创建GIT仓库

创建一个新的仓库

cd *project_dir*

git init

git add .

复制一个已经创建的仓库

git clone *existing_dir new_dir*

git clone *git://github.com/user/repo.git*

git clone *http://github.com/user/repo.git*

本地修改

显示工作路径下全部已修改的文档

git status

显示与上次提交版本文件的不同

git diff

把当前所有修改添加到下次提交中

git add .

指定某个文件的修改添加到下次提交中

git add -p <file>

提交本地所有修改，-m选项提供注释

git commit -a -m "My messages"

修改上一次提交

git commit --amend

撤销文件的修改

git checkout --file

提交历史

从最新提交开始显示所有提交记录

git log

显示指定文件的所有修改

git log -p <file>

谁，在什么时间，修改了文件的什么内容

git blame <file>

分支与标签

显示所有分支

git branch -a

查看远程分支列表

git branch -r

切换当前分支

git checkout <branch>

基于当前分支创建新分支

git branch <new-branch>

将远程分支迁到本地

git checkout branch_name origin/branch_name

将远程分支迁到本地并且切换到该分支

git checkout -b branch_name origin/branch_name

删除本地分支

git branch -d <branch>

给当前的提交打标签

git tag <tag-name>

更新与发布

列出当前配置的远程线

git remote -v

显示远程端信息

git remote show <remove>

添加新的远程端

git remote add <shortname> <url>

下载远程端的所有改动到本地 *不会自动合并到当前*

git fetch <remote>

下载远程端的所有改动到本地 *自动合并到当前*

git pull <remote> <branch>

删除远程端分支

git branch -dr <remote/branch>

将本地版本发布到远程端

git push <remote/branch>

发布标签

git push --tags

合并与重置

将分支合并到当前

git merge <branch>

将当前版本重置到分支中 *请勿重置已发布的提交！*

git rebase <branch>

退出重置

git rebase --abort

解决冲突后继续重置

git rebase --continue

使用配置好的合并工具去解决冲突

git mergetool

在编辑器中手动解决冲突后，标记文件为已解决冲突

git add <resolved-file>

git rm <resolved-file>

撤销

放弃工作目录下的所有修改

git reset --hard HEAD

放弃某个文件的所有本地修改

git checkout HEAD <file>

重置一个提交 *通过创建一个截然不同的新提交*

git revert <commit>

从HEAD重置到上一次提交的版本，并将之后修改标记为未添加到缓存区的修改

git reset <commit>

将HEAD重置到上一次提交的版本，并保留未提交的本期修改

git reset --keep <commit>

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com