

Git

- 将一个文件夹变为Git仓库: `git init`
- 查看文件情况: `git status`
- 追踪文件更新: `git add`
- 添加所有修改: `git add .`
- 查看修改内容: `git diff`
- 清除文件追踪: `git reset`
- 创建提交者名字: `git config --global user.name "xxx"`
- 创建提交者邮箱: `git config --global user.email "xxx"`
- 修改commit编辑器: `git config --global core.editor vim`
- 提交: `git commit -m "xxx"` 或者省略 `-m` 在vim模式下写
- 忽略某些文件: `.gitignore` 中放需要忽略的文件的名字
- 停止追踪某一文件: `git rm --cached xxx`
- 创建分支: `git branch xxx` 或 `git checkout -b xxx`
- 切换分支: `git checkout xxx`
- 合并分支: `git merge 另一分支名字`
- 删除分支: `git branch -d xxx` (`-D` 强制删除)
- 删除远程分支: `git push origin --delete`
- 记住用户名密码: `git config credential.helper store`
- 复制仓库: `git clone +网址`
- 更新本地: `git pull` 或 `git fetch --all`
- 移出文件: `git rm`
- 停止追踪: `git rm --cached`
- 重命名文件: `git mv file_from file_to`
- 查看提交历史: `git log (-p)`
- 查看提交历史简略信息: `git log --stat`
- 提交暂存区文件然后修改提交信息: `git commit --amend`
- 取消暂存: `git reset HEAD <file>...`
- 将远程仓库名字从A改为B: `git remote rename A B`

Commit 格式

Lazygit

基本操作

- 左右键移动面板
- `h/j/k/l/方向键` 移动面板
- `?` 查看帮助
- `a` stage/unstage全部文件
- `空格` stage/unstage一个文件
- `回车` stage一部分
- `tab` 在stage和unstage之间跳转
- 在stage面板按 `d` 进行unstage
- 在file面板 `c` 进行commit
- `A` 将本次修改内容放入上次commit中
- `c` 进入vim写commit
- `d` 删除更改
- `D` 查看更多放弃更改的选项
- `esc` 退会到上一层面板
- `q` 退出

分支面板

- 分支面板 `n` 创建新分支
- 分支面板 `空格` 切换分支
- `s` 隐藏更改
- 在储藏面板 `g` 显示隐藏更改
- `M` 将一个分支的更改合并到另一分支（先切换到要接受合并的分支，然后选择需要更改的分支，M）
- `[]` 在面板标签之间跳转

提交面板

- `回车` 进入查看哪些内容被更改
- `d` 恢复一个文件的更改
- `c` 滚回旧版本
- `,.` 翻页
- `<>` 回到顶端 回到底部
- `/` 搜索 -> 支持 `vim` 快捷键
- `r/R` 重命名更改信息，若已经 `push`，则需要 `force push`
- `空格` 滚回版本

- **g** 显示重置选项 soft 已更改变成未更改 然后在文件中D
- 历史记录 Reflog中 **z/cmd+z** 撤消/重做
- **c** 复制commit
- **v** 粘贴commit
- **d** 删除commit
- **s** 合并两次提交
- **f** 作为fix commit
- **Ctrl+f** 查看有某一文件的全部提交
- **M** 查看commit之间的diff

合并冲突

- **回车** 进入右侧预览界面
- **上下键** 选择
- **左右键** 在不同冲突间跳转
- **空格** 选取需要的部分
- **b** 两种更改都保留