git使用

安装 brew install git
查看版本 git --version

配置git

还需要最后一步设置，在命令行输入：
git config --global user.name "Your Name"
git config --global user.email "email@example.com"
git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。

创建空目录

首先，选择一个合适的地方，创建一个空目录：

mkdir learngit
cd learngit
pwd　　//pwd命令用于显示当前目录//

第二步，通过git init命令把这个目录变成Git可以管理的仓库：

git init
如果你没有看到.git目录，那是因为这个目录默认是隐藏的，用ls -ah命令就可以看见。

现在我们编写一个readme.txt文件，一定要放到learngit目录下（子目录也行），因为这是一个Git仓库，放到其他地方Git再厉害也找不到这个文件。

把一个文件放到Git仓库只需要两步。
第一步，用命令git add告诉Git，把文件添加到仓库：
git add 文件名.后缀

第二步，用命令git commit告诉Git，把文件提交到仓库：
$ git **commit** -m “描述”
git commit命令，-m后面输入的是本次提交的说明，可以输入任意内容，当然最好是有意义的，这样你就能从历史记录里方便地找到改动记录。

为什么Git添加文件需要add，commit一共两步？因为commit可以一次提交很多文件，所以你可以多次add不同的文件，如：
git add file1.txt
git add file2.txt file3.txt
git commit -m "add 3 files."

了解文件更改

git status命令可以让我们时刻掌握仓库当前的状态
git status告诉你有文件被修改过，用git diff可以查看修改内容。

git log命令显示从最近到最远的提交日志
如果嫌输出信息太多，看得眼花缭乱的，可以试试加上--pretty=oneline参数：
git log --pretty=oneline

回退到上一个版本

首先，Git必须知道当前版本是哪个版本，在Git中，用HEAD表示当前版本，上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100。
使用git reset命令：
git re**set** --hard HEAD^

HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard commit_id。
穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。
要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

放弃更改

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。
git checkout -- 文件名.后缀

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD file，就回到了场景1，第二步按场景1操作。

删除文件

要从版本库中删除该文件，那就用命令git rm删掉，并且git commit
git rm 文件名.后缀
git commit －m “描述”
删错了，因为版本库里还有呢，所以可以很轻松地把误删的文件恢复到最新版本：
git checkout -- 文件名.后缀