本文整理了一些常用的 Git 操作，老司机可以温故知新，新手可以点赞收藏。文末提供了入门教程及学习资源，请自行下滑 ~

操作

全局

```
git config --global user.name '你的名字'
git config --global user.email '你的邮箱'
```

当前仓库

```
git config --local user.name '你的名字'
git config --local user.email '你的邮箱'
```

查 global

```
git config --global --list
```

查 当前仓库

```
git config --local --list
```

删 global

```
git config --unset --global 要删除的配置项
```

删 当前仓库

```
git config --unset --local 要删除的配置项
```

本地操作

查　变更　况

```
git status
```

将当前　录及其子　录下所有变更　加入到暂存区

```
git add .
```

将仓库内所有变更　加入到暂存区

```
git add -A
```

将指定文件　加到暂存区

```
git add 文件1 文件2 文件3
```

比　工作区和暂存区　所有差异

```
git diff
```

比　某文件工作区和暂存区　差异

```
git diff 文件
```

比　暂存区和 HEAD　所有差异

```
git diff --cached
```

比　某文件暂存区和 HEAD　差异

```
git diff --cached 文件
```

比　某文件工作区和 HEAD　差异

```
git diff HEAD 文件
```

创建 commit

```
git commit
```

将工作区指定文件　复成和暂存区

```
git checkout 文件1 文件2 文件3
```

将暂存区指定文件　复成和 HEAD

```
git reset 文件1 文件2 文件3
```

将暂存区和工作区所有文件　复成和 HEAD　样

```
git reset --hard
```

difftool 比　任意两个 commit　差异

`git difftool 提交1 提交2`

查　哪些文件　　Git　控

`git ls-files --others`

将未处　完　变更先保存到 stash 中

`git stash`

临时任务处　完后　　之前　工作

- pop 不保留 stash
- apply 保留 stash

`git stash pop`

`git stash apply`

查　所有 stash

`git stash list`

取回某次 stash　变更

`git stash pop stash@{数字n}`

优　修改最后　次 commit

```
git add.
git commit --amend
```

分支操作

查　当前工作分支及本地分支

`git branch -v`

查　本地和　　分支

`git branch -av`

查　　分支

`git branch -rv`

切换到指定分支

`git checkout 指定分支`

基于当前分支创建新分支

`git branch 新分支`

基于指定分支创建新分支

```
git branch 新分支 指定分支
```

基于某个 commit 创建分支

```
git branch 新分支 某个 commit 的 id
```

创建并切换到　分支

```
git checkout -b 新分支
```

安全删　本地某分支

```
git branch -d 要删除的分支
```

强　删　本地某分支

```
git branch -D 要删除的分支
```

删　已合并到 master 分支　所有本地分支

```
git branch --merged master | grep -v '^\*\| master' | xargs -n 1 git branch -d
```

删　　origin 已不存在　所有本地分支

```
git remote prune origin
```

将 A 分支合入到当前分支中且为 merge 创建 commit

```
git merge A分支
```

将 A 分支合入到 B 分支中且为 merge 创建 commit

```
git merge A分支 B分支
```

将当前分支基于 B 分支做 rebase，以便将B分支合入到当前分支

```
git rebase B分支
```

将 A 分支基于 B 分支做 rebase，以便将 B 分支合入到 A 分支

```
git rebase B分支 A分支
```

变更历史

当前分支各个 commit　　显

```
git log --oneline
```

显　就　　n 个 commit

```
git log -n
```

　图　显　所有分支　历史

```
git log --oneline --graph --all
```

查　　及到某文件变更　所有 commit

```
git log 文件
```

某文件各　最后修改对应　 commit 以及作

```
git blame 文件
```

标　操作

查　已有标

```
git tag
```

新建标

```
git tag v1.0
```

新建带备　标

```
git tag -a v1.0 -m '前端食堂'
```

　指定　 commit 打标

```
git tag v1.0 commitid
```

推　　个本地标

```
git push origin v1.0
```

推　全　未推　　本地标

```
git push origin --tags
```

删　　个本地标

```
git tag -d v1.0
```

删　　个　　标

```
git push origin :refs/tags/v1.0
```

　　交互

查　所有　　仓库

```
git remote -v
```

　加　　仓库

```
git remote add url
```

删　　仓库

```
git remote remove remote的名称
```

命名　　仓库

`git remote rename 旧名称 新名称`

将　　所有分支和标　　变更　拉到本地

`git fetch remote`

把　　分支　变更拉到本地，且 merge 到本地分支

`git pull origin 分支名`

将本地分支 push 到

`git push origin 分支名`

删　　分支

`git push remote --delete 远端分支名`

`git push remote :远端分支名`