

大牛總結的 Git 使用技巧，寫得太好了



🕒 2019年09月28日 13:45:09 👁 1560 views 📄

摘要

前言本文是參考廖雪峰老師的Git資料再加上我自己對Git的理解，記錄我的Git學習歷程，作下此文是為以後學習，工作，開發中如果遇到問題可以回過頭來參考參考。因為水平有限，難免會有出錯的地方，歡迎指正。

前言



本文是參考廖雪峰老師的Git資料再加上我自己對Git的理解，記錄我的Git學習歷程，作下此文是為以後學習，工作，開發中如果遇到問題可以回過頭來參考參考。因為水平有限，難免會有出錯的地方，歡迎指正。

Git是什麼

官方話：Git是一個免費的開源分布式版本控制系統，旨在快速高效地處理從小型到大型項目的所有事務。

引用廖雪峰老師的話，它能自動幫我記錄每次文件的改動，還可以讓同事協作編輯，這樣就不用自己管理一堆類似的文件了，也不需要把文件傳來傳去。如果想查看某次改動，只需要在軟件里瞄一眼就可以。

為什麼要學習Git

- 面試要被問。可以應付面試。
- 很多公司開發都用Git來處理項目。現在不學，以後肯定還要學。
- 在我看來Git是現如今所有程序員都要掌握的，以後與同事共同開發項目必定要用到的，熟練掌握Git命令，可以提高開發的效率。

安裝Git

Windows

直接在官網上去下載。下載完成後，隨便在某個文件下右鍵如果有Git Bash Here就安裝成功。安裝後，還要在命令行輸入

```
$git config --global user.name "你的名字"
$git config --global user.email "你的郵箱"
```

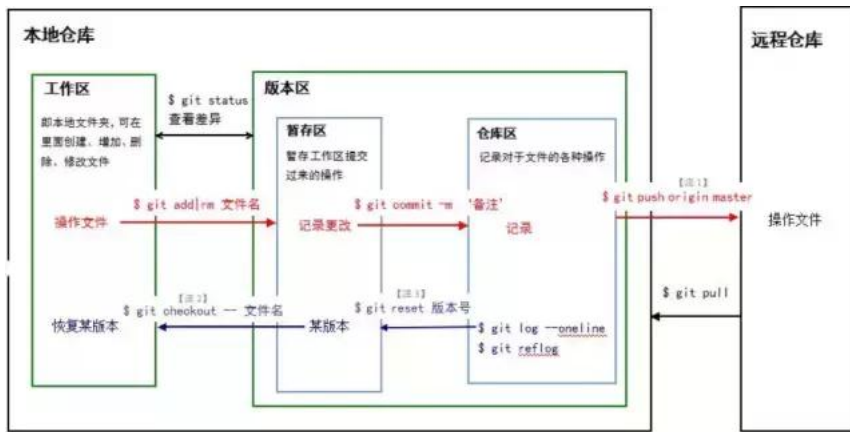
global表示全局，這台機器所有的Git倉庫都會使用這個配置。允許單個倉庫使用其他的名字和郵箱。

Mac

Mac也可以像Windows一樣，按上面的步驟安裝。

也可以直接從AppStore安裝Xcode，Xcode集成了Git，不過默認沒有安裝，你需要運行Xcode，選擇菜單“Xcode”->“Preferences”，在彈出窗口中找到“Downloads”，選擇“Command Line Tools”，點“Install”就可以完成安裝了。

倉庫



本地倉庫是對於遠程倉庫而言的。本地倉庫 = 工作區 + 版本區。

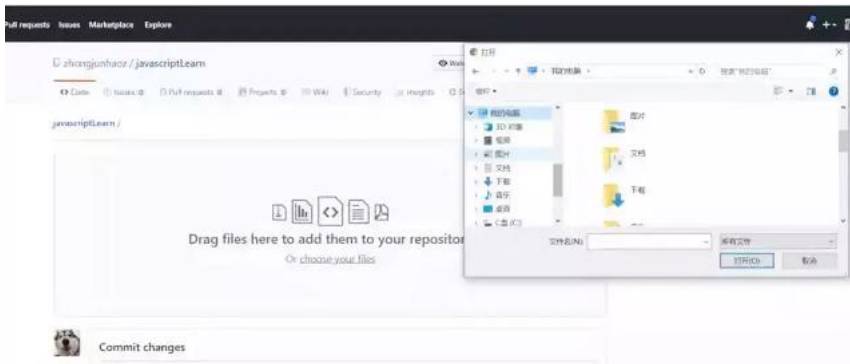
- 工作區即磁盤上的文件集合。
- 版本區(版本庫)即.git文件。
- 版本庫 = 暫存區(stage) + 分支(master) + 指針Head。

以我使用最頻繁的git命令為例，即提交到github為例。

- `git init` 原本本地倉庫只包含着工作區，這是最常見的工作狀態。此時，`git init`一下，表示在本地區域創建了一個.git文件,版本區建立。
- `git add` . 表示把工作區的所有文件全部提交到版本區裡面的暫存區
- 當然你也可以通過 `git add ./xxx/` 一條一條分批添加到暫存區。
- `git commit -m "xxx"` 把暫存區的所有文件提交到倉庫區，暫存區空空蕩蕩。
- `git remote add origin https://github.com/name/name_cangku.git` 把本地倉庫與遠程倉庫連接起來。
- `git push -u origin master` 把倉庫區的文件提交到遠程倉庫里。
- 一旦提交後，如果你又沒有對工作區做任何修改，那麼工作區就是“乾淨”的。會有這樣的信息nothing to commit, working tree clean

提交到GitHub

以前不熟悉git命令的時候，我提交項目到github上都是直接在網頁上直接拉取文件提交上去的。有點羞恥。



- `git init` .初始化，表示把這個文件變成Git可以管理的倉庫。初始化後打開隱藏的文件可以看到有一個.git文件。
- `git add` . 後面的一個點表示把這個文件全部提交到暫存區。
- `git add ./readme.md/` 表示把這個文件下面的readme.md文件提交到暫存區。
- `git commit -m "你要評論一點什麼東西"` `git commit`的意思是把暫存區的全部文件提交到本地倉庫。-m後接評論。
- `git remote add origin https://github.com/name/name_cangku.git`表示把你本地的倉庫與GitHub上的遠程倉庫連接起來。只需要連接一次，以後提交的時候就可以不用謝這條命令了。name是你的github名字，name_cangku是你的倉庫名。注意不要把後面的.git給漏掉了。因為我前面就是這麼走過來的，繞了很多彎路。至於如何在GitHub上新建倉庫，網上有許多教程，這裡不再贅述了。
- `git push -u origin master` 把本地倉庫提交到遠程倉庫。(最後一步)在你的遠程倉庫上刷新一下就可以看到你提交的文件了。
- 最後提到的是，在`git commit -m ""`之前，可以重複`git add`到暫存區。但是`git commit`會把你之前存放在暫存區的全部文件一次性全部提交到本地倉庫。

版本的回溯與前進

提交一個文件，有時候我們會提交很多次，在提交歷史中，這樣就產生了不同的版本。每次提交，Git會把他們串成一條時間線。如何回溯到我們提交的上一個版本，用`git reset --hard + 版本號`即可。版本號可以用`git log`來查看，每一次的版本都會產生不一樣的版本號。

回溯之後，`git log`查看一下發現離我們最近的那個版本已經不見了。但是我還想要前進到最近的版本應該如何？只要`git reset --hard + 版本號`就行。退一步來講，雖然我們可以通過`git reset --hard + 版本號`,靠記住版本號來可以在不同的版本之間來回穿梭。

但是,有時候把版本號弄丟了怎麼辦？`git reflog`幫你記錄了每一次的命令，這樣就可以找到版本號了，這樣你又可以通過`git reset`來版本穿梭了。

撤銷

場景1：在工作區時，你修改了一個東西，你想撤銷修改，git checkout -- file。廖雪峰老師指出撤銷修改就回到和版本庫一模一樣的狀態，即用版本庫里的版本替換工作區的版本。

場景2：你修改了一個內容，並且已經git add到暫存區了。想撤銷怎麼辦？回溯版本，git reset --hard + 版本號,再git checkout -- file,替換工作區的版本。

場景3：你修改了一個內容，並且已經git commit到了master。跟場景2一樣，版本回溯，再進行撤銷。

刪除


如果你git add一個文件到暫存區，然後在工作區又把文件刪除了，Git會知道你刪除了文件。如果你要把版本庫里的文件刪除，git rm 並且git commit -m "xxx".

如果你誤刪了工作區的文件，怎麼辦？使用撤銷命令，git checkout --就可以。這再次證明了撤銷命令其實就是用版本庫里的版本替換工作區的版本，無論工作區是修改還是刪除，都可以“一鍵還原”。

分支

分支，就像平行宇宙，廖雪峰老師如是說。你創建了一個屬於你自己的分支，別人看不到，還繼續在原來的分支上正常工作，而你在自己的分支上幹活，想提交就提交，直到開發完畢後，再一次性合併到原來的分支上，這樣，既安全，又不影響別人工作。用 Git 和 Github 提高效率的 10 個技巧！這篇也推薦看下。

創建與合併分支


在沒有其他分支插進來時，只有一個master主分支。每次你git push -u origin master 提交就是增加一條時間軸，master也會跟着移動。


創建一個other的分支，通過other提交，雖然時間軸向前走了，但是主分支master還在原來的位置。


理論分析完，看一下命令怎麼寫。


創建分支other,切換到other分支。

```
git branch other
git checkout other
```

查看當前所有分支

```
git branch
```

```
* other
master
```

當前的分支會有一個*

用other提交

```
git add ./xxx/
git commit -m "xxx"
```

other分支完成，切換回master

```
git checkout master
```

此時，master分支上並沒有other的文件，因為分支還沒有合併。

合併分支

```
git merge other
```

合併完成之後，就可以在master分支上查看到文件了。

刪除other分支

```
git branch -d other
```

我由此想到，在以後工作中，應該是一個開放小組共同開發一個項目，組長會創建很多分支，每一個分支可以交給一個人去開發某一個功能，一個小組共同開發而且不會相互干擾。誰的功能完成了，可以由組長合併一下完成了的分支。哦，完美！

解決合併分支問題



假如有這樣一種情況，分支other已經commit了，但是此時指針指回master時，並且master沒有合併，而是git add / commit 提交了。這樣，就產生了衝突，主分支master文件內容與other分支的內容不一樣。合併不起來！所以，

- 修改文件的內容，讓其保持一致。
- git add git commit 提交。
- 分支合併了。



-
- git log --graph 查看分支合併圖
- git branch -d other 刪除分支，任務結束。

分支管理策略

git merge --no-ff other 禁用Fast forward模式，因為使用Fast forward模式，刪除分支後，分支歷史信息會丟失。超詳細的 Git 實戰教程，傻瓜一看也會！這篇也推薦看下。

BUG分支

廖雪峰老師提到，工作中每個bug都可以通過一個新的臨時分支來修復，修復後，合併分支，然後將臨時分支刪除。但如果你手上有分支在工作中，你的上級要你改另外的分支的BUG。

你要把現在正在工作的分支保存下來，git stash,把當前工作現場“存儲”起來，等以後恢復後繼續工作。當你解決BUG後，git checkout other回到自己的分支。用git stash list查看你剛剛“存放”起來的工作去哪裡了。

此時你要恢復工作：

- git stash apply恢復卻不刪除stash內容，git stash drop刪除stash內容。
- git stash pop恢復的同時把stash內容也刪了。
- 此時，用git stash list查看，看不到任何stash 內容。

總結：修復bug時，我們會通過創建新的bug分支進行修復，然後合併，最後刪除；當手頭工作沒有完成時，先把工作現場git stash一下，然後去修復bug，修復後，再git stash pop，回到工作現場

刪除分支

- git branch -d + 分支有可能會刪除失敗，因為Git會保護沒有被合併的分支。
- git branch -D + 分支 強行刪除，丟棄沒被合併的分支。

多人協作

- git remote 查看遠程庫的信息，會顯示origin，遠程倉庫默認名稱為origin
- git remote -v顯示更詳細的信息
- git push -u origin master推送master分支到origin遠程倉庫。
- git push -u origin other 推送other到origin遠程倉庫。

抓取分支



產生上圖的衝突時，

- git pull 把最新的提交從遠程倉庫中抓取下來，在本地合併，解決衝突。在進行git pull
- 如果git pull 也失敗了，還要指定分支之間的鏈接，這一步Git會提醒你怎麼做。然後再git pull。

廖雪峰老師的總結：多人協作的工作模式通常是這樣：

- 首先，可以試圖用git push origin

- 推送自己的修改；
- 如果推送失敗，則因為遠程分支比你的本地更新，需要先用git pull試圖合併；
- 如果合併有衝突，則解決衝突，並在本地提交；
- 沒有衝突或者解決掉衝突後，再用git push origin
- 推送就能成功！
- 如果git pull提示no tracking information，則說明本地分支和遠程分支的鏈接關係沒有創建，用命令git branch --set-upstream-to origin/。

Rebase

git rebase 把分叉的提交歷史“整理”成一條直線，看上去更直觀。缺點是本地的分叉提交已經被修改過了。

最後在進行git push -u origin master

rebase的目的是使得我們在查看歷史提交的變化時更容易，因為分叉的提交需要三方對比。

標籤管理

比如一個APP要上線，通常在版本庫中打一個標籤(tag),這樣，就確定了打標籤的版本。將來無論什麼時候，取某個標籤的版本，就是把那個打標籤的時刻的歷史版本取出來。所以，標籤也是版本庫的一個快照。

Git的標籤雖然是版本庫的快照，但其實它就是指向某個commit的指針。

tag其實就是一個讓人容易記住的有意義的名字，它跟某個commit綁在一起。比如tag v2.1就是把歷史上的一個版本的東西叫做v2.1

創建標籤

步驟：

- git branch查看當前分支,git checkout master切換到master分支。
- git tag <name> 打標籤，默認為HEAD。比如git tag v1.0
- 默認標籤是打在最新提交的commit上的。如果想要打標籤在以前的commit上，要git log找到歷史提交的commit id.
- 如果一個commt id是du2n2d9,執行git tag v1.0 du2n2d9就把這個版本打上了v1.0的標籤了。
- git tag 查看所有標籤，可以知道歷史版本的tag
- 標籤不是按時間順序列出，而是按字母排序的。
- git show <tagname> 查看標籤信息。
- git tag -a <標籤名> -m "<說明>",創建帶說明的標籤。-a指定標籤名，-m指定說明文字。用show可以查看說明。

操作標籤

- git tag -d v1.0 刪除標籤。因為創建的標籤都只存儲在本地，不會自動推送到遠程。所以，打錯的標籤可以在本地安全刪除。
- git push origin <tagname> 推送某個標籤到遠程
- git push origin --tags 一次性推送全部尚未推送到遠程的本地標籤
- 如果標籤推送到遠程。git tag -d v1.0 先刪除本地標籤v1.0。git push origin :refs/tags/v1.0刪除遠程標籤v1.0

自定義Git

- git config --global color.ui true讓Git顯示顏色，會讓命令輸出看起來更醒目
- 忽略特殊文件 創建一個.gitignore文件，把需要忽略的文件名填進去。Git就會自動忽略這些文件。我也在學習中遇到過這樣的問題，比如node_modules文件就可以忽略。

忽略文件原則：忽略操作系統自動生成的文件，比如縮略圖等；忽略編譯生成的中間文件、可執行文件等，也就是如果一個文件是通過另一個文件自動生成的，那自動生成的文件就沒必要放進版本庫，比如Java編譯產生的.class文件；忽略你自己的帶有敏感信息的配置文件，比如存放口令的配置文件。

- 強制提交已忽略的文件。git add -f <file>
- git check-ignore -v <file>檢查為什麼Git會忽略該文件。
- 給Git命令配別名,這個有點騷，就是你以後想輸入git rebase時,你給它一個“外號”，就叫它git nb。以後你可以通過git nb來代替git rebase。

常用Git命令總結

- git config --global user.name "你的名字" 讓你全部的Git倉庫綁定你的名字
- git config --global user.email "你的郵箱" 讓你全部的Git倉庫綁定你的郵箱
- git init 初始化你的倉庫
- git add . 把工作區的文件全部提交到暫存區
- git add ./<file>/ 把工作區的<file>文件提交到暫存區
- git commit -m "xxx" 把暫存區的所有文件提交到倉庫區，暫存區空空蕩蕩
- git remote add origin https://github.com/name/name_cangku.git 把本地倉庫與遠程倉庫連接起來
- git push -u origin master 把倉庫區的主分支master提交到遠程倉庫里
- git push -u origin <其他分支> 把其他分支提交到遠程倉庫
- git status查看當前倉庫的狀態
- git diff 查看文件修改的具體內容
- git log 顯示從最近到最遠的提交歷史

- git clone + 倉庫地址下載克隆文件
- git reset --hard + 版本號 回溯版本，版本號在commit的時候與master跟隨在一起
- git reflog 顯示命令歷史
- git checkout -- <file> 撤銷命令，用版本庫里的文件替換掉工作區的文件。我覺得就像是Git世界的ctrl + z
- git rm 刪除版本庫的文件
- git branch 查看當前所有分支
- git branch <分支名字> 創建分支
- git checkout <分支名字> 切換到分支
- git merge <分支名字> 合併分支
- git branch -d <分支名字> 刪除分支,有可能會刪除失敗，因為Git會保護沒有被合併的分支
- git branch -D + <分支名字> 強行刪除，丟棄沒被合併的分支
- git log --graph 查看分支合併圖
- git merge --no-ff <分支名字> 合併分支的時候禁用Fast forward模式,因為這個模式會丟失分支歷史信息
- git stash 當有其他任務插進來時，把當前工作現場“存儲”起來,以後恢復後繼續工作
- git stash list 查看你剛剛“存放”起來的工作去哪裡了
- git stash apply 恢復卻不刪除stash內容
- git stash drop 刪除stash內容
- git stash pop 恢復的同時把stash內容也刪了
- git remote 查看遠程庫的信息，會顯示origin，遠程倉庫默認名稱為origin
- git remote -v 顯示更詳細的信息
- git pull 把最新的提交從遠程倉庫中抓取下來，在本地合併,和git push相反
- git rebase 把分叉的提交歷史“整理”成一條直線，看上去更直觀
- git tag 查看所有標籤，可以知道歷史版本的tag
- git tag <name> 打標籤，默認為HEAD。比如git tag v1.0
- git tag <tagName> <版本號> 把版本號打上標籤，版本號就是commit時，跟在旁邊的一串字母數字
- git show <tagName> 查看標籤信息
- git tag -a <tagName> -m "<說明>" 創建帶說明的標籤。-a指定標籤名，-m指定說明文字
- git tag -d <tagName> 刪除標籤
- git push origin <tagname> 推送某個標籤到遠程
- git push origin --tags 一次性推送全部尚未推送到遠程的本地標籤
- git push origin :refs/tags/<tagname> 刪除遠程標籤<tagname>
- git config --global color.ui true 讓Git顯示顏色，會讓命令輸出看起來更醒目
- git add -f <file> 強制提交已忽略的文件
- git check-ignore -v <file> 檢查為什麼Git會忽略該文件

喜歡的點個關注，一起學習探討新技術。

[WhatsApp](#)[LINE](#)[Facebook](#)[Twitter](#)[更多...](#)

頻道熱榜~

1 面向“2021”的5G新機，到底誰才是當之無愧的旗艦守門員？ 熱

2 買小米手機的“四不碰”選擇，還是高配低價最容易中招，千萬記住 爆

3 李江濤講故事，蔣凡婚姻結局 爆

4 每天給妻子轉5萬，160天後他跳樓自殺：這個女人太狠了 爆

5 第二代微雲台來了，國產手機技術更新速度令人矚目 爆

6 昨日的因，今日的果，說說馬雲這些年“犯下的錯” 熱

2021年3月汽車銷量排行出爐，別克GL8銷量超1.8萬輛

3月SUV銷量十五強榜，長安CS75不敵哈弗H6，博越未上榜？

大眾新車扎堆上新！兩款SUV車長超5m 就等上市了

你能認出來嗎？這居然是雪鐵龍C5

WEY以重任，長城將於5月上新 “轎車陣容”

續航1000km的車來了！廣汽發大招，還有8分鐘充滿的電池

捷途X70 PLUS諸葛版，品價比更高了

2021房車市場分析：符合大眾需求的房車產品增多

9.28萬起，比亞迪第二代宋Pro上市，6款車型怎麼選更值？

領克03獲得大眾青睞月度銷量為5284輛

國產知名車企“倒了”？一年虧損112億，創始人早已套現瀟灑離場

新車 | 起售價25.28萬元，新款極星2正式上市，增單電機前驅版

實時熱門搜索 🕒 8分鐘前更新

1 琳妲男友 熱

2 小周周 新

3 袁詠儀10歲兒 爆

4 宥勝認錯道歉 爆

5 竹野內豐 流星

6 一日三餐 熱

7 鍾欣凌部落格

8 模範出租車線上看 爆

9 林依晨真面目

10 琳妲 王柏融

11 哈利王子 禿頭 爆

12 一日三餐漁村篇線上看 熱

每日要聞

陸劇吧

