git应用

1、git文件状态简介

·git中文件的三种状态：committed （已提交），modified （已修改），staged （已暂存）

committed:表示该 文件已经被安全地保存在本地数据库中了

modified:表示修改了某个文件，但还没有提交保存

staged:表示把已修改的文件放在下次提交时要保存的清单中

git的基本工作流程：在工作目录中修改某些文件

保存到暂存区域

提交更新，转储到 git 目录中

2、初次运行git前的配置

·git config命令用来配置或读取Git的工作环境。

git config --system 配置对所有用户都普遍适用的配置

git config --global 配置对所有用户都普遍适用的配置

（1）配置用户信息：

git config --global user.name “你的个人用户名称”

git config --global user.email “你的个人用户邮箱”

（2）设置文本编辑器

git config --global core.editor“文本编辑器”

（3）设置差异分析工具

git config --global merge.tool“差异分析工具”

3、Git常用命令介绍

（1）创建git仓库

从现有目录创建：切换到现有目录 git init

从现有仓库创建：git clone [url网址]

（2）检查文件状态：git status

（3）添加文件到暂存区域：git add 文件名

git add -A:提交所有文件到暂存区域

（4）提交文件到git目录

将暂存区域提交：git commit -m “文件说明”

（5）移除文件

git rm文件名

git rm-f文件名：删除暂存区的文件

（6）比较文件修改的差异

git diff

（7）移动文件或者重命名文件

git mv origin\_filename rename\_filename

（8）查看提交历史

git log

（9）回退到以前的版本

git reset –hard版本号

（10）添加远程库

1）创建SSH key:ssh-keygen –t rsa –C “YourEmail”

2) 查看.ssh目录，找到id\_rsa和id\_rsa.pub两个文件

3）登录GitHub账户，输入id\_rsa.pub公钥

4）测试是否成功ssh-T git@github.com

5）在github上创建一个与本地仓库同名的空仓库

6）运行命令：git remote add origin git@github.com:用户名/learngit.git（origin可修改成其他名字）

7）git push -u origin master：推送到远程仓库

注：当你第一次使用Git的clone或者push命令连接GitHub时，会得到一个警告：

The authenticity of host 'github.com(xx.xx.xx.xx)' can't be established.

RSA key fingerprint is xx.xx.xx.xx.xx.

Are you sure you want to continue connecting(yes/no)?

这是因为Git使用SSH连接，而SSH连接在第一次验证GitHub服务器的Key时，需

要你确认GitHub的Key的指纹信息是否真的来自GitHub的服务器，输入yes回车即可。

（11）删除远程库

git remote rm项目仓库名称

（12）从远程库克隆

git clone项目仓库地址

4、Git的分支管理

•Git分支简介：Git中的分支，其实本质上仅仅是个指向commit对象的可变指针

•创建与合并分支

•冲突管理和远程分支

•忽略特殊文件

（1）查看分支 git branch

（2）创建分支 git branch分支名

（3）切换分支 git checkout分支名

（4）创建+切换分支 git checkout -b分支名

（5）合并某分支到当前分支 git merge分支名

git merge --no -ff分支名

（6）删除分支 git branch -d分支名

**冲突管理和远程分支**

冲突管理：手动打开冲突文件，留下自己需要的内容，再次提交文件

推送远程分支：git push origin分支名称

在本地创建和远程分支对应的分支：git checkout–b分支名称

origin/分支名称

建立本地分支和远程分支关联：git branch --set-upstream-to origin/分支名称 分支名称

抓取远程分支的更新与本地分支合并：git pull <远程主机名> <远程分支名>:<本地分支名>

注：与当前分支合并可以省略本地分支名

**忽略特殊文件**

•创建并编写.gitignore文件：输入你想忽略的文件,例如：.obj、.tmp

•将.gitignore文件提交到git

github简介

一、GitHub基本概念

1、Repository（仓库）：用来存放你的项目代码，每个项目对应一个仓库。

2、Star（收藏）：即收藏别人的项目，方便查看

3、Fork:复制克隆项目，与所克隆的项目是独立存在的

4、PullRequest：发起请求，等待原项目负责人查看，考虑是否合并到源仓库中

5、Watch：如果你watch了一个项目，如果以后这个项目有更新时可及时接收到通知。

6、Issue:发现代码有Bug时，向项目负责人提提案，讨论解决问题

7、gist:用来粘贴数据（代码）

二、开源项目贡献流程

1、通过新建Issue提交使用问题或者建议给作者

2、通过PullRequest

Fork作者项目

对项目进行修改并提交PullRequest

等待作者审核

三、团队合作开发项目

1、通过 Collaborators

2、Fork & Pull Request

1）Fork别人的项目到自己仓库

2）将项目clone到本地进行修改，并更新到远程仓库

3）向项目作者提Pull Request

4）作者merge合并项目

如何上传文件至github

登录GitHub账号后，点击右上角的加号图标，选择“New Repository”来创建一个新的仓库。

填写仓库的名称、描述（可选）、选择仓库的可见性（公开或私有）、是否初始化仓库（建议勾选以包含README文件、.gitignore文件以及license文件等）、以及是否添加README文件等选项。

点击“Create Repository”按钮完成仓库的创建。

配置Git：配置Git的用户名和邮箱地址。可以通过Git命令行来完成，输入以下命令：

git config --global user.email "你的邮箱地址"

git config --global user.name "你的用户名"

请确保使用你在GitHub上注册时使用的邮箱地址。

上传文件

方法一：使用Git命令行

初始化本地仓库：在包含你要上传文件的文件夹中打开命令行（Git Bash或终端），执行git init命令来初始化一个新的Git仓库。

添加文件到暂存区：使用git add .命令将文件夹中的所有文件添加到暂存区，或者使用git add 文件名来添加特定文件。

提交更改：执行git commit -m "提交信息"命令来提交更改到本地仓库。将“提交信息”替换为对本次提交的简短描述。

连接到远程仓库：执行git remote add origin 远程仓库地址命令来将本地仓库与GitHub上的远程仓库连接起来。将“远程仓库地址”替换为创建的GitHub仓库的URL。

推送文件到远程仓库：执行git push -u origin main（或默认分支名，如master，但GitHub现在默认使用main作为主分支名）命令来将本地仓库中的文件推送到GitHub上的远程仓库。

方法二：使用GitHub Desktop

如果不熟悉Git命令行，可以使用GitHub Desktop这一可视化工具来上传文件。

下载并安装GitHub Desktop：从GitHub Desktop官网下载并安装GitHub Desktop。

添加仓库：打开GitHub Desktop，点击“Add”按钮添加要上传的仓库。

上传文件：在GitHub Desktop中，可以直接将文件拖放到仓库的界面中，或者通过“File”菜单选择“Add local repository”来添加本地已存在的仓库。

提交并推送：在GitHub Desktop中填写提交信息，然后点击“Commit to main”（或默认分支名）按钮来提交更改。之后，点击“Push origin”按钮将更改推送到GitHub上的远程仓库。