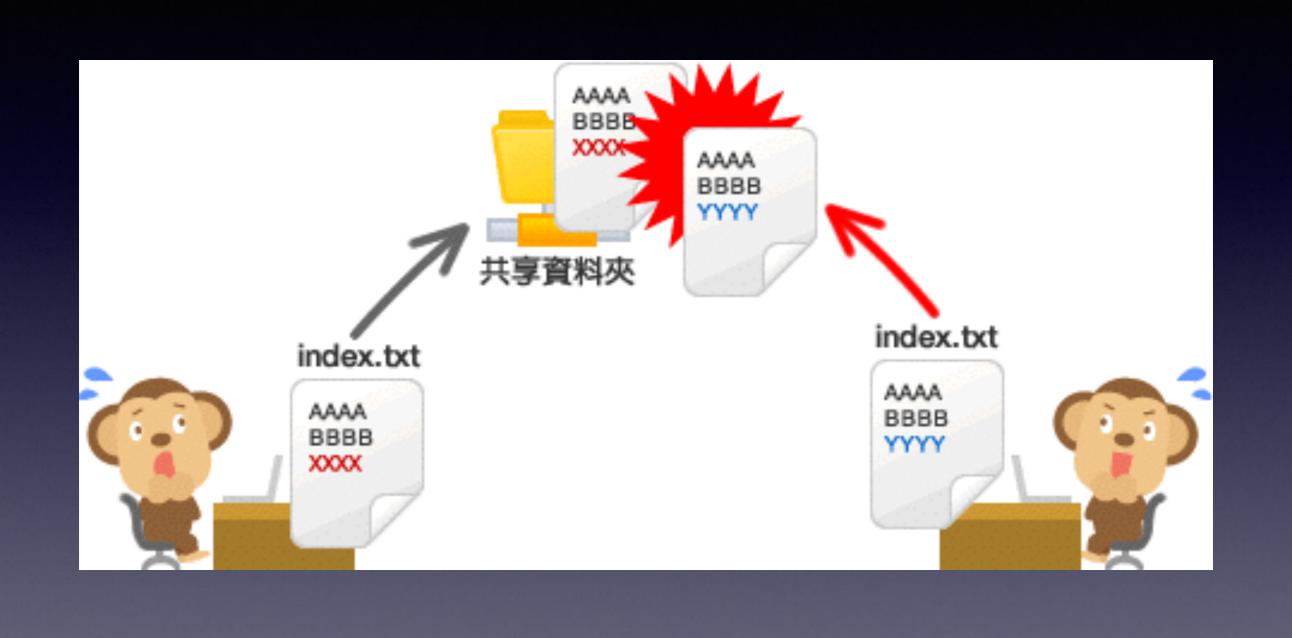
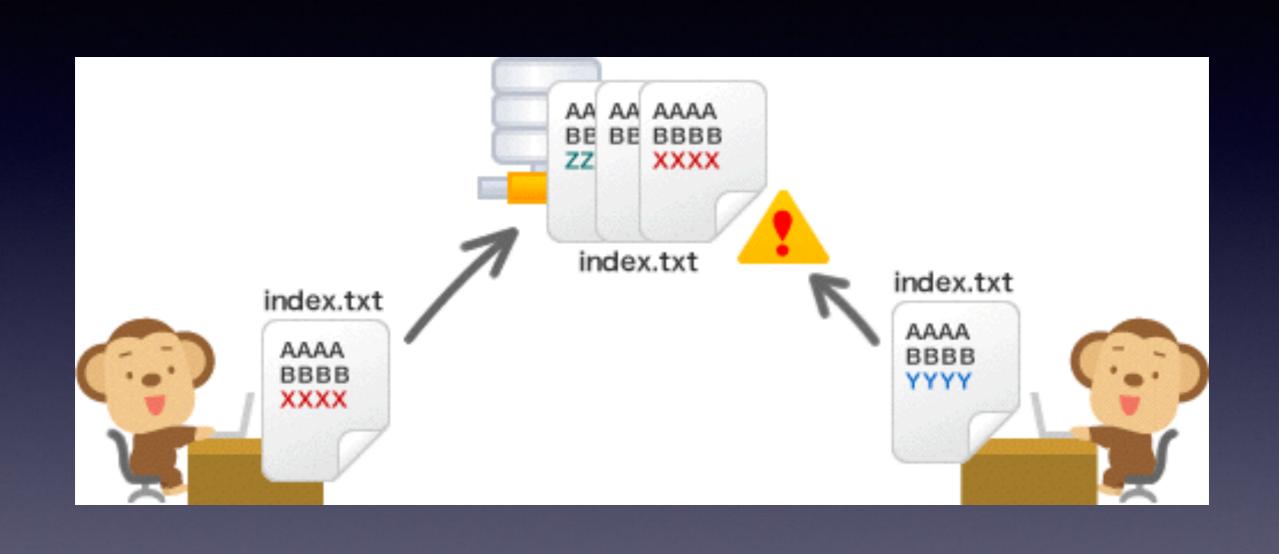
Matlab概論 GitHub

Brian Wang 王昱景 brian.wang.frontline@gmail.com



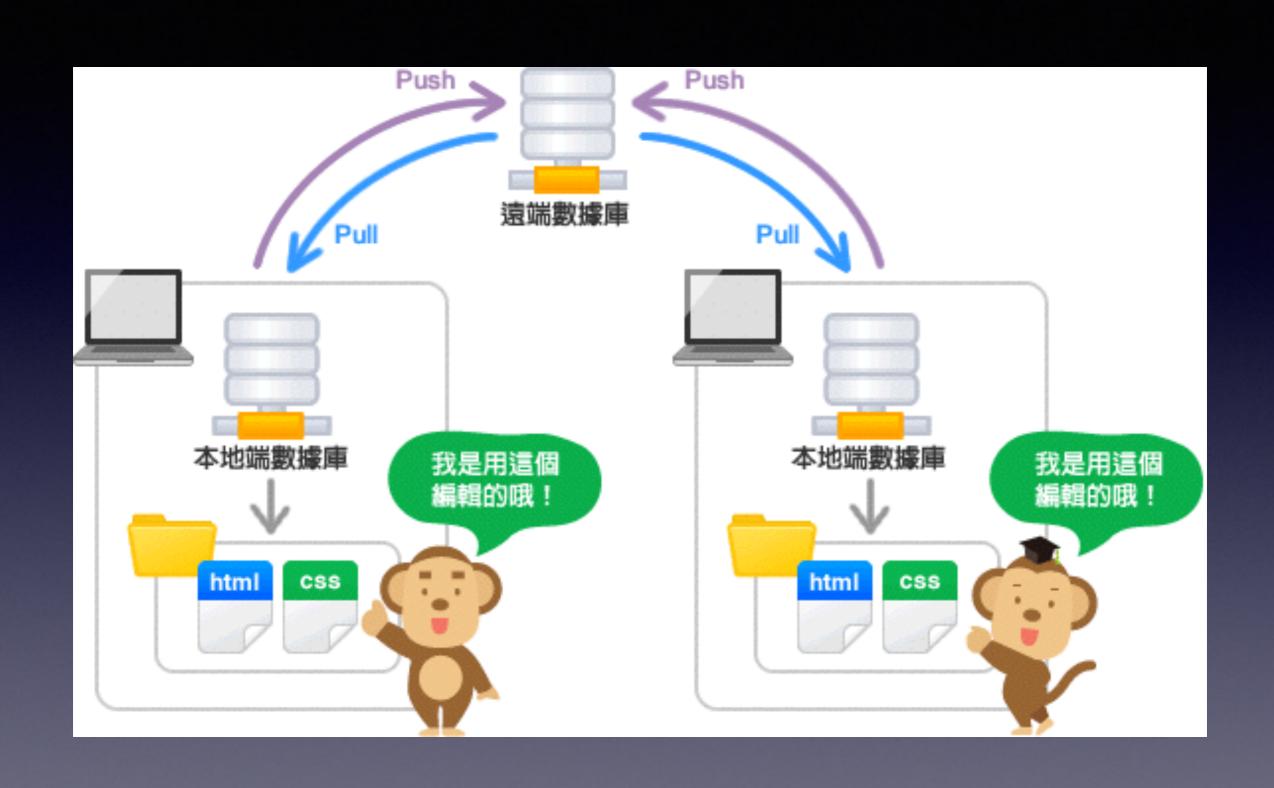




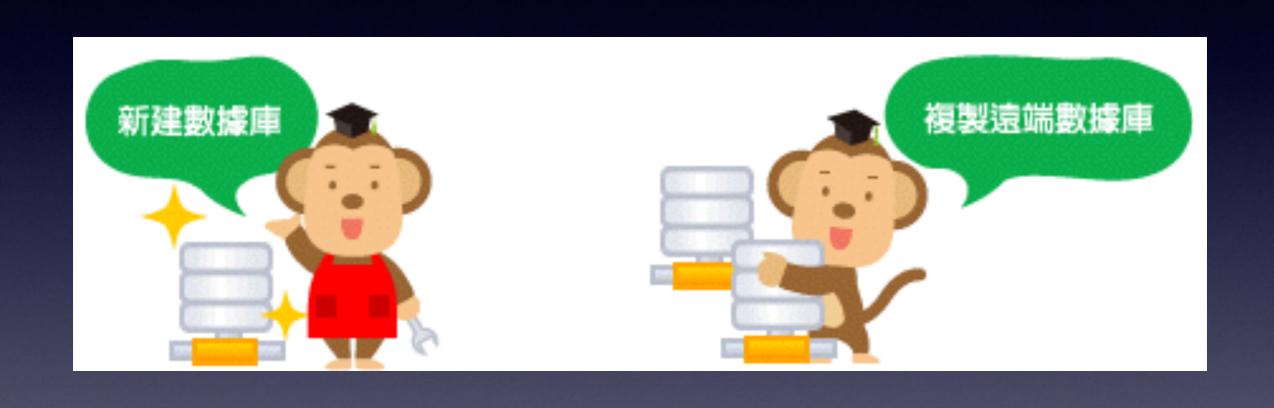




- 使用數據庫管理歷史記錄
- 數據庫 (Repository) 是記錄檔案或目錄狀態的地方, 儲存內容的修改歷史記錄。
- 在數據庫的管理下除了儲存修改歷史記錄外,還可以追蹤內容的狀態和版本喔。



- 遠端數據庫和本地端數據庫
- 遠端數據庫: 配有專用的伺服器,為了讓多人共享而建立的數據庫。
- 本地端數據庫:為了方便用戶個人使用,在自己的機器上配置的數據庫。

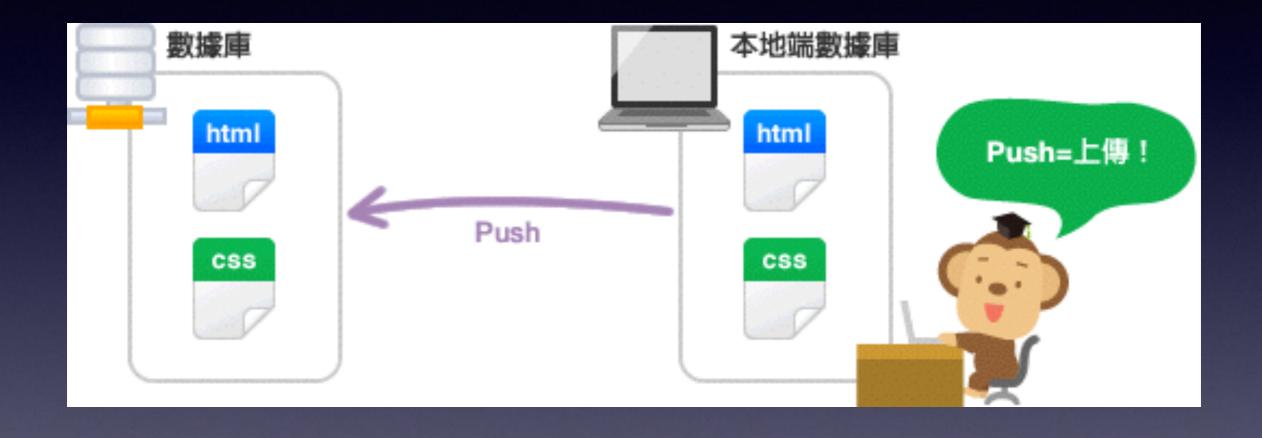




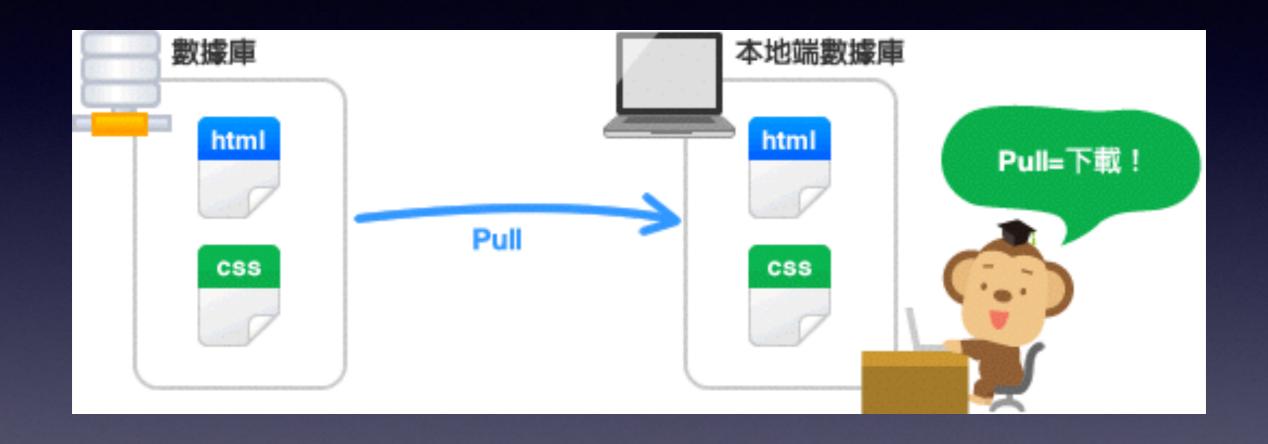
- 記錄修改的提交
- 若想把變更與新增的檔案/目錄儲存到數據庫中, 您需要執行提交(Commit)。
- 執行提交後,數據庫裡會產生上次提交的狀態與現在狀態的差異記錄(也被稱為 Revision)。

修改html檔案哦 工作目錄 數據庫 索引 html html html 提交 註冊 css css 沒有加入索引的檔案 不能被提交喔

- 工作目錄與索引
- 工作目錄(Working Tree)是保存您目前正在處理 檔案的目錄。
- 索引(Index)位於工作目錄和數據庫之間,是為 了向數據庫提交作準備的暫存區域。



- Push (推送)
- 為了將本地端數據庫的修改歷史共享到遠程數據 庫,必須上傳本地端數據庫中儲存的修改歷史。

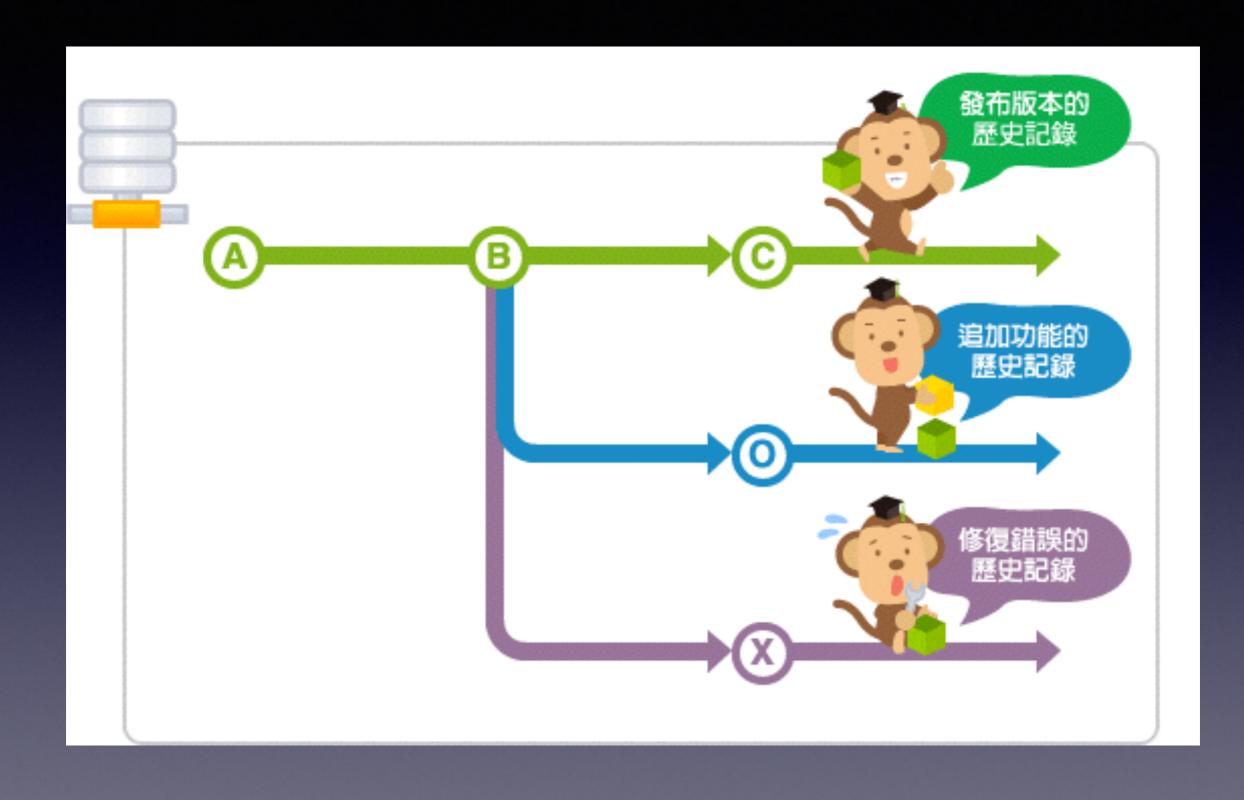


• Pull (拉取)

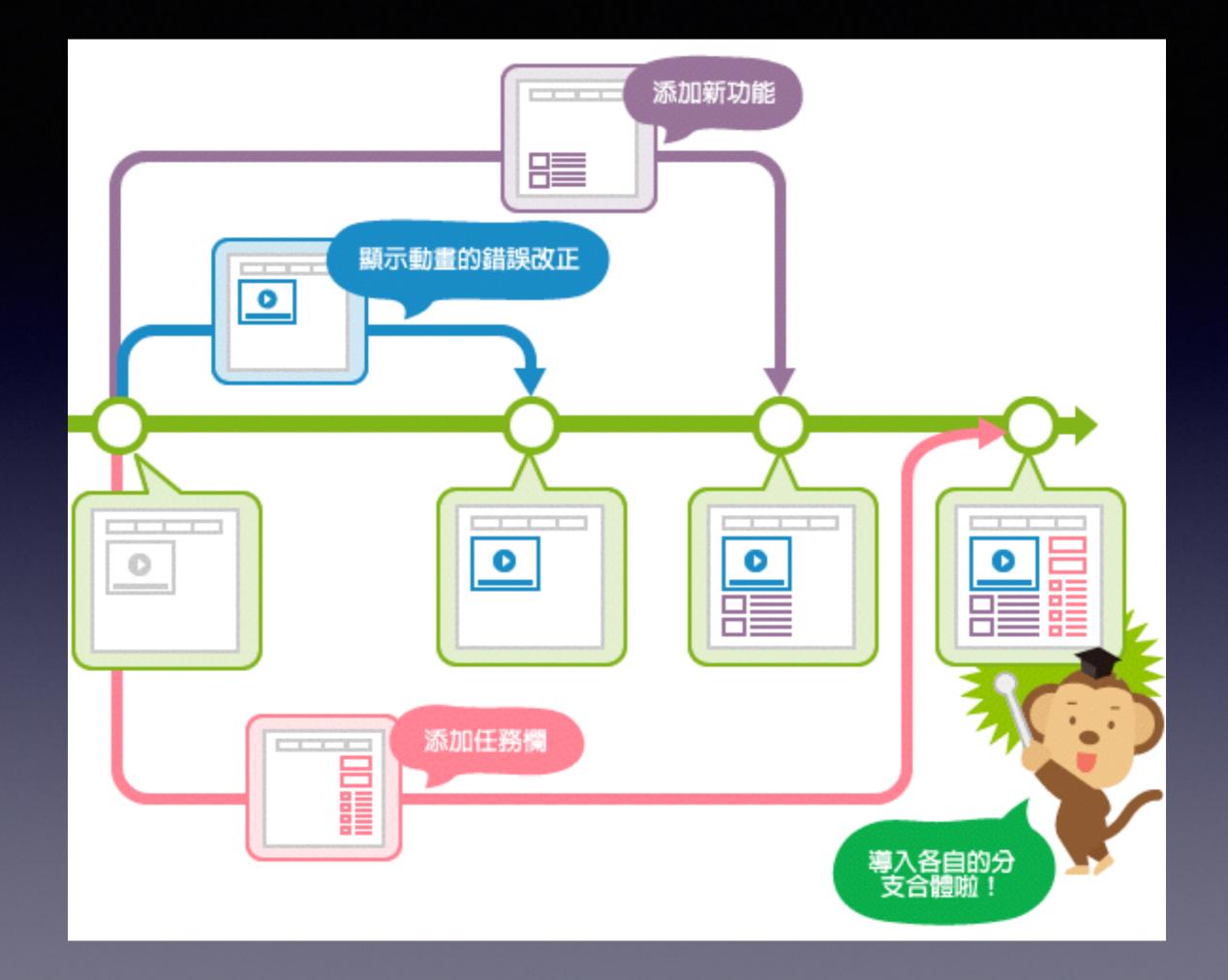
• 欲同步遠端數據庫以更新本地端數據庫,請使用 Pull (拉取)。

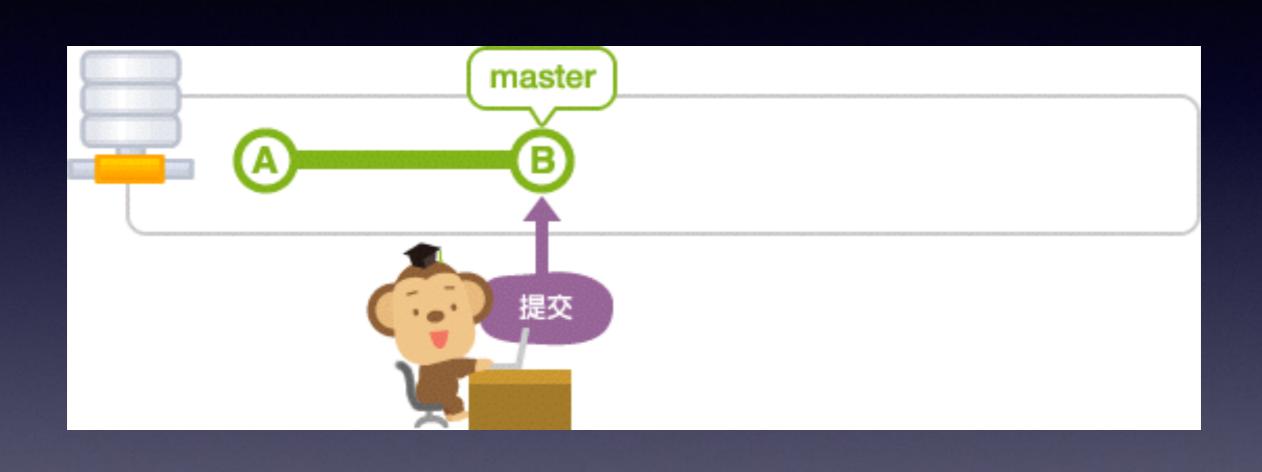


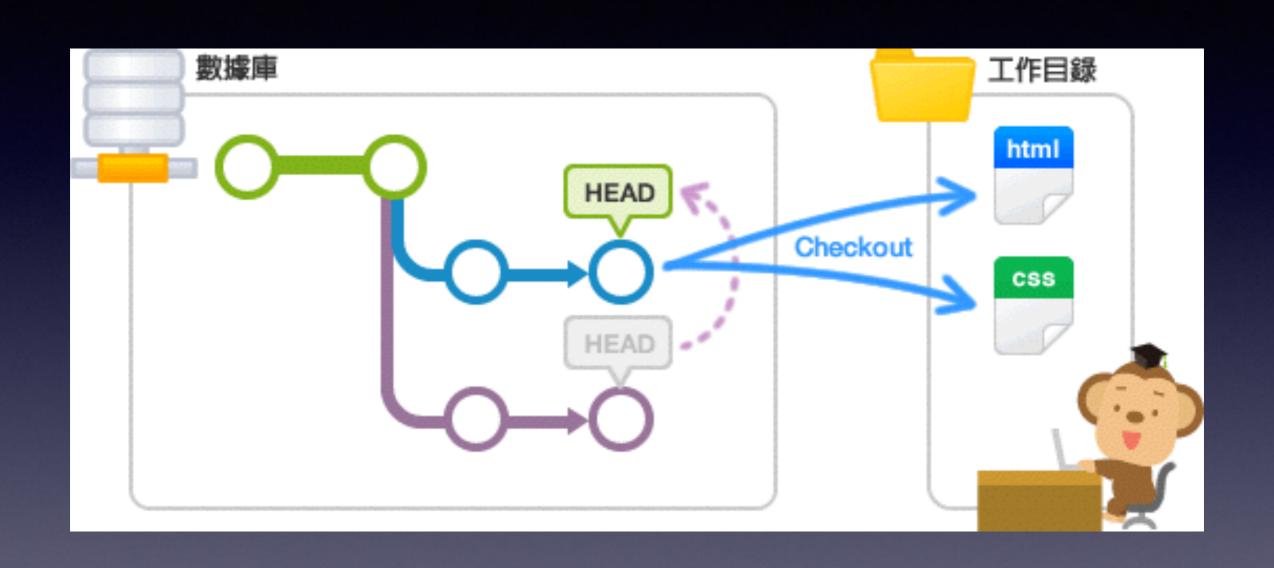




- 分支 (branch)
- 分支是為了將修改記錄的整體流程分開儲存,讓 分開的分支不受其他分支的影響,所以在同一個 數據庫裡可以同時進行多個不同的修改。



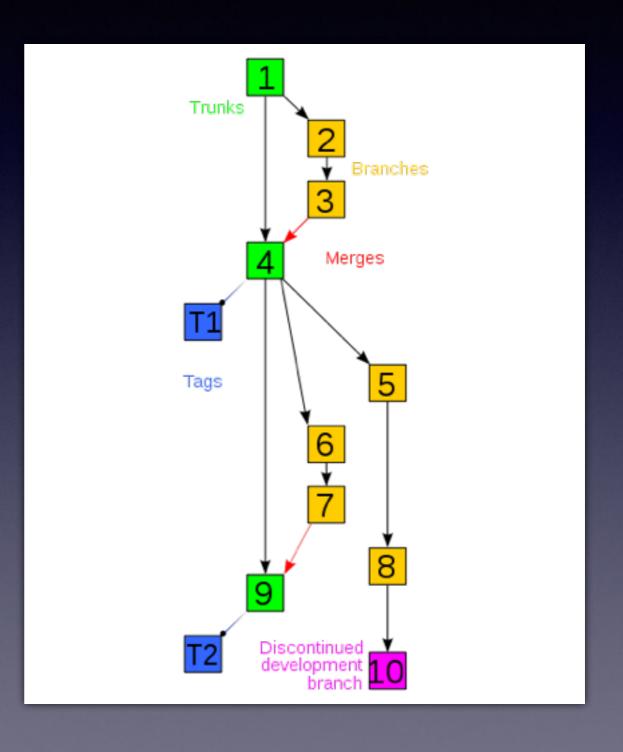






版本控制

- 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

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

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

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

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.



Learn Git in your browser for free with Try Git.





About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



Pro Git by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.







Tarballs



Windows Build



Source Code

Companies & Projects Using Git



facebook

Microsoft twitter Linked in.

























结構

- 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

Book

 Pro Git https://git-scm.com/book THE EXPERT'S VOICE®

SECOND EDITION

Pro Git

EVERYTHING YOU NEED TO KNOW ABOUT GIT

Scott Chacon and Ben Straub

.....

apress*

Tutorial

- Code School https://www.codeschool.com/learn/git
- Getting Git Right https://www.atlassian.com/git/



tryGit https://try.github.io/



• 連猴子都能懂的Git入門指南 連猴子 https://backlogtool.com/git-guide/tw/

Git

advanced version control skills. Learn how to collaborate on projects effectively with GitHub.



Paths > Git

Git 0/4 Complete

Git is the most popular version control system that developers use to track and share code. These courses will take you from a complete beginner to proficiency using Git and GitHub.



Try Git

Be introduced to the basic concepts of Git version control.



Git Real

Get a more advanced introduction and guide to Git.



Git Real 2

Learn more advanced Git techniques.



Mastering GitHub

Better collaboration through GitHub.



0/4 (0%)

My Account

FILTER COURSES BY:

All Courses

Git Basics

Advanced Git

Don't see a specific course? Request a course

Screencasts

In addition to our courses, we provide supplemental learning through screencasts. As you embark on the Git Path, the following videos will help you study on.



Feature Branches & Pull Requests

View All Git Screencasts

Create Free Account

Contact Code School

Blog

FivejS Podcast Front-end Five Podcast IOS Bytes Podcast



RubyS Podcast





Code School teaches web technologies in the comfort of your browser with video lessons, coding challenges, and screencasts. We strive to help you learn by doing. Learn More



(D) PLURALSIGHT

Getting Git Right



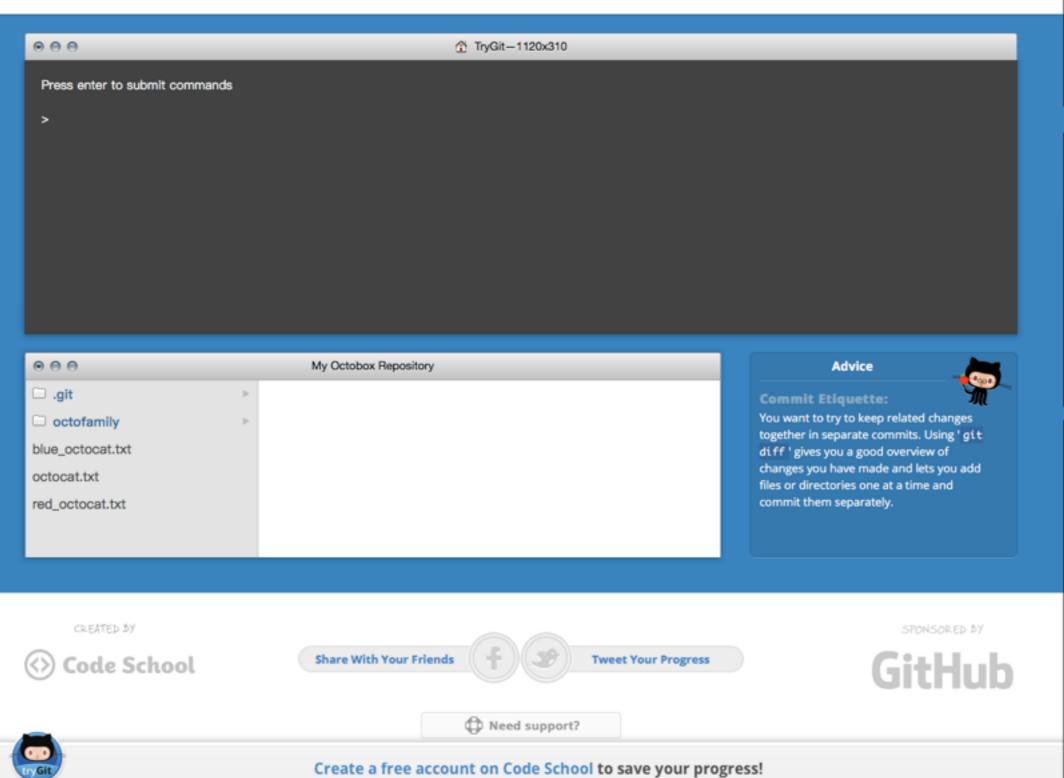
1.14 · Staged Differences

Another great use for **diff** is looking at changes within files that have already been staged. Remember, staged files are files we have told git that are ready to be committed.

Let's use git add to stage octofamily/octodog.txt, which I just added to the family for you.

→ git add octofamily/octodog.txt





Git命令快速參考

連猴子都能懂的Git入門指南

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。









Powered by



● 繁體中文 ▼ **★**

Copyright @ 2004-2016 Nulab Inc. All rights reserved.

Software

git for Windows https://git-for-windows.github.io



- GitHub https://desktop.github.com
- SourceTree https://www.sourcetreeapp.com



 TortoiseGit https://tortoisegit.org







Tools & Features

GitHub Desktop

Overview Release Notes | Help

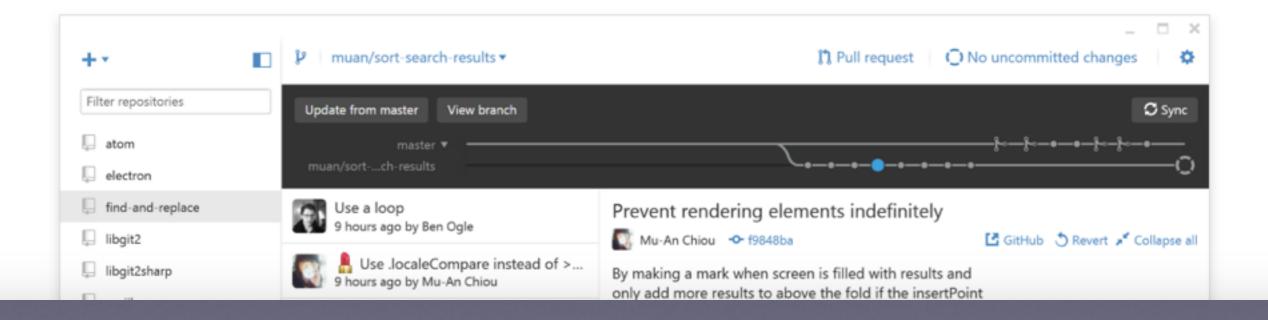
Simple collaboration from your desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise.

Available for Mac and Windows

Download GitHub Desktop Windows 7 or later

By clicking the Download button you agree to the End-User License Agreement

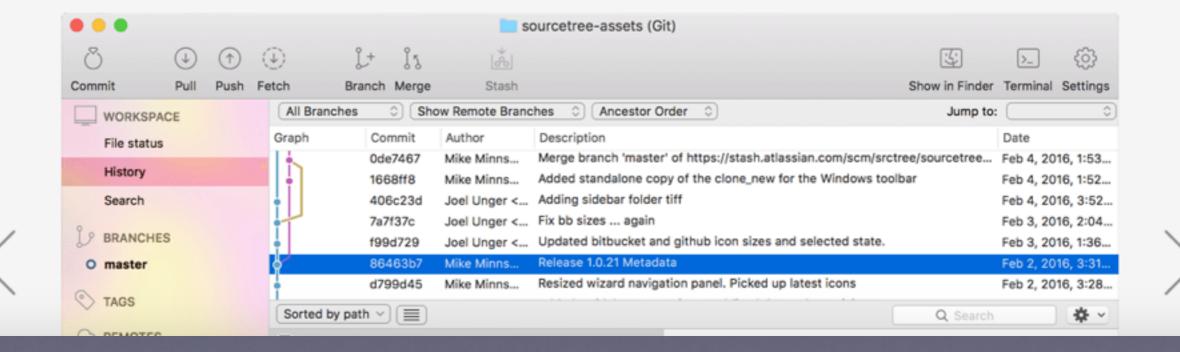




A free Git & Mercurial client for Windows or Mac.



Download SourceTree for Windows 7+



The Power of Git – in a Windows Shell

TortoiseGit provides overlay icons showing the file status, a powerful context menu for Git and much more!

Learn more about TortoiseGit.



News

- 0 2016-02-29 | Released TortoiseGit 2.0.0.0
- 0 2015-11-01 | Released TortoiseGit 1.8.16.0
- 0 2015-08-18 | Public repository for website online. Feel free to help ♂.
- O 2015-08-16 | New website put online. Still lots of stuff to do ;-)
- 9 2015-08-04 | Moved from GoogleCode to GitLab

Releases 🖈

- O Stable: 2.0.0.0 | Released on 2016-02-29
- O Preview: 1.8.16.4 | Released on 2016-02-14



Download

Here you find the TortoiseGit installer and language packs.



Contribute

TortoiseGit is open source and needs your help! Get involved!



Support

Manuals, FAQ, bug reporting, mailing list, and more...

Service

 GitHub https://github.com



bitbucket
 https://www.atlassian.com/software/bitbucket



Personal

Open source

Business

Explore

Pricing Blog Support

Sign in

Sign up



Welcome home, developers



Code, Manage, Collaborate

Bitbucket is the Git solution for professional teams

Get started for free

Host it yourself with Bitbucket Server

Built for professional teams

Distributed version control system that makes it easy for you to collaborate with your

Server

GitLab
 https://about.gitlab.com

Code, test, and deploy together



GitLab CE

Community Edition

Download and install the open source GitLab CE on your own server in 2 minutes.

Download



GitLab EE

Enterprise Edition

Additional features and commercial support for running GitLab on your own server.

Pricing



GitLab.com

SaaS

Free multi-tenant cloud hosting.
Unlimited (private) repos and
collaborators.

Sign Up



GitHost.io

Hosted Servers

A private single-tenant GitLab instance run by us. Available with CE and EE.

Learn More

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

GitHub Education

 Student Developer Pack https://education.github.com/pack

Student Developer Pack

The best developer tools, free for students



Learn to ship software like a pro







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

