学习git

1. 安装git

安装完成后在命令行输入：

$ git config --global user.name "Your Name"

$ git config --global user.email email@example.com

1. 创建版本库

mkdir <file\_name>

git init 将当前目录变成可以管理的仓库（建立.git目录）

git add <文件名>

git commit –m ‘<提交的说明>’

1. 时光机穿梭

git status 掌握仓库当前状态

git diff <file\_name> 查看文件difference

git diff HEAD -- <file\_name> 查看工作区和版本库里面最新版本的区别

1. 版本回退

git log 从最近到最远提交日志

（嫌输出信息太多： git log –pretty=oneline ）

git reset –hard HEAD^

cat <file\_name>

git reset –hard <版本号> （回到未来）

git reflog 记录每一次命令

1. 撤销修改

git checkout -- <file\_name>

1. 删除文件

rm < file\_name >

git rm

git checkout其实是用版本库里的版本替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”。

1. 远程仓库github

进入github.com，点右上角your repositories ，点new建立仓库，根据github提示（在你新建网络仓库下面）然后回到本地的仓库输入对应代码：

git init

git add <文件名>

git commit –m “<本次修改说明>”

git remote add origin git@github.com:usersunny/.......

git push –u origin master

1. 从远程库克隆

git clone [git@github.com:usersunny/](mailto:git@github.com:usersunny/).......

1. 创建与合并分支

git checkout –b dev 创建+切换到dev分支

git branch 查看当前分支

git branch <name> 创建分支

git checkout <name> 切换分支

git merge <name> 合并某分支到当前分支

git branch –b <name> 删除分支

1. 解决冲突

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。

用git log --graph命令可以看到分支合并图。

1. 分支管理策略

--no-ff 表示禁用Fsat forward

git log 查看历史分支

1. Bug分支

git stash “储藏”当前工作现场

1. Feature分支

开发一个新feature，最好新建一个分支

如果要丢弃一个没有被合并过的分支，可以通过git branch -D <name>强行删除。

1. 多人协作

多人协作的工作模式通常是这样：

首先，可以试图用git push origin <branch-name>推送自己的修改；

如果推送失败，则因为远程分支比你的本地更新，需要先用git pull试图合并；

如果合并有冲突，则解决冲突，并在本地提交；

没有冲突或者解决掉冲突后，再用git push origin <branch-name>推送就能成功！

如果git pull提示no tracking information，则说明本地分支和远程分支的链接关系没有创建，用命令git branch --set-upstream-to <branch-name> origin/<branch-name>。

这就是多人协作的工作模式，一旦熟悉了，就非常简单。

查看远程库信息，使用git remote -v；

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；

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

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

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

从远程抓取分支，使用git pull，如果有冲突，要先处理冲突。

1. Rebase

实际操作时深入理解

1. 标签管理：创建标签

git tag <name> 新建标签

git tag 查看所有标签

git log –pretty=oneline –abbrev-commit 显示历史提交的commit id，可以在对应commit id上新增标签： git tag <标签号> <commit id>

git show <tagname>

git tag –a <tagname> -m ‘<说明>’ <commit id> 创建带有说明的标签

1. 操作标签

git push origin <tagname> 推送某个标签到远程

git push origin –tags 一次性推送全部尚未推送到远程的本地标签

git tag –d <tagname> 删除标签（本地）

git push origin :refs/tags/<tagname> 删除标签（远程）