在合适的地方创建空目录 \$ mkdir learngit \$ cd learngit \$ pwd /Users/michael/learngit 在该目录下生成.git目录 \$ git init 创建repository \$ Is -ah 新建标签、默认HEAD、可指定commit id 把文件提交到仓库,可多次git add添加多个文件 \$ git tag <tagname> <commit id> \$ git add <file> \$ git commit -m <message> 指定标签信息 \$ git tag -a <tagname> -m "blablabla..." 查看状态和查看修改内容 查看所有标签 \$ git status \$ git tag \$ git diff <file> 查看标签说明文字 \$ git show <tagname> 查看commit历史 (+简单版) 标签管理 \$ git log 删除本地标签 \$ git log --pretty=oneline \$ git tag -d <tagname> 删除远程标签 版本回退 \$ git push origin :refs/tags/<tagname> \$ git reset --hard HEAD^ \$ git reset --hard HEAD~<num> 推送本地标签 \$ git push origin <tagname> 查看命令历史,以便回到未来 推送全部未推送过的本地标签 \$ git reflog \$ git push origin --tags 版本管理 修改后未add、恢复到版本库状态 修改后add又修改,恢复到add状态 Git 创建并切换到dev分支 //用版本库中版本替换工作区版本 \$ ait switch -c dev \$ git checkout -- file 相当于 \$ git branch dev 把暂存区的修改撤销掉, 重回工作区 \$ git switch dev \$ git reset HEAD <file> 撤销修改 查看当前分支 修改后add, 想回到修改前 \$ git branch \$ git reset HEAD <file> \$ git checkout --file 切换分支 \$ git switch <name> 删除文件 分支管理 \$ git rm file 合并指定分支到当前分支 \$ git commit -m "remove file" \$ git merge <name> 删除dev分支 第一次把当前分支main推送到Github \$ git branch -d dev \$ git push -u origin main 以后每次推送 合并时生成新的commit \$ git push origin main \$ git merge --no-ff -m "merge with no-ff" dev 查看远程库信息 \$ git remote -v 删除远程库 Github协同 \$ git remote rm <name> 从远程库克隆 \$ git clone <链接> \$ git clone git@github.com:michaelliao/gitskills.git \$ git clone https://github.com/michaelliao/gitskills.git