

Git 命令清单

Workspace: 工作区
Index / Stage: 暂存区
Repository: 仓库区 (或本地仓库)
Remote: 远程仓库
master: 默认开发主分支
origin: 默认远程版本库
Head: 默认开发分支

一、新建代码库

\$ git init	# 在当前目录新建一个Git代码库
\$ git init [project-name]	# 新建一个目录，将其初始化为Git代码库
\$ git clone [url]	# 下载一个项目和它的整个代码历史

二、配置

\$ git config --list	# 显示当前的Git配置
\$ git config [--global] user.name "[name]"	
\$ git config [--global] user.email "[email address]"	

三、增加/删除

\$ git add [file1] [file2] ...	# 添加指定文件到暂存区
\$ git add [dir]	# 添加指定目录到暂存区，包括子目录
\$ git add .	# 添加当前目录的所有文件到暂存区
\$ git rm [file1] [file2] ...	# 删除工作区文件，并且将这次删除放入暂存区
\$ git rm --cached [file]	# 停止追踪指定文件，但该文件会保留在工作区
\$ git mv [old] [new]	# 改名文件，并且将这个改名放入暂存区

四、代码提交

```
$ git commit -m [message]           # 提交暂存区到仓库区
$ git commit -a                     # 提交工作区自上次commit之后的变化，直接到仓库区
$ git commit -v                     # 提交时显示所有diff信息
```

五、分支

```
$ git branch                         # 列出所有本地分支
$ git branch -r                     # 列出所有远程分支
$ git branch -a                     # 列出所有本地分支和远程分支
$ git branch [branch-name]          # 新建一个分支，但依然停留在当前分支
$ git checkout -b [branch]           # 新建一个分支，并切换到该分支
$ git checkout [branch-name]         # 切换到指定分支，并更新工作区
$ git merge [branch]                 # 合并指定分支到当前分支
$ git rebase [branch]                # 衍合指定分支到当前分支
$ git branch -d [branch-name]        # 删除分支
$ git push origin --delete [branch-name] # 删除远程分支
$ git branch -dr [remote/branch]     # 删除远程分支
```

六、标签

```
$ git tag                           # 列出所有tag
$ git tag [tag]                      # 基于最新提交新建一个tag
$ git tag -d [tag]                   # 删除本地tag
$ git push origin :refs/tags/[tagName] # 删除远程tag
```

七、查看提交信息和状态信息

```
$ git status                         # 显示有变更的文件
$ git diff                           # 显示暂存区和工作区的差异
$ git log                            # 显示当前分支的版本历史
$ git log --stat                     # 显示commit历史，以及每次commit发生变更的文件
$ git log -p [file]                 # 显示指定文件相关的每一次diff
$ git blame [file]                  # 显示指定文件是谁人在什么时间修改过
$ git diff HEAD                     # 显示工作区与当前分支最新commit之间的差异
```

```
$ git diff --shortstat "@{0 day ago}" # 显示今天你写了多少行代码
$ git reflog                          # 显示当前分支的最近几次提交
```

八、远程操作

```
$ git fetch [remote]                # 从远程仓库获取代码
$ git remote -v                     # 显示所有远程仓库
$ git remote show [remote]          # 显示某个远程仓库的信息
$ git remote add [remote] [url]     # 添加远程版本库
$ git pull [remote] [branch]        # 下载代码并与本地分支合并
$ git push [remote] [branch]        # 上传本地指定分支到远程仓库
$ git push [remote] --force          # 强行本地覆盖远程，即便有冲突
$ git push [remote] --all            # 推送所有分支到远程仓库
```

九、撤销

```
$ git checkout [file]               # 恢复暂存区的指定文件到工作区
$ git checkout HEAD [file]          # 撤销指定的未提交文件的修改内容
$ git checkout .                     # 恢复暂存区的所有文件到工作区
$ git reset --hard HEAD              # 重置暂存区与工作区，与上一次commit保持一致
$ git revert [commit]                # 撤销指定的提交
$ git stash                          # 暂时将未提交的变化移除
$ git stash pop                      # 稍后移入
```