**Git 的学习笔记**

* ***基本操作***

git init 在本地的当前目录初始化git仓库。

git status 查看当前目录有什么变化。

git add –A 添加所有未添加的文件到staging。

git commit –m “提交信息” 将文件从staging添加到repository里。

git log 查看提交到repository里的记录。

***对workspace的操作***

git clean –xf 删除当前目录下所有未被track的文件。

***对staging 的操作***

git rm --cached “文件路径” 不删除物理文件，将文件从缓存里删除。

git rm –f “文件路径” 删除物理文件，将文件从缓存区删除。

git diff 看看staging里的哪些文件做了哪些改动。

git checkout -- . 撤销这些改动。

***对local repository 的操作***

git log 先查看版本号

git reset --hard 1df3wd5(取版本号的前7位) 版本回退。

git reflog 先查看版本号

git reset --hard ad93b89(去版本号的前7位) 版本回到回退之前的状态。

* ***本地git 与 github 的连接***

git config --global user.name "你的用户名"

git config --global user.email "你的邮箱"

ssh-keygen –t rsa –C “你的邮箱” 生成ssh key

将生成的ssh key复制到图中提示的路径下。

进入github -> setting -> SSH and GPG keys -> key 复制

* ***远程我的仓库与本地我的电脑关联***

创建远程仓库 -> clone with SSH -> git remote add origin 你复制的地址

* ***分支***

git checkout –b 分支名 创建分支“分支名”

git checkout –d 分支名 删除分支”分支名”

git push origin 分支名 将本地分支推送到远端仓库

git branch 查看本地有哪些分支

git branch –a 查看本地和远端仓库所有分支

git checkout –b de origin/dev 创建本地分支de，并将远程分支dev的内容放入该分支

git checkout 分支名 切换到分支“分支名”

ls 查看该分支内容

* ***更新远端仓库的团队项目到主机***

git remote –v 查看有没有设置upstream

* 如果没有 git remote add upstream 团队项目地址

git fetch upstream 获取团队最新版本。

git merge upstream/dev 将源分支(团队最近版本所在的分支)合并到本机当前分支

***或者***：

git pull origin dev pull = fetch + merge 下载并合并远端仓库dev分支到本机当前分支

***解决冲突***

在git pull origin dev 时提示 *fatal: Exiting because of an unresolved conflict.*

则:

手动修改本地发生冲突的文件 -> git add “文件” -> git commit –m “提交信息” -> git pull origin dev

注： 可以用冲突解决工具Meld。

***将本地项目push到远端仓库***

git push 将本地的当前分支，push到远端对应的同名仓库

git push origin HEAD:分支 将当前分支push到远端的仓库“分支”中。

注：如果远端分支的文件已经被改动，则需要：

git pull origin dev -> 解p决冲突 -> 再次 git push

***合作开发：***

***Fork方法***

要贡献代码给别人仓库，先fork别人的仓库，然后自己的仓库里会有一个一模一样的仓库，pull（clone）下来修改后push到自己的仓库，然后发起一个”pull request”给你想贡献代码的人，他会收到你的pull request，同意即可合并。

【问题】可能在你修改的过程中，别人动了代码。

***合作开发方法***

将自己的 用户名或者公钥发给别人获取权限，拥有权限之后，可以直接操作对方的仓库： pull, push。