

## Git 速查卡

### Git 配置

```
git config --global user.name <name>
设置作者名称
git config --global user.email <email>
设置作者电邮
git config --system core.editor <editor>
设置文本编辑器
git config --global color.ui auto
设置命令行自动着色
git config --global alias
为 Git 命令设置别名
```

### Git 基础

```
git init <dir>    在 dir 目录创建空的仓库
git init          将当前目录初始化为仓库
git clone <repo>  将 repo 仓库克隆到本机
git add <dir>     暂存 dir 目录中的所有更改
git add .         暂存当前目录中的所有更改
git add <file>    暂存 file 文件
git add -p <file> 暂存 file 中的部分更改
git commit        提交暂存快照
git commit -m "<msg>" 同上, 且通过 msg 指定提交消息
git commit -a     提交已跟踪的更改文件
git status        显示仓库状态
git log           显示整个提交历史
git diff          显示未暂存的更改
git diff --staged 显示已暂存的更改
git rm <file>     删除 file 文件
git mv <src> <dest> 将文件从 src 移到 dest
git format-patch origin 准备补丁
git am -3 <patch> 应用补丁
git grep "<pattern>" 搜索工作目录
```

### Git 差异

```
git diff HEAD      显示工作目录与上次提交之间的差异
git diff --cached  显示暂存更改与上次提交之间的差异
git diff commit1 commit2 显示两次提交之间的差异
git diff branch1 branch2 显示两个分支之间的差异
```

### 撤销更改

```
git revert <commit> 撤销 commit 所做的更改
git reset <file>    从暂存区移除 file
git restore <file>  放弃对 file 所做的更改
git clean -n        显示哪些文件已从工作目录移除
```

### Git 重置

```
git reset          重置暂存区
git reset --hard   重置暂存区和工作目录
git reset <commit> 将当前分支回移到 commit, 重置暂存区
git reset --hard <commit> 同上, 且重置暂存区和工作目录
```

### 重写 Git 历史

```
git commit --amend 替换上一次提交
git rebase <base>  将当前分支 rebase 到 base
git reflog         显示变更记录
```

### Git 分支

```
git branch          列出仓库的分支
git branch -av      列出所有存在的分支
git checkout <branch> 切换到 branch 分支
git branch <branch> 创建新的分支
git checkout -b <branch> 创建并检出名为 branch 的新分支
git branch -d <branch> 删除本地分支
git branch -dr <remote/branch> 删除远端分支
git merge <branch> 将 branch 合并到当前分支
git cherry-pick <commit> 将其它分支的 commit 合并到当前分支
git tag <tag>       给当前提交打标签
```

### 远端仓库

```
git remote -v      列出所有远端
git remote show <remote> 显示有关远端的信息
git remote add <name> <url> 添加新的远端仓库
git fetch <remote> <branch> 抓取指定的 branch 分支
git pull <remote>    抓取当前分支的远端拷贝, 并立即合并
git pull --rebase <remote> 同上, 但使用 rebase 代替合并
git push <remote> <branch> 将 branch 推送到 remote
git push <remote> all 推送本地所有的分支
git push <remote> --tags 推送本地所有的标签
git push <remote> --force 强制推送, 慎用
```

### Git 提交历史

```
git log -<limit>    限制显示提交数
git log --oneline    将每个提交以单行显示
git log -p           显示每个提交的完整 diff
git log --stat       显示 diffstat
git log --author="<pattern>" 搜索特定作者的提交
git log --grep="<pattern>" 搜索匹配 pattern 消息的提交
git log <since>..<until> 显示在 since 和 until 之间的提交
git log -- <file>    只显示具有特定文件的提交
git log --graph --decorate 绘制基于文本的提交图
git blame <file>     显示谁更改了 file 文件
git show <commit>    以人类可读的格式显示任意 Git 对象
git show <commit:file> 同上, 且针对特定文件
```

### 临时提交

```
git stash          保存修改并暂存更改
git stash list     列出储存的文件更改
git stash pop      从存储栈的顶端写入工作目录
git stash drop     放弃存储栈顶端的更改
```

### 忽略文件

对于要忽略的文件可加入 .gitignore, 支持通配符。

```
/logs             忽略 logs 目录
*.swp             忽略所有扩展名为 swp 的文件
```