# GIT 备忘单

Git 是免费开源的分布式版本控制系统，负责 GitHub 的一切

相关的发生在您的计算机上。 此备忘单具有最重要和最常见的功能

使用 Git 命令以方便参考。

## 安装和图形用户界面

对于 Git 的特定平台安装程序，GitHub 还提供了轻松了解最新版本的命令线工具，同时为日常提供图形用户界面交互、审查和存储库同步。

适用于 Windows 的 GitHub

http://windows.github.com

Mac 版 GitHub

http://mac.github.com

对于 Linux 和 Solaris 平台，最新版本可在

Git 官方网站。

适用于所有平台的 Git

http://git-scm.com

## 设置

配置跨所有本地存储库使用的用户信息

### 设置一个在查看版本历史时可识别的名称

git config --global user.name “[firstname lastname]”

### 设置将与每个历史标记关联的电子邮件地址

git config --global user.email “[valid-email]”

### 为 Git 设置自动命令行着色以便于查看

git config --global color.ui auto

## 阶段和快照

使用快照和 Git 暂存区

### 在工作目录中显示修改后的文件，为您的下一次提交准备

git status

### 丢弃工作目录中的文件

git restore [file]

### 将现在看起来的文件添加到您的下一次提交（阶段）

git add [file]

### 取消暂存文件，同时保留工作目录中的更改

git reset [file] || git restore --staged [file]

### 已更改但未上演的内容的差异

git diff

### 已上演但尚未提交的内容的差异

git diff --staged

### 将您的暂存内容作为新的提交快照提交

git commit -m “[descriptive message]”

## 设置和初始化

### 配置用户信息、初始化和克隆存储库

git config --global user.name “your name”

git config --global user.email “yourt email”

### 将现有目录初始化为 Git 存储库

git init

### 通过 URL 从托管位置检索整个存储库

git clone [url]

## 分支与合并

隔离分支中的工作，更改上下文并集成更改

### 列出你的分支机构。 一个 \* 将出现在当前活动的分支旁边

git branch

### 在当前提交处创建一个新分支

git branch [branch-name]

### 切换到另一个分支并将其检出到您的工作目录中

git checkout

### 将指定分支的历史合并到当前分支中

git merge [branch]

### 显示当前分支历史中的所有提交

git log

### 检查日志、差异和对象信息

NSPECT & 比较

### 显示当前活动分支的提交历史

git log -a

### 显示不在分支上的分支上的提交

git log branchB..branchA

### 显示更改文件的提交，即使跨重命名

git log --follow [file]

### （\*）显示不在分支中的分支中的差异

git diff branchB...branchA

### （\*）以人类可读的格式显示 Git 中的任何对象, 全面展示某个提交的全部内容

git show [SHA] SHA=>commit : id

## 分享和更新

从另一个存储库检索更新并更新本地存储库

### 添加一个 git URL 作为别名

git remote add [alias] [url]

### 从那个 Git 远程获取所有分支

git fetch [alias]

### 查看远程所有分支

git branch --all

### 将远程分支合并到您当前的分支中以使其保持最新状态

git merge [alias]/[ branch]

### 将本地分支提交传输到远程存储库分支

git push [alias] [branch]

### 从跟踪远程分支获取并合并任何提交

git pull

## 跟踪路径变化

版本控制文件删除和路径更改

### 从项目中删除文件并暂存删除以进行提交

git rm [file]

### 从项目中删除文件夹并暂存删除以进行提交

git rm -r -cached [file]

### 更改现有文件路径并暂存移动

git mv [现有路径] [新路径]

### （\*）显示所有提交日志，并指出任何移动的路径

git log --stat -M

## 改写历史

重写分支、更新提交和清除历史记录

### 在指定分支之前应用当前分支的任何提交

git rebase [branch]

### 清除暂存区，从特定提交重写工作树

git reset --hard [commit]

## (\*)忽略模式

防止无意的暂存或提交文件

### 使用任一直接字符串将具有所需模式的文件另存为 .gitignore匹配或通配符

logs/ \*.notes pattern\*/

### 所有本地存储库的系统范围忽略模式

git config --global core.excludesfile [file]

## (\*)临时提交

临时存储修改过的跟踪文件以更改分支

### 保存修改和分阶段的更改

git stash

### 列出隐藏文件更改的堆栈顺序

git stash list

### 从存储堆栈顶部写入工作

git stash pop

### 丢弃存储堆栈顶部的更改

git stash drop

# 更换电脑后如何同步github上原有的项目？

git init 建立本次存储库

git remote add origin github(ssh或https) 建立远程连接

git fetch origin 获取远程更新

git merge origin master 把更新内容合并到本地分支

精简版：git pull origin master 第一次git push 如果报错rejected便执行一次此命令

# 如何修改commit注释?

## 修改最新提交的注释

git commit --amend

i进入编辑模式

esc 推出编辑

输入:eq保存并推出

## 修改历史多条提交注释

1. git log 看下需要修改的提交注释
2. git rebase -i HEAD~2
3. 数字2是从最旧的注释开始计算两个注释进行修改
4. 同上进行操作, 需要修改哪条注释就把注释前面的pick换成edit
5. git commit --amend git rebase --continue
6. 每修改一个注释就要执行一次上面的两个命令, 直到修改结束
7. amend代表执行修改注释, 编辑修改注释内容
8. continue 代表保存当前注释, 并继续下一个注释

4）git push --force origin master 强制提交注释修改结果, 覆盖旧的注释结果

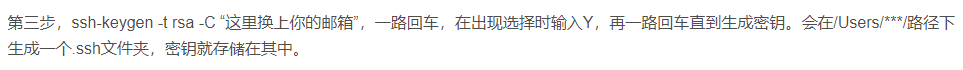
# github如何删除tags



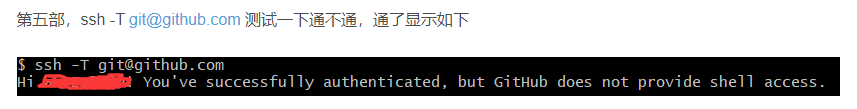
# 如何创建账号并且连接github











# 如何修改github远程地址？

1. git remote -v 查看git远程地址
2. git remote rm origin 删除远程地址
3. git remote add origin master

## 删除github远程地址

git remote rm origin

## 添加github远程地址 [fetch,push]两个地址

git remote add origin URL

## 修改github远程地址

git remote set-url origin URL

git remote set-url --push URL

## 如何使用SSH同时连接github和gitlab?

https://juejin.cn/post/6844903586262941710