**学习日志**

姓名：李奥 学号： 1812998 日期： 6.15

**学习内容安排：（以6.15为例）**

上午：

Git简介、安装Git、创建版本库、版本回退、工作区与暂存区、管理修改、撤销修改、删除文件

下午：

远程仓库、分支管理、标签管理、使用GitHub、忽略特殊文件、配置别名、使用SourceTree

**学习反馈：**

已掌握知识：

1. Git简介

有效追踪文件变化，易于回滚，解决多人协作问题，提高开发效率。

2. 创建版本库

$ mkdir learngit 创建一个目录，其中learngit为目录名

$ cd learngit 进入到该目录下

$ pwd 显示当前目录

$ git init 初始化仓库，将该目录变成可以管理的仓库

$ git add readme.txt 将文件添加到仓库

$ git commit -m “wrote a readme file” 将文件提交到仓库，其中引号中的内容作为提交说明

3. 版本回退

$ git status 掌握工作区当前状态（是否有文件修改了没添加，是否有文件待提交）

$ git diff 查看修改内容

提交修改后的文件步骤和添加新文件步骤相同。

$ git log / $ git log --pretty=oneline 查看提交日志，HEAD表示当前版本

$ git reflog 查看命令历史，可以确定要回到未来的哪个版本

$ git reset --hard commit\_id 修改版本，其中commit\_id为版本id号前几位

4. 工作区和暂存区

工作区—电脑里能看到的目录

版本库—隐藏目录.git

暂存区（stage）--git add 将文件修改添加到暂存区，git commit 将暂存区的所有内容提交到当前分支

第一个分支（master）

指向master的指针HEAD

5. 管理修改

每次修改，如果不用git add到暂存区，那就不会加入到commit中。

6. 撤销修改

$ git checkout -- file 丢弃工作区修改

$ git reset HEAD <file> 丢弃暂存区修改，之后重复上一步

7. 删除文件

$ git rm file 删除一个文件，之后需要commit

8. 远程仓库

$ git remote add origin git@server-name:path/repo-name.git； 关联一个远程库

$ git push -u origin master 第一次推送master分支的所有内容

$ git push origin master 此后，每次本地提交后，只要有必要，就可以使用命令推送最新修改

$ git clone 要克隆一个仓库，首先必须知道仓库的地址，然后使用命令克隆。

Git支持多种协议，包括https，但ssh协议速度最快。

9. 分支管理

$ git branch 查看分支

$ git branch <name> 创建分支

$ git checkout <name> / $ git switch <name> 切换分支

$ git checkout -b <name> / $ git switch -c <name> 创建+切换分支（推荐用switch）

$ git merge <name> 合并某分支到当前分支

$ git branch -d <name> 删除分支git

当Git无法自动合并分支时，就必须首先解决冲突。解决冲突后，再提交，合并完成。

解决冲突就是把Git合并失败的文件手动编辑为我们希望的内容，再提交。

$ git log --graph 用命令可以看到分支合并图。

合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出来曾经做过合并。

修复bug时，我们会通过创建新的bug分支进行修复，然后合并，最后删除；

git stash 当手头工作没有完成时，先把工作现场保留一下，然后去修复bug

git stash pop 回到工作现场

git cherry-pick <commit> 在master分支上修复的bug，想要合并到当前dev分支，可以用命令，把bug提交的修改“复制”到当前分支，避免重复劳动。

git branch -D <name> 删除没有被合并过的分支

git remote -v 查看远程库信息

本地新建的分支如果不推送到远程，对其他人就是不可见的

git push origin branch-name 从本地推送分支，如果推送失败，先用git pull抓取远程的新提交

git checkout -b branch-name origin/branch-name 在本地创建和远程分支对应的分支，本地和远程分支的名称最好一致

git branch --set-upstream branch-name origin/branch-name 建立本地分支和远程分支的关联

10. 标签管理

git tag <tagname> 新建一个标签，默认为HEAD，也可以指定一个commit id

git tag -a <tagname> -m "blablabla..." 指定标签信息

git tag 查看所有标签

git push origin <tagname> 推送一个本地标签

git push origin –tags 推送全部未推送过的本地标签

git tag -d <tagname> 删除一个本地标签

git push origin :refs/tags/<tagname> 删除一个远程标签

11. 忽略特殊文件

忽略某些文件时，需要编写.gitignore

.gitignore文件本身要放到版本库里，并且可以对.gitignore做版本管理

未掌握知识：

学习心得：git的分支系统十分优越先进