GIT CHEAT SHEET V 1.1.1 

**Git cheat sheet(Git备忘录汉化版)** 翻译：huxiaosc

Git是开源的分布式版本控制系统，有助于你在笔记本电脑或台式电脑上更好的使用GitHub. 该备忘录总结了常用的Git命令行说明，以便快速参考.

**安装 GIT**

GitHub提供的桌面客户端，包括一个提供了所有通用仓库指令的用户图形界面和Git 高阶脚本的自动更新命令行版本.

**GitHub for Windows**

https://windows.github.com

**GitHub for Mac**

https://mac.github.com

Linux 和 POSIX 系统可用的Git分发版本请查阅Git SCM 官方网站。

**Git for All Platforms**

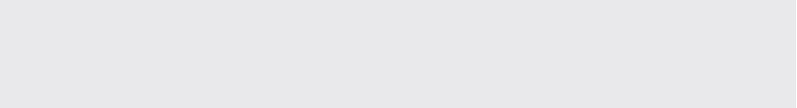
http://git-scm.com

**配置工具**

为所有本地库配置用户信息



**$ git config --global user.name "[name]"**

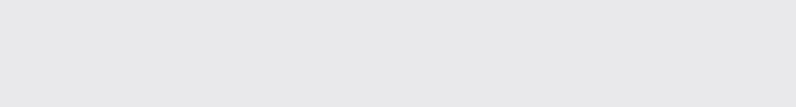


**变更处理**

检查修改和提交记录

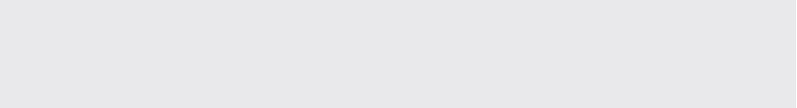


**$ git status**



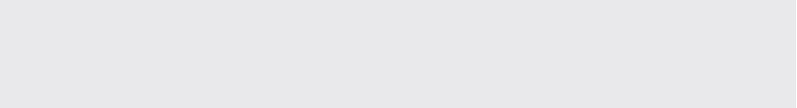
列出准备提交的所有新增或者修改的文件。

**$ git diff**



比较工作目录和暂存区域快照之间的差异

**$ git add [file]**



将文件的快照信息添加到索引库

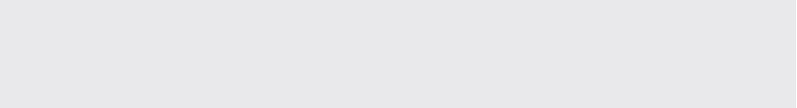
**$ git diff --staged**



比较暂存区域快照和版本库之间的差异



**$ git reset [file]**



将当前的分支重设到最后一次提交

**$ git commit -m "[descriptive message]"**



永久记录文件快照到版本库历史

设置你提交处理时的签署名

**$ git config --global user.email "[email address]"**



设置你提交处理时的邮箱名



**$ git config --global color.ui auto**



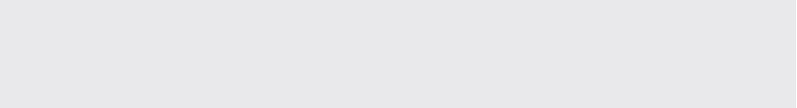
启用命令行输出的辅助色

**创建仓库**

开始一个新的仓库或者从已有仓库拷贝



**$ git init [project-name]**



使用指定的名称创建一个本地仓库

**$ git clone [url]**

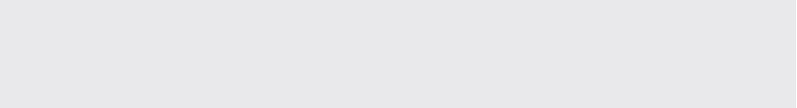


**批量变更**

Name a series of commits and combine completed eﬀorts

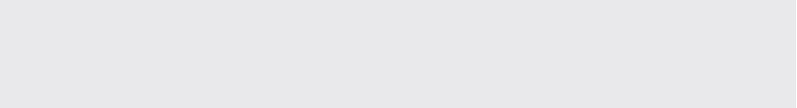


**$ git branch**



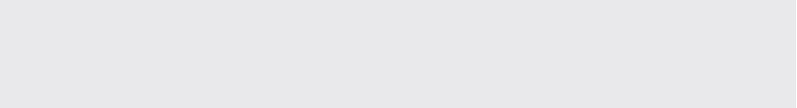
列出当前仓库的所有分支

**$ git branch [branch-name]**



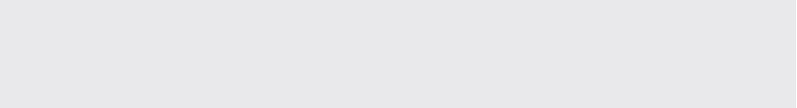
创建一个新的分支

**$ git checkout [branch-name]**



切换到指定的分支并更新工作目录

**$ git merge [branch]**



合并指定的分支版本到当前分支

**$ git branch -d [branch-name]**



下载一个项目和它的所有版本历史信息 删除指定分支

GIT CHEAT SHEET

**删除文件**

移动和删除版本文件



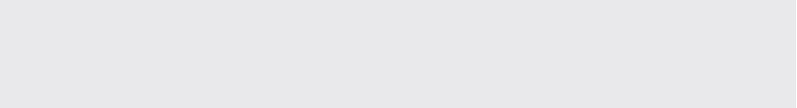
**$ git rm [file]**



从 当前工作目录删除文件，并且从仓库管理系统中移除。



**$ git rm --cached [file]**



从本地索引库中移除文件，但不删除本地文件。

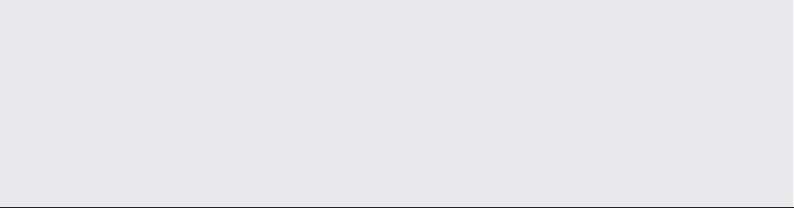
**$ git mv [file-original] [file-renamed]**



修改文件名并添加到本地索引目录

**SUPPRESS TRACKING**

包含临时文件和目录



**\*.log**

**build/**

**temp-\***

A text file named .gitignore 通过指定的表达式过滤忽略文件和目录



**$** **git ls-files --other --ignored --exclude-standard**



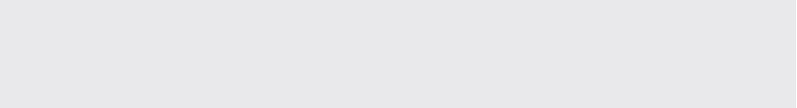
列出项目中所有需要忽略的文件

**历史记录查看**

浏览并检查项目文件的变化

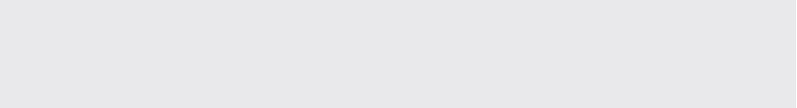


**$ git log**



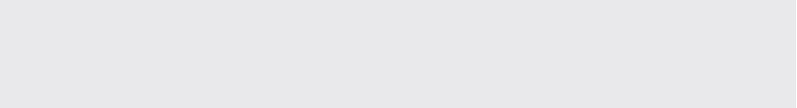
列出当前分支的版本历史

**$ git log --follow [file]**



列出一个文件的版本历史列表

**$** **git diff [first-branch]...[second-branch]**



显示文件差异，默认显示当前工作区和暂存区的版本差异

Git diff HEAD 显示当前工作区和HEAD版本的差异

**$** **git show [commit]**



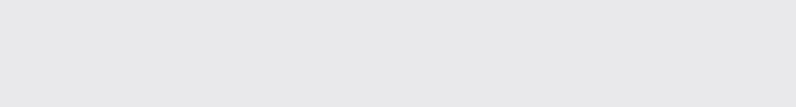
显示指定 commit 修改的内容。默认最近一次的

**回滚提交**

清除错误和还原历史



**$ git reset [commit]**



将当前的工作区内容重设到指定 commit，默认最后一次提交

**$** **git reset --hard [commit]**



回滚到指定的commit，毁尸灭迹

**保存片段**

搁置和恢复未完成的修改



**$** **git stash**



临时存储所有修改的文件



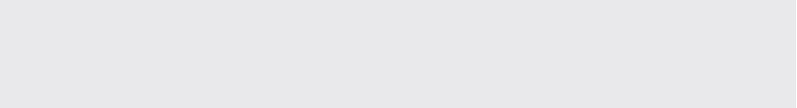
**$ git stash pop**



从最近的贮藏区恢复



**$ git stash list**



列出所有贮藏列表

**$ git stash drop**



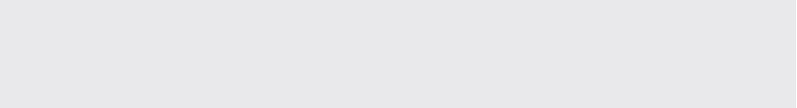
清除最近的贮藏列表

**同步变化**

Register a repository bookmark and exchange version history

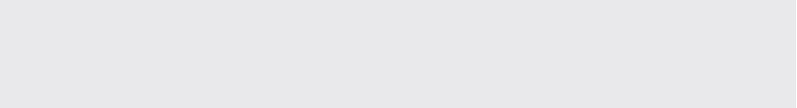


**$ git fetch [bookmark]**



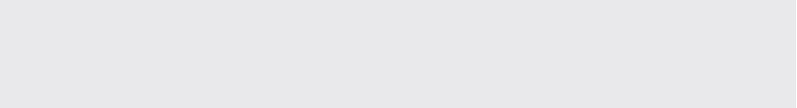
从仓库目录下载所有代码到本地仓库，不会自动merge

**$ git merge [bookmark]/[branch]**



合并分支目录到当前的本地分支版本

**$ git push [alias] [branch]**



上传所有的本次分支提交记录到 GitHub

**$ git pull**



从仓库目录下载所有代码到本地仓库并且自动merge

**远端仓库**

管理远程仓库列表



**$ git remote -v**



显示远端仓库，并显示对应的地址



**$ git remote add [shortname] [url]：**

添加一个新的远程仓库，可以指定一个简单的名字

