**·文件状态简介**

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

**·运行配置**

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

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”

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

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

**·常用命令**

*（1）创建git仓*库

从现有目录创建：切换到现有目录，gitinit

从现有仓库创建：gitclone [url网址]

（2）检查文件状态：gitstatus

（3）添加文件到暂存区域

gitadd 文件名

gitadd -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：gitcommit -m “文件说明”

（5）移除文件

gitrm文件名

gitrm-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 “Your Email”

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

（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

**·分支管理**

创建与合并分支

（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

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>

**·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、Pull Request：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）