Git Tutorial

http://ihower.tw/git 2017/3



- 張文鈿 a.k.a. ihower
 - https://ihower.tw
- Git user since 2008



Agenda

- 什麼是「版本控制系統」
- Git 簡介
- Git 內部原理
- Git 指令
- 更多 Git Tips
- GitHub 簡介

我的版本控制之路

- 2002 目錄管理法 (定時 copy 備份一次)
- 2005 SubVersion
- 2007 SVK
- 2008 Git (像 SVN 一樣只會 push/pull)
- 2009 Git (主要還是在 master 開發,有大功能才會開 feature branches)
- 2011 Git (根據 git flow,做任何事幾乎都開 topic branch。大玩 rebase)

I. 版本控制系統

5

Why? 要解決的問題

- 檔案被別人或自己覆蓋,甚至遺失
- 想復原前幾天寫的版本
- 想知道跟昨天寫的差在哪裡
- 是誰改了這段程式碼,為什麼
- 軟體發行,需要分成維護版跟開發版



- 可以隨時復原修改,回到之前的版本
- 多人協作時,不會把別人的東西蓋掉
- •保留修改歷史記錄,以供查詢
- 軟體發行時,可以方便管理不同版本

軟體開發的基本工具

- 版本控制系統不祇可以幫助妳追蹤修訂手上的工作進度,讓妳在千鈞一髮之際還能拾回過往辛苦的結晶,甚至能夠讓妳跟其他人協同工作、合作無間。
 <u>http://jedi.org/blog/archives/004784.html</u>
- 那些台灣軟體產業所缺少的 版本控制系統 <u>http://blog.ez2learn.com/2011/10/20/taiwan-software-lacking-of-vcs/</u>

Version Control System (VCS)

- •建立 Repository (儲存庫),用來保存程式碼
- 方便散佈程式給團隊,有效率協同開發
- 記錄誰改變什麼、在什麼時候、因為什麼原因
- Branch(分支),可因不同情境分開開發
- Tag(標籤) 重要里程碑,以便參照



- 所有跑起來這個專案需要的東西
 - 所有原始碼、範例設計檔、文件等等
 - 暫存檔、log 檔案、build files 等編譯後的產物 則不需要存進 Repository



Centralized VCS (Lock型,悲觀鎖定)



Centralized VCS (Merge型, 樂觀鎖定)

CVS, Subversion, Perforce



Distributed VCS Git, Mercurial(Hg), Bazaar



Local development

- 集中式的VCS系統,沒網路就不能開發, 無法 commit,無法看 history log。
 - 坐高鐵,坐飛機的時候
 - 網路故障的時候
 - 咖啡店沒有無線網路的時候
- 分散式VCS系統即使沒網路,照常可以 commit 和看 history log。
 不用擔心server備份

2. Git 簡介

16

Git

- 開放源碼的分散式版本管理系統(DVCS)
- 發明人 Linus Torvalds (Linux creator)
- 目的是管理 Linux Kernel 原始碼
- 2005/4 開始開發,2005/6 開始管理 Linux Kernel, 2005/12 釋出 I.0 版

誰在用 Git?

- Git
- Linux Kernel
- Ruby on Rails
- Perl
- Eclipse
- Android
- PostgreSQL

- Google
- Microsoft
- Facebook
- Twiter
- LinkedIn
- Adobe
- IBM

Why Git is Better than X http://zoomq.qiniudn.com/ZQScrapBook/ZqFLOSS/data/20081210180347/

- 便宜快速的本地分支
- 所有內容都在本地端
- Git 很快
- Git 很省空間
- Staging 功能
- 它是分散式的
- 適用任何工作流程 • 我們有 GitHub!

0.0	Why Cit is Retire Than X
 I 🕐 Major // Sh-two why pitister III Knowle Revel Tabley Revel Land 	enhans.com/Whie-staging-area d Q: Google a increaser fication flow Texture (actions) First Datum 2
Why Git is E	Setter than X not the setter
我會致意能總法是因为五花了很多可 (Autmagniture) 還角 Novield-Print- 推送這種做的原因,這樣點發任何一1	电依赖 Gatatera 加速,最佳部金红质者 (Ambayanti 完厚者 :這個態故術的就是為什麼人家都要這 X 执武 Out,因為非量件心 這個思示用的全員。
	2222 2018
便宣的本地分支	Ng tor and pattern
所有內容都在本則端	an parata
Git 很快	ior and partners
Git 很小	
暫存區域	Ng ter lant metere
它是分散式的	and and a second se
逾用任何工作流程	ant particula
我們有 GitHub !	Ng ini jan patron
容易學習	gating
	ORES OFF

Interstells duritate mantament ov Social Litects, a Githucoar. Myos, diageneeidth anything entherate and you have a good reason, please <u>emailme</u> so loan field. The sourcefor the stells o<u>m Stills b</u>-feelfree to send perchas if powershift imprese it.

Alex in : Jaconso Stanish Course Index Datab Encourses Octoberan Stanish Simultai and Encoderal Disea 2010 9 2 2 10 10 Santa Chen 1010 -

Taol THTM.





DAG 儲存方式 (DAG: Directed acyclic graph) 利用有向無環圖來記錄 metadata 建構出 snapshots。 相同內容只會有一份記錄。



00	i rails (branch: master)				
C			· ·		
Refresh			Terminal Reveal		
RAILS	= = € = ++ ++ ++				
Stage			Q* Subject, Author, SHA		
BRANCHES	Subject	Date	Author		
\$9 2-3-stable	O matter originHEAD originimater Update 4.0 Release Notes	July 4, 2012 7:23 PM	Rafael Mendonca F		
29 3-0-stable	Kill not used constant since removal of runner method	July 4, 2012 8:02 AM	Carlos Antonio da		
29 3-1-stable	remove duplicate requires of mocha.	July 4, 2012 5:28 AM	Aaron Patterson		
98 3 2 stable	remove the runner method copied from minitest	July 4, 2012 5:23 AM	Aaron Patterson		
67 3-2-stable	hook mocha in through m/t before_setup, after_teardown hooks	July 4, 2012 5:21 AM	Aaron Patterson		
57 fix_invalid_datatime_att	Merge pull request #6948 from frodsan/plugin_skip_ignore	July 4, 2012 1:00 AM	Rafael Mendonça F		
🐉 master 🛛 🥑	remove test/dummy from gitignore when "plugin new -T"	July 4, 2012 12:58 AM	Francesco Rodriguez		
\$9 patch_logger_encoding	Merge branch 'lexmag-reference_statements'	July 4, 2012 12:48 AM	Rafael Mendonça F		
REMOTES	On't need to use delete in the options hash	July 4, 2012 12:24 AM	Rafael Mendonça F		
REMOTES	 Add changelog entry for references statements 	July 3, 2012 3:44 AM	Aleksey Magusev		
▶ ⊕ origin	 Refactor references schema definitions 	July 3, 2012 3:43 AM	Aleksey Magusev		
TAGS	Make references statements reversible	July 3, 2012 3:42 AM	Aleksey Magusev		
	• Add references schema statements	July 3, 2012 3:41 AM	Aleksey Magusev		
OTHER	Merge pull request #6946 from threedaymonk/queue-refs	July 3, 2012 11:44 PM	Aaron Patterson		
	Make TestQueueTest work with marshalling queue	July 3, 2012 8:03 PM	Paul Battley		
	Censure jobs do not refer to the queue	July 3, 2012 6:59 PM	Paul Battley		
	C Ensure test jobs are marshallable	July 3, 2012 6:55 PM	Paul Battley		
	Marga pull request #6040 from acapillari/obsenus update	July 3, 2012 5:29 PM	Faul Battley		
	Merge pull request #6940 from acapilleri/observer_update		Carlos Antonio da		
	SHA: b5a2f24b6b2b466a5692f0d46b4a2b8154ee0e53 Gist it				
	Author: Rafael Mendonça França <rafaelmfranca@gmail.com></rafaelmfranca@gmail.com>				
	Date: Wed Jul 04 2012 19:23:51 GMT+0800 (CST)				
	Subject: Update 4.0 Release Notes				
	Refs: master origin/HEAD origin/master				
	Parent: f3353895a04dfd9a06cf1ca2e7bbfb80a38dbfad				
	Update 4.0 Release Notes				
	Related with 17d2115e3df6caf15d554b44e28177069fa3ed5b				
	guides/source/4_0_release_notes.textile				
	<pre>guides/source/4_0_release_notes.textile @@ -329,6 +329,20 @@ Moved into a separate gem <tto 329 329 330 330 h3. Active Record 331 331 332 +* Add <tt>add_reference</tt> and <tt>remove_reference 333 +</tt></tto </pre>	<pre>>sprockets-rails. ence schema statements.</pre>	Aliases, <tt>add_belongs_to+</tt>		
(\$ +) +0 + ↓ ± ±	9286 commits loaded				

Git graph example





finally figuring out that git commands are strangely named graph manipulation commands--creating/deleting nodes, moving pointers around

1 Mar via TweetDeck 🖕 Unfavorite 📭 Retweet 👆 Reply

用 Git 表示台北捷運

http://gugod.org/2009/12/git-graphing/

25



Try Git 十五分鐘體驗

https://www.codeschool.com/courses/try-git

3. Git 原理和內部運作

27

Source Control Made Easy

 <u>https://pragprog.com/screencasts/v-jwsceasy/</u> <u>source-control-made-easy</u>

Git SHAI (demo)

- echo sweet > sweet.txt
- git add .
- find .git/objects -type f
- Git 內部儲存在 .git/objects/aa/ 823728ea7d592acc69b36875a482cdf3fd5c8d
- 這是 "blob" SP "6" NUL "sweet" LF 的 SHAI
- printf "blob 6\000sweet\n" | shasum
 - 或 echo 'sweet' | git hash-object -w --stdin
- git cat-file -p aa823728ea7d592acc69b36875a482cdf3fd5c8d

Git is Content Tracker

- Git is content-addressable filesystem
- SHAI 是根據內容產生的,跟檔名無關
 - 內容一樣的檔案,可以 reuse,例如 revert 的情況
 - 解耦合(decouple)了內容與當時的檔名
- Git 有四種 Objects 記錄內容:Blob、Tree Commit、Tag



- 給定 message 容易計算出 hash
- 非常困難從 hash 推回 message
- 非常困難修改 message 而 hash 不變
- 非常困難有不同的 messages 而 hash 一樣
- hash 的分布很分散,跟 message 關聯不大

Git Object Database

Working Directory







Tree

記錄該目錄下有哪些檔案(檔名、內容的SHAI)和Trees



Commit

記錄 commit 訊息、Root tree 和 Parent commits 的 SHAI



Object DAG Example


Trace Git commit (demo)

- 隨便抓一個 Commit 的 SHAI 開始:
 - git cat-file -p a08181bf3 (觀察這個 commit,找出 tree 位置)
 - git cat-file -p ea44d629 (觀察這個 tree,找出任一個 blob SHAL)
 - git cat-file -p d9647d8a (觀察這個 blob 的內容)

Object DAG Model





- Reference 會指向一個 Commit
- tag 不會移動,指向的 commit 都一樣
 - (帶有額外資訊的 tag 內部會用 Object 儲存)
- branch 指向該 branch 最新的 commit
- HEAD 指向 current branch

Reference is cheap!

- 開新 branch 和 tag 只不過是 refs 而已,直 到真的有 commit 前都沒有什麼負擔
- 不像有些 CSV 開分支會複製一份原始碼, 非常耗費資源

Integrity

- SHAI 是內容的 checksum
- Integrity: 如果檔案內容有損毀,就會發現跟
 SHAI不同。如果 tree 改檔名,也會被發現。
- 這在分散式系統非常重要:資料從一個開發者
 傳到另一個開發者時,確保資料沒有被修改

Object DAG Model



Trace Git branch (demo)

- cat .git/HEAD 拿到目前工作目錄 current branch 是指向哪一個 branch
- cat .git/refs/heads/master 拿到 master branch 指向的 commit
- cat .git/refs/tags/foobar 拿到 foobar tag 指向的 commit

Object DAG Full Example



自從用了 Git,就再也不會想用 SVN 了

- svn 開分支和 merge 超痛苦,能避免開 branch 就避免開 branch
- svn 看 log 超痛苦,需要等待網路連線
- svn commit 或 checkout 超慢,主機放國外常
 常 commit 一半中斷

4-1. Git 指令基礎

46

Git is a collection of command-Line utilities

	1. She	1	
New Tab Info Close Execute			
[~/sanabox] (master) Fu	by-1.9.5-pi253 git	maliak	
addall	filton-branch	rellink	
andatt	flow	renoce	
amend	Format-patch	repuck	
ancotate	Feck	request-pull	
analy	TSCK .	Recot	
apply	9	reset	
hisect	ge oet-ton-commit-id	rever c	
blame	get-tur-commtt-tu		
br	grep	173 mb	
branch	help	send-email	
bundla	iman-send	shortlos	
chackaut	init	show	
checkouc	instanab	show-branch	
cherry cherry-nick	1	snow-branch	
cherry-plok		smart-tog	
	Log	smart-merge	
	merge	smart-pull	
clean	mergetool	st	
clone		stage	
CD	name-rev	stash	
column	nb	status	
commit	notes	submodule	
contig	p+	SVN	
CP	pair	tog	
credential-cache	pull	uncommit	
credential-osxkeychain	push	unstage	
credential-store	ra	whatchanged	
describe	rc	MIO	
diff	rebase		
alfittool	reflog		

LIDA-T'A'2-

Install Git

- 安裝 Git
 - Mac 有內建,或用 homebrew 裝最新版 Git
 - <u>http://git-scm.com/downloads</u>

• 安裝 Git GUI

- https://git-scm.com/downloads/guis/
- 推薦 SourceTree (Mac, Windows)
- GitX (Mac)
- gitq, gitk (Linux)
- 如果你之前用 Subversion,不建議再用 TortoiseGit

48

Setup

- git config --global user.name "Your Name"
- git config --global user.email "your@email.com"
- git config --global color.ui true
- 之後也可以直接編輯 ~/.gitconfig

Setup (Windows only)

- 處理換行字元誤認為變更
- git config --global core.autocrlf true
- git config --global core.safecrlf true

從頭建立 Repository (demo)

- mkdir sandbox
- cd sandbox
- git init

第一次 commit (demo)

- touch README
- git add README
- git status
- git commit -m "First Commit"
- stree
 - 打開 GUI,需要先在 SourceTree 中 Install Command Line Tools

修改看看 (demo)

- 編輯 README 做些變更
- git status
- git diff
- git add .
 - 一次加入所有變更
- git status
- git diff --staged

git commit -m "Update README"

三種區域:Working tree/Staging Area/Repository



Working tree 裡的檔案有四種狀態



Why? Staging Area

- Working tree 可能很亂,包含了想要 commit 的內容和不相關的實驗修改等
- Staging area 的設計可以讓你只 commit 想要的檔案,甚至是想要的部份修改而已
- Staging Area 又叫作 Index

只 commit 部分檔案 (demo) ^{可以用 GUI} 幫忙挑選

- touch a.rb
- touch b.rb
- git add a.rb
- git commit "Update a"
- git add b.rb
- git commit "Update b"

staging 之後又再修改內容 (demo)

- 修改 a.rb
- git add a.rb
- 再次修改 a.rb
- git commit -m ''commit a''
 - 這時 commit 的內容是第一次修改當時的內容 而已

只 commit 同一檔案部分內容 (demo)

- git add --patch
 - y 加到 staging
 - n 不要加到 staging
 - s 可以拆小一點 hunk
- 或者用 GUI 例如 gitx 來選取



刪除和搬移檔案 (demo)

- git rm a.rb
- git mv b.rb c.rb
- git add .
- git commit "Remove a, Rename b to c"
- 沒有 copy ? 因為 Git 是追蹤內容(相同內容SHAI相同) 你只要 cp 即可,不用擔心浪費空間。

復原 Revert

- •新增一筆 commit 來做還原
 - 例如本來的 commit 是新增一行,那麼 revert commit 就會移除那一行
- git revert e37c75787
- git revert HEAD



- git tag foo
- git tag foo <SHAI>
- git tag bar -m "some message"
- git tag
- git tag -d foo



- git log
- git log --oneline
- git log --oneline --decorate --graph
- git log 很多參數可以用,但是建議還是用 GUI 吧...

查看程式碼逐行的歷史紀錄

- git blame <filename>
- git blame -L 100,10 <filename> 從第一百行開始顯示10行

比較差異 Diff

- git diff <SHA1> 拿 Working Tree 比較
- git diff <SHAI > <SHAI >
- git diff --stat <SHAT>
- git diff diff --staged 拿 Staging Area 來比較

砍掉 untracked 檔案 (demo)

- git clean -n 列出打算要清除的檔案
- git clean -f 真的清除
- git clean -x 連 .gitignore 裡列的檔案也清掉

忽略不需要追蹤的檔案

- 編輯 gitignore (屬於專案的設定,會 commit 出去)
- 編輯 ~/.gitignore (你個人的 Global 設定)
- 空目錄不會 commit 出去,必要時需要 放 .gitkeep 檔案
- .gitignore 大集合
 <u>https://github.com/github/gitignore</u>



- 不需要 commit 出去?
- tmp/*
- log/*
- 你個人環境的設定檔(例如你偏愛的編輯器設定檔)
- 可以自動產生的檔案
- build/* 等 compile 之後的檔案
- .DS_Store
- Thumbs.rb

Commit 基本原則

https://blog.alphacamp.co/2015/01/13/git/

- 適當的粒度/相關性/獨立性
 - 以一個小功能、小改進或一個 bug fixed 為單位。
 - 對應的 unit test 程式在同一個 commit
 - 無相關的修改不在同一個 commit
 - 語法錯誤的半成品程式不能 commit
- git diff --check 可以檢查多餘的空白
- commit 訊息很重要
 - 第一行寫摘要
 - 有需要寫實作細節的話, 放第二行之後

http://www.freshconsulting.com/atomic-commits/ http://chris.beams.io/posts/git-commit/

70

4-2. Git 指令:開分支

何時開 Branch?

- Topic feature 開發新功能
- Bug fixes
- 重構 (refactor)
- 任何實驗




(master) git branch iss53



(iss53) git commit



(master) git branch hotfix (hotfix) git commit



(master) git merge hotfix



(iss53) git commit



(master) git merge iss53



開 branch (demo)

- git branch new_feature
- git branch 或 git branch <SHAI>
- git checkout new_feature
 - (以上兩步可以合一 git checkout -b new_feature)
- touch new_feature.rb
- git add new_feature.rb
- git commit -m ''Ney

用 checkout 切換 分支,同一時間你的 working tree 只能待在-

合併 branch 回來 (demo)

- git checkout master
- git merge new_feature

Merge CONFLICT 怎麼辦?

- git status 看哪些檔案有 conflict
- 手動編輯這些檔案,解決 <<<<< ====
 >>>>> 有衝突的部分
- 或是 git mergetool 用工具解決
- git reset HEAD --hard 放棄合併

git reset HEAD --hard 超好用,可將 working tree 全部復原

Branch 更名和刪除

- git branch -m old_name new_name
- git branch -M old_name new_name (強制覆蓋)
- git branch new_feature -d
- git branch new_feature -D (強制删除)



- Straight merge 預設的合併模式
 - 保留被合併的 branch commits 記錄,並加上一個 merge commit,看線圖會有兩條 Parents 線。
 - 如果是 fast-forward,就不會有 merge commit。除非加上 --no-ff 參數。

四種合併方式 (cont.)

- Squashed commit
 - git merge new_feature --squash
 - 壓縮成只有一個 merge-commit,不會有被合併的 log。SVN的 merge 即是如此。
- cherry-pick
 - git cherry-pick 714cba
 - 只合併指定的 commit

使用 branch 注意事項

- 切換 working tree 的 branch 時,如果有檔案
 在 staging area 或是 modified,會無法切換。
- 可以先 commit,反正只要不 push 出去,再
 reset 回來即可
- 或是用 stash 指令暫存起來
 - git stash
 - git stash apply
 - git stash clear

Branches 當做存檔(savepoint)

- git branch is cheap!
- 建立 branch 即可保留當時的 Graph
- 就像遊戲存檔

無名 Branch 狀態

- git checkout <branch_name> 前往特定 Branch
- git checkout <SHAI> 則會前往特定節點,此
 時會是 detached HEAD 狀態,一種沒有名字
 的 current branch 狀態
- git checkout -b new_branch 就會給一個

4-3. Git 指令: 遠端操作

89

以 GitHub Repository 為例 建立一個專案

- 產生 SSH Key
 - ssh-keygen -t rsa -C "<u>your_email@youremail.com</u>"
- cat ~/.ssh/id_rsa.pub 至 Github Settings > SSH Keys 把上述key放上去
- 至 Github 開一個專案(不要勾 Initialize this repository with a README)
- git remote add origin git@github.com:your_account/sandbox.git
- git push -u origin master
- 之後只需要 git push
 - 出現 [[rejected] 表示遠端上的commits比較新, 需要先做 git pull 拉下來

SSH Secure Shell

- 取代 telnet 通訊協定,提供加密通訊
- 使用 public-key cryptography 公開金鑰加密 (非對稱加密)

拷貝從一個已經存在的 Repository

- SSH 建議用這個,請在 Github 上設定你的 Public Key
 - git clone git@github.com:ihower/sandbox.git
- HTTPS 速度差,要打 Github 帳號密碼,但較沒有防火牆問題(有的公司會限制 SSH 22 port)
 - git clone <u>https://ihower@github.com/ihower/sandbox.git</u>

Clone 別人的專案

- 盡量使用 SSH 協定
 - git clone git@github.com:ihower/sandbox.git
- 除非有防火牆問題,改用 HTTPS 協定
 - git clone <u>https://github.com/ihower/sandbox.git</u>

Pull 從遠端更新

- git pull origin master 或 git pull
 - 實際作用是先 git fetch 遠端的 branch
 - 然後與本地端的 branch 做 merge , 產生一個 merge commit 節點

Push 將 Commit 送出去

- git push 或 git push -u origin your_branch_name
 - 實際的作用是將本地端的 branch 與遠端的 branch 作 fast-forward 合併
- 如果出現 [rejected] 錯誤的話,表示遠端的 commits 比較新,你必須先作 pull 更新下來。
- 預設不會 push tags 資訊
 - git push --tags

列出和取出遠端 branch

- git fetch (更新遠端資訊回來,但不做合併)
 - 事實上 git pull 就是在做 git fetch 加上 git merge
- git branch -r
- git checkout -t origin/foobar
- git branch -r --merged (列出已經被合併的

删除遠端 Branch

- git push origin :foobar
- git pull 和 git fetch 不會清除已經被刪掉的 branch
 - 請用 git fetch -p
- github 也有提供 branches 列表可以點選刪除

Pull Request (merge request)

- Github 的殺手級功能
- 無論你有沒有權限寫入,你都可以送 Pull
 Request 請求合併你的 branch 分支
- 搭配團隊 Code Review 流程
- ・目前最流行的 Open Source 協作方式
- Github Fork 可以直接複製一整個 repo. 到你

Pull Request 流程練習

- A 加 B 成為 collaborator
- B clone A 的 project、開一個 feature branch, commit 一些東西,然後 push 推上去
- B 在 Github 介面上送 Pull Request 給A
- A pull 下來,進行合併 merge,然後關 PR, 並且砍掉被合併的 branch

4-4. Git 指令: 還沒 push 前可以 做的事

100

還沒 push 分享出去的 branch,你可以...

- 不同於 SVN, Git 的 Commit 其實還在本地端,所以可以修改 commit logs!!
 - 修改 commit 訊息,甚至內容
 - 合併 commits
 - 打散 commit 成多個 commits
 - 調換 commits 順序

reset (還原到特定節點)

- 跟 revert 不同, reset 是直接砍掉 commits
 歷史記錄
- git reset e37c75787
- git reset HEAD^ (留著修改在 working tree)
- git reset HEAD[^] --soft (修改放到 staging area)
- git reset HEAD^ --hard (完全清除)

快速修正上一個 Commit

- 上一個commit有typo,想要重新commit
 - git reset HEAD^
 - 修正
 - git commit -am ''message''
- 可以簡化成
 - 修正
 - git add .
 - git commit --amend

rebase (重新 commit 一遍)

• git rebase -i e37c7578

pick 048b59e first commit pick 995dbb3 change something pick aa3e16e changed

```
# Rebase 0072886..1b6475f onto 0072886
#
# Commands:
  p, pick = use commit
#
  r, reword = use commit, but edit the commit message
#
  e, edit = use commit, but stop for amending
#
  s, squash = use commit, but meld into previous commit
#
#
  f, fixup = like "squash", but discard this commit's log message
  x, exec = run command (the rest of the line) using shell
#
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
```

rebase is dangerous!! It rewrite history!

But I like it!

105

第四種合併方式: rebase

- rebase 調整分支點 (僅適用於 local branch)
 - I. 從要被合併的 branch 頭重新分支
 - 2. 將目前 branch 的 commits 記錄一筆一筆重新 apply/patch 上去
 - 等於砍掉重練 commit log,僅適合還沒分享給 別人的 local branch。

一個範例:開發中的 new_feature 想要合併 master 主幹的變更



如果是 merge (new_feature) git merge master


如果用 rebase,就沒有 G 點! (new_feature) git rebase master



rebase notes

- 注意到 D`和 E`和本來的 D 和 E commits
 已經被視為不同 commit, SHAI也不同了
- rebase 過程中,每一次 apply 都有可能發生 conflict,跟 merge 只發生一次不同。
- 發生 conflict 怎麼辦? 可以選擇修好後 continue, 忽略 skip 或放棄 abort

- 記得一定要離開 rebase 的 context 情境

已經 push 分享出去的 commits 記錄,請千萬不要再修改記錄 push 出去!!



rebase + merge 的 龜毛 合併法

- 原因
 - feature branch 很亂,不時 merge 與主幹同步
 - feature branch 有 typo,commit 訊息想改
 - feature branch 有些 commits 想合併或拆開
- 作法
 - 先在 feature branch 做 git rebase develop -i
 - (反覆整理直到滿意) git rebase 分岔點 -i

超級乾淨,每一次的 merge commit 就代表一個功能完成

Subject		
O develop origin/HEAD origin/develop		
A Marge branen eature Merce	e) into develop	
ф с		
beautiful-clean-nice.c	commit/histrov/	
0		
O Mar i nch lfaatura	Vinte develop	
o Merge (ch reature)	into develop	
Dianci		
Q Merge branch 'feature/	into develop	
ÒC		
0 C		- 6
9 -		
0 C		A
		4
		3

Demo (before merge)



Demo (normal merge)



15

Demo (rebase + merge)





Rebase demo screencast

<u>http://ihowertw/blog/archives/6704</u>

注意事項(I)

- 必須要加 --no-ff 才會有 merge commit。不 然會是 fast-forward。
- rebase 之後的 feature branch 就不要再 push 出去了
- 如果有遠端的 feature branch,合併完也砍 掉

注意事項 (2)

- 不求一次 rebase 到完美,不然中間的 conflict 會搞混。
- 可以一次改點東西就 rebase 一次,然後開個 臨時的 branch 存檔起來,再繼續 rebase 自己 直到滿意為止。





- git archive
- git bisect
- git grep
- git show
- git gc
- git format-patch, send-email, am

使用 git pull --rebase 可以避免 無調的 merge 節點,讓線圖變乾淨

Merge branch 'master' into Merge branch 'master' into Merge branch 'master' into Merge branch 'master' in Merge branch 'master' Merge branch 'master' Merge branch 'master' of gli Merge branch 'master' of e

特別是很多人在同 一個 branch 同時開發,例 如主幹 master,容易發

git pull 何時用 merge? 何時用 rebase?

• 修改比較多,預期會有 conflict,用 merge - 是不是一開始就應該開 feature branch 來做呢? ·修改範圍較小,不太預期有 conflict, 建議 可以用 --rebase 但是如果採用 git flow 利用 branches,就可 以大幅降低無謂 merge

Bash Prompt

- 顯示目前在哪一個 branch
- <u>https://gist.github.com/ihower/</u> <u>dec41671245eceff56d1</u>

000)		1. Shell				
New Tab	Info 🗄 Customize	Close : Exec	ute			_	
<u>Å</u>	Shell		Shell				
[~/rail	s/jobs.ruby.tw	(master)	1821h14m	\$ g	itx	\sim	-
[~/rail	ls/jobs.ruby.tw	(master)	1821h14m	\$			-
							2 1
							.
						0	

bash-completion

- <u>http://bash-completion.alioth.debian.org/</u>
- Install via Homebrew (Mac only)
 - brew install bash-completion
 - cp /usr/local/etc/bash_completion.d/gitcompletion.bash ~/.git-bash-completion.sh
 - 編輯 ~/.bash_profile 加入 [-f ~/.git-bash-completion.sh] && . ~/.git-bashcompletion.sh

Useful alias 編輯 ~/.gitconfig

[alias]

```
co = checkout
  ci = commit
  st = status
  br = branch - v
  rt = reset --hard
  unstage = reset HEAD
  uncommit = reset --soft HEAD^
  l = log --pretty=oneline --abbrev-commit --graph
decorate
  amend = commit --amend
  who = shortlog -n - s - -no - merges
  g = grep - n - color - E
```

- cp = cherry-pick -x
- nb = checkout -b

建立沒有 parent 的獨立 branch

- 建立毫不相關的分支,用來保存其他資訊, 就像目錄一樣。不需要被 merge 回主幹。
- 用途範例: Github page
- git checkout --orphan new_branch
 - git rm -rf . # 砍掉重練
 - 加新檔案
 - git add .
 - git commit -m ''new branch''

Orphan branch example



28

弄丟 commit 和 branch 怎麼辦?

http://www.programblings.com/2008/06/07/the-illustrated-guide-to-recovering-lost-commits-with-git/

- 例如搞砸了 git reset, rebase 或誤刪 branch
- 在 git gc 前都還有機會救回來 (預設會保留孤兒 commits 90天)
- git reflog 或 git fsck --lost-found
- git cherry-pick <SHAT>
- 或 git merge <SHAI>

git-filter-branch 過濾

• <u>只取特定目錄</u>

http://gugod.org/2012/07/split-a-git-repository/

• 删除特定敏感檔案

https://help.github.com/articles/removesensitive-data

Git Submodule 或 Subtree

- Git 不像 SVN 可以只取出其中的子目錄
- 如何將第三方程式碼加進來? 像 SVN Externals?
- Submodule
 - <u>http://josephjiang.com/entry.php?id=342</u>
 - <u>http://josephjiang.com/entry.php?id=357</u>
- Subtree
 - http://git-scm.com/book/ch6-7.html
- •我沒用到
 - 因為 Ruby/Rails Develops 用不太到 XD

- Ruby 用 Bundler 工具來管理 dependencies,可以確保大家都跑相同的套件版本,又可以解決套件相依性問題

git svn

- git 可以吃 SVN repository
- git svn clone <u>http://url/svn/trunk/</u> project_name
- (normal git operations)
- git svn dcommit

自行架設 Git Server

- git --bare via SSH
- Gitosis
- Gitolite
- GitLab
 - <u>https://about.gitlab.com</u>

6. 第三方服務

134



= github SOCIAL CODE HOSTING

誰在用 GitHub?

- twitter
- facebook
- rackspace
- digg
- Yahoo!
- shopify
- EMI
- six apart

- jQuery
- YUI 3
- mootools
- Ruby on Rails
- node.js
- symfony
- mongodb
- Erlang



define	_attr_method correctly defines methods with invalid identifiers \bigtriangledown	commit	c834a751d2acbd55b580
1	about 16 hours ago	parent	fda45f4fc493f5596375

rails /

r	name	age	message	history
E	actionmailer/	March 10, 2011	Fix typo [spastorino]	
🖬 (actionpack/	about 22 hours ago	fixes an issue with number_to_human when conver [joshk]	
i	activemodel/	about 16 hours ago	define_attr_method correctly defines methods wi [spastorino]	
i	activerecord/	2 days ago	Merge branch 'master' of git://github.com/lifo/ [fxn]	
a (activeresource/	March 05, 2011	Active Resource typos. [rtlechow]	
a (activesupport/	4 days ago	Revert "It should be possible to use ActiveSupp [josevalim]	
	bin/	June 19, 2010	add missing shebang to rails bin. LH [#4885 sta [smtlaissezfaire]	

Ruby is the #2 most popular language on GitHub Repositories Languages Timeline Search Tips Explore Ruby Recently Created Recently Updated All Languages Most Watched Today Most Forked Today ActionScript 🌉 wbailey / kata mxcl / homebrew Ada Injekt / slop thoughtbot / capybara-webkit Arc apneadiving / Google-Maps-for-Rails c opscode / cookbooks ASP 🗠 codegram / resort edavis10 / redmine Assembly shuber / attr_encrypted drbrain / meme Boo С Most Watched This Week Most Forked This Week C# mxcl / homebrew C++ injekt / slop mxcl / homebrew Clojure 🚺 rails / rails diaspora / diaspora CoffeeScript I rails / rails NoamB / sorcery chicagoruby / booksiebot7000 ColdFusion diaspora / diaspora 🔤 drbrain / meme Common Lisp D Delphi Most Watched This Month Most Forked This Month Duby postrank-labs / goliath mxcl / homebrew Eiffel mxcl / homebrew I rails / rails Emacs Lisp 🖄 diaspora / diaspora I rails / rails Erlang mojombo / jekyll meskyanichi / backup F# diaspora / diaspora plataformatec / devise Factor FORTRAN Most Watched Overall Most Forked Overall Go Groovy I rails / rails mxcl / homebrew Haskell S mxcl / homebrew I rails / rails HaXe diaspora / diaspora lo joshuaclayton / blueprint-css thoughtbot / paperclip plataformatec / devise Shopify / active_merchant Java JavaScript

Fork network

The rails network graph

All branches in the network using rails/rails as the reference point. Read our blog post about how it works.

Show Help



如何送 Patch 給開源專案?

- Fork 成我的遠端 Repository
- Clone
- Fix bugs 或 Enhance features
- Push
- •送 Pull request 給原作者,請他合併

Fork & Pull Requests

有 75% 的被 fork 開源專案,有 outside merge 回來的記錄!!

被 watch 超過一人以上的開源專案,有 25% 有 outside contributions 紀錄!!

Pull Requests – gugod/App-perlbrew – GitHub			
• • •	https://github.com/gugod/App-perlbrew/	rpulls 🔤 🖒 (Qr Google	
iGoogle 🔠	Read Today Read Later Instapaper I	Reader Blog Twitter Facebook Plurk GitHub Tumblr Yahoo! WebTV Redmine(O) Redmine(T) 担罪(G) 字典(Y)	
	aithub	interest in the second	
		Evolves Gibbob - Gist - Blog - Help - O O- Search	
		Capacitation and the O Capacitation	
	gugod / App-peribrew		
	Source Commits Network	Pull Requests (1) Issues (15) Wiki (2) Graphs Branch: moster	
	All Requests 1	1 open request Keyboard shortouts available	
	Maure	Read Open Closed Popularity	
	Tours	Use Devel::PatchPerl to make old perls build	
		Old perl releases have the tendency to not build anymore on modern platforms and require some pat	
	Find a user	Traf submitted to gugod/App-perbrew 2 days ago Updated about 3 hours ago @ 2 comments	
	raf 1	documents -j and enables parallel testing CLOSED	
	trojr 0	parallel testing needs TEST_JOBS=N for "make test_harness"	
	tokuhirom 0	dagolden submitted to gugod/App-perbrew March 12, 2011 Updated March 13, 2011 e [®] 1 comment	
	melo 0	Fix switch and use CLOSED	
	dagolden 0	With version 0.16 of 'perlbrew', the 'switch' command doesn't work as specified. The bashrc code	
	book 0	melo submitted to gugod/App-peribrew March 08, 2011 Updated March 10, 2011	
	avar 0	trivial spelling fix CLOSED	
	franckcurry 0	No description available	
	abh 0	abh submitted to gugod/App-perbrew March 06, 2011 Updated March 06, 2011 of 1 comment	
	doy 0	Topic/exec brew only	
	gfx 0	Hi I've added an option to exec: -brew-only This way, I can install a module using 'perlbr	
	polettix 0	Tranckcuny submitted to gugod/App-perbrew February 27, 2011 Updated March 01, 2011 🔊 3 comments	
	yibe 0	Lise star on Solaria	
	dalinaum 0	Use gui un ovinna etoseb	





The problem is it's in the unicorn for a reason. Unicorn offers zero-downtime restarts. When you send it USR2 it forks and reloads rails/rack and then starts responding to users requests.

When Gemfile is changed it doesn't pick up changes. For example, I have working unicorn instance, then I add a gem, I deploy, send USR2, and unicorn couldn't restart. Because it loaded old Gemfile and have no idea about new gem. Practically it dumps this backtrace (https://gist.github.com/5c27d25d134e1cfa0eae) to the unicorn.stderr.log and continues to run old instance with the new current directory.

I propose Gem.refresh should reload Gemfile. What do you guys think?

elucid says:

I don't think this is actually a problem with Bundler. I've put up a simple repo at https://github.com/elucid/unicorn-reload to demonstrate how USR2 reloads with Unicorn work. If the old Gemfile is being reloaded after an upgrade, I think this probably has to do with the way that Unicorn was started in the first place. In particular, new Unicorn master processes will be started in the exact same directory that the original was. If you update your app code in place and nothing else changes, that should be fine. However if each deploy has its own directory you could'nd pointing at an old one. You can manually set this to something stable (e.g. a symlink that gets updated on each release) by calling working_directory somedir in your Unicorn config file.
🥥 rails / r	Unwatch & Fork @ 7,862 & 1	1,352
Source	Commits Network Pull Requests (22) Graphs Tree: c834	a75
Switch Brand	ches (12) v Switch Tags (83) v Comments Contributors	
Ruby on Ra http://rubyor	alls Downloads	B
нттр	Git Read-Only https://github.com/rails/rails.git	
define_at	ttr_method correctly defines methods with invalid identifiers astorino (author) out 16 hours ago commit c834a751d2acbd55b5i tree 423e181f132e2977oc1 parent fdo45f4fc493f559632	80 12 75
Showing 2 ch	hanged files with 6 additions and 5 deletions.	
active	model/lib/active_model/attribute_methods.rb 5	× •
active	model/test/cases/attribute_methods_test.rb 6	× 0
) active	emodel/lib/active_model/attribute_methods.rb 🕑 show inline notes View file @ c834	4a75
108 108	emodel/lib/active_model/attribute_methods.rb endel/lib/active_model/attribute_methods.rb else	4a75
m active 108 108 109 109	emodel/lib/active_model/attribute_methods.rb Show hime notes View file © c834 (0 -108,9 +108,8 00 module ActiveHodel else # use eval instead of a block to work around a memory leak in dev	4a75
C 2	<pre>emodel/lib/active_model/attribute_methods.rb @ show kille notes View file @ c834 else # use eval instead of a block to work around a memory leak in dev @ about 3 hours ago @ </pre>	4a75
activ activ 108 109 109 2	emodel/lib/active_model/attribute_methods.rb # of a block to work around a memory leak in dev # use eval instead of a block to work around a memory leak in dev # of a block to work around a memory leak in dev # of a block to cleanup comment? # of a block to cleanup comment? # of a block to clean	4a75
active active 108 109 109 2	emodel/lib/active_model/attribute_methods.rb emodel/lib/active_model/attribute_methods.rb for -108,9 +108,8 ff module Active#odel else # use eval instead of a block to work around a memory leak in dev forgot to cleanup comment? spastorino recorceab	4a75
active active 108 109 109 2	emodel/lib/active_model/attribute_methods.rb show inline notes View file @ c834 (@ -108,9 +108,8 (@ module ActiveModel else	viev
C 2	emodel/lib/active_model/attribute_methods.rb encodel/lib/active_model/attribute_methods.rb else	⊷75 view
<pre>netive continue continue</pre>	<pre>emodel/lib/octive_model/attribute_methods.rb @ show inline notes View file @ c834 @ -108,9 +108,8 @@ module ActiveRodel else # use eval instead of a block to work around a memory leak in dev Forgot to cleanup comment? Forgot to cleanup comment? forgot to cleanup comment? forgot to cleanup comment? dmitry, hey thanks for pointing me to this. I saw this yeah I have to try a few more things and do another com Add a line note # mode in fogi </pre>	view
<pre>netive netive neti</pre>	emodel/lib/active_model/attribute_methods.rb enodel/lib/active_model/attribute_methods.rb else # use eval instead of a block to work around a memory leak in dev forgot to cleanup comment? Forgot to cleanup comment? forgot to cleanup comment? Add a line note # mode in fcgi = sing.class_eval <<=orb, _FILE_, _LINE_+1	∙a75 viev
activ activ 108 108 109 109 2 2	<pre>emodel/lib/active_model/attribute_methods.rb</pre>	view
activ activ 108 109 109 2 2 110 111 112 113 111	<pre>emodel/lib/active_model/attribute_methods.rb else # use eval instead of a block to work around a memory leak in dev forgot to cleanup comment? forgot to cleanup comment? forgot to cleanup comment? dmitry, hey thanks for pointing me to this. I saw this yeah I have to try a few more things and do another com forgot a line note f mode in fogi sing.class_eval <<-eorb,FILE_,LINE + 1 def # (name); #(value.nil? ? 'nil' ; value.to_s.inspect); end eorb value = value.nil? ? 'nil' ; value.to_s </pre>	view
active active 108 108 109 109 2 2 110 111 112 113 111 11 11 11 11 11 11 11 11 11 11 1	<pre>emodel/lib/active_model/attribute_methods.rb if online notes forgot to cleanup comment? forgot to cleanup comment to this. I saw this yeah I have to try a few more things and do another com forgot to cleanup comment to this. I saw this yeah I have to try a few more things and do another com forgot to cleanup comment to this. I saw this yeah I have to try a few more things and do another com forgot to cleanup comment to this. I saw this yeah I have to try a few more things and do another com forgot to cleanup comment to this. I saw this yeah I have to try a few more things and do another com forgot to cleanup comment to this.</pre>	view
active active 108 108 109 109 2 2 110 111 112 113 111 112 113 111 112 114 113 115 114	<pre>emodel/lib/active_model/attribute_methods.rb @ show inline notes View file @ c834 # =108,9 +108,8 ## module ActiveModel else # use eval instead of a block to work around a memory leak in dev forgot to cleanup comment? spastorino mococode dmitry, hey thanks for pointing me to this. I saw this yeah I have to try a few more things and do another com forgot to cleanup comment? Add a line note # mode in fogi sing.class_eval <<-eorb,FILE_, _LINE_+1 def #(name); #(value.nil? ? 'nil' : value.to_s.inspect); end eorb value = value.nil? ? 'nil' : value.to_s sing.send(:define_method, name) { value } end end</pre>	view

Github pages http://pages.github.com/

- 叫做 your.github.com 的 repository,Github 會 自動建立 http://your.github.com 的靜態網頁。
- 擁有 gh-pages 的分支的 your_project, Github 會自動建立 http://your.github.com/your_project 靜熊網頁。

在 Git,你可以建立毫不相關的分支, 用來保存其他資訊,就像目錄一樣。 (開 Branch 的用途不一定就是要 merge)

46



Enterprise-grade at startup speed.

Unleash your potential while maintaining control.

සි Open collaboration

Encounging code review and transparency creates opportunities for developers to improve code quality while gaining new skills throughout the development process.

GitHub Flow

Whether you are fixing a typo or working on the next big thing, take advantage of GitHub Flow to keep you focused on code, not tools.

Social coding

Keeping up with everything is easy with GitHub. Follow projects and conversations, reach out with a quick @mention and check out profile pages to see what everyone has been up to.

GitHub is your code portfolio!

48



Unlimited CI Runners to do parallel testing





 Scott Chacon 公開的 OmniGraffle diagram 圖檔 <u>https://github.com/schacon/git-presentations</u>



參考書籍和影片

- <u>http://pragprog.com/screencasts/v-jwsceasy/</u> <u>source-control-made-easy</u>
- https://peepcode.com/products/git
- https://peepcode.com/products/advanced-git
- Git Internals, Peepcode
- Pragmatic Version Control Using Git, Pragmatic
- Pragmatic Guide to Git, Pragmatic
- Continuous Delivery Ch. 14

Version Control with Git, O'Reilly



- <u>http://ihower.tw/blog/category/git</u>
- http://nfarina.com/post/9868516270/git-is-simpler
- http://tomayko.com/writings/the-thing-about-git
- <u>http://think-like-a-git.net</u>
- http://yanpritzker.com/git-book/
- <u>http://progit.org/book/</u>
- <u>http://git-scm.com</u>
- <u>http://help.github.com/</u>
- <u>http://gitimmersion.com</u>
- <u>http://www.gitready.com/</u>
- http://gitref.org
- http://try.github.com