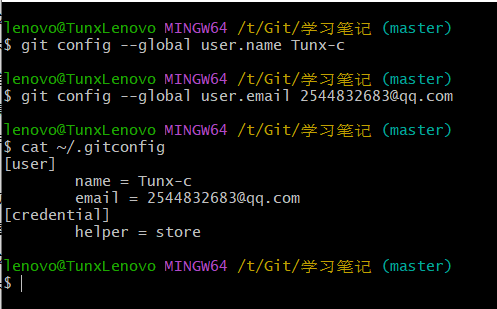
## 配置

**设置用户签名**

git config --global user.name 用户名

git config --global user.email 邮箱



配置文本编辑器：git config --global core.editor "code --wait"

查看配置：git config --global alias.co checkout

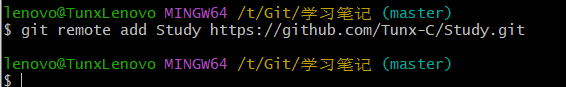
配置忽略文件：创建一个.gitignore文件

## 创建仓库

本地创建仓库 git init

托管平台创建仓库

关联远程仓库 git remote add 仓库名 <remote-repository-url>



## 基本操作

**初始化仓库** git init

**查看状态** git status

**添加文件到暂存区**

git add filename

支持通配符

**提交暂存区至版本库**

git commit -m "Commit message" //提交信息

**查看提交历史**

git log

git log --oneline

**比较工作目录和暂存区之间的差异**

git diff

**比较暂存区和最新提交之间的差异**

git diff --staged

git diff --cached

**比较两个提交之间的差异**

git diff <commit-hash-1> <commit-hash-2>

**比较指定文件的差异**

git diff <filename>

**版本回退**

回退到上个版本

git reset --hard HEAD^

回退到上上个版本

git reset --hard HEAD^^

回退到前10个版本

git reset --hard HEAD~10

回退到指定版本号的版本

git reset --hard a7f365c128827cdf1f0d796141ce3c644cc78770

**查询版本号**

git reflog

**恢复工作目录更改**

将指定文件恢复到版本库中的版本

git restore <filename>

将所有文件恢复到版本库中的版本

git restore .

注意，旧版本不支持restore命令，需要使用该命令

git checkout -- <filename>

放弃指定文件修改

git reset HEAD <filename>

放弃所有文件修改

git reset HEAD .

**撤销提交**

撤销一个提交

git revert [版本号]

可以指定一个范围，撤销多个连续的提交

git revert <start-commit>..<end-commit>

## 分支管理

不影响主线代码的情况下进行并行开发

**创建分支 开发新功能且保护主线代码**

创建一个分支

git branch <branch-name>

创建并切换到分支

git branch -b <branch-name>

**切换分支**

切换分支

git checkout <branch-name>

查看当前所在分支

git checkout

**合并分支**

git merge <branch-name>

**解决冲突**

两个分支有不同的更改

打开冲突文件编辑

使用git add 标记冲突已解决

提交冲突更改

**删除分支**

git branch -d <branch-name>

## 远程仓库

本地仓库和远程仓库传输为SSH加密，还需要将SSH秘钥添加到远程仓库设置中

**关联远程仓库**

git remote add origin <remote-repository-url>

**推送代码**

git push origin <branch-name>

首次推送一般需要-u参数来建立跟踪关系

**拉取代码**

git pull origin <branch-name>

保持本地代码同步

**解决远程仓库冲突**

远程仓库可能会冲突，一般在推送代码前，要拉取最新代码以避免冲突

**查看远程仓库**

git remote -v

## 高级操作

## 团队协作

## 速查

基本操作

初始化仓库：git init

克隆远程仓库：git clone <repository-url>

查看状态：git status

添加文件到暂存区：git add <filename>

提交更改：git commit -m “Commit message”

查看提交历史：git log

查看文件差异：git diff <filename>

查看远程仓库列表：git remote -v

查看远程分支列表：git branch -r

分支操作

创建分支：git branch <branch-name>

切换分支：git checkout <branch-name> 或 git switch <branch-name>

创建并切换分支：git checkout -b <branch-name> 或 git switch -c <branch-name>

合并分支：git merge <branch-name>

删除分支：git branch -d <branch-name>

远程仓库操作

关联远程仓库：git remote add <remote-name> <repository-url>

推送代码：git push <remote-name> <branch-name>

拉取代码：git pull <remote-name> <branch-name>

创建Pull Request：在远程仓库上操作

获取远程仓库更新：git fetch

高级操作

重写历史：git commit --amend、git rebase -i <commit-hash>

使用储藏：git stash、git stash apply、git stash pop

管理子模块：git submodule add <repository-url>

重置和回退：git reset --soft <commit-hash>、git reset --hard <commit-hash>、git reflog

团队协作

创建Pull Request：在远程仓库上操作

代码审查：通过Pull Request 进行

解决冲突：手动编辑文件解决冲突

分支保护和权限管理：在远程仓库上设置

常见问题解决

解决冲突：手动编辑冲突文件

撤销错误提交：git reset --soft HEAD^、git reset --hard HEAD^、重写历史

丢弃未提交更改：git restore <filename>