git与github

**git**是一个版本控制的工具

**github:**网站->程序员基友网站，可以去学习，代码托管，远程仓库，放静态资源的博客域名地址

安装工具

要注册账户

github与git相关联

在github上创建新的项目

把创建好的项目克隆到本地，形成一个版本

cd 进入盘符

cd d:

ls或者ll查看目录

可以按tab键来补全你的文件名

**退回** cd ..

**清屏** clear

**克隆：**git clone github地址。

有了master才能进行版本控制

工作区添加了一个文件

从工作区到暂存区

**git add 文件名**

快捷方式（一次性提交多个文件）

**git add .**

从暂存区到版本区

**git commit -m "注释"**

形成版本之后要把代码放到远程仓库

**git push origin master**

当某个（些）文件已经提交过了，如果要直接从工作区到版本区

git commit -a -m "注释"

设置密码长期保存

git config --global credential.helper store

**对比：**

查看工作区与暂存区的区别

git diff

暂存区与版本区的区别

git diff --cached

工作区与版本区的区别

git diff master

**撤销：**

git reset HEAD 文件名 可以将暂存区文件撤销会工作区（绿色变红色）

工作区代码还原暂存区或版本区

git checkout -- 文件名

git commit -m “注释” --amend

如果有2个以上文件，一个提交到版本库了，另一个忘记提交，

可以先将没提交的文件拉到暂存区，然后通过git commit -m “注释” --amend 撤销回来，

最后自动一次性提交暂存区中的文件和撤销回来的版本形成一个新的版本，

撤销回来的版本就销毁了，git log查看是否提交成功

**删除:**

如果手动删除了工作区的文件，也想在git中把它删除

git rm 文件名

一次性把工作区与暂存区的文件都删掉

git rm -f 文件名

git rm --cached 文件名(只删除暂存区，不删除工作区)

删除整个文件夹：

git rm -rf 文件名

**恢复:**

恢复一个文件：

git checkout 历史记录编码 文件名（要恢复的文件名） 指定的文件还原。

git reset --hard 历史记录编码 还原整个版本

git reset --hard HEAD^ [1,2,3,4]

回滚版本记录

输出第一次为最近的记录，第二次就是倒数第二个历史记

git reset --hard HEAD~2 回滚倒数第三个历史记录 （跳过了2个）

如果说git log找不到历史ID，可以通过git reflog去查看操作过的历史记录

**同步到远程仓库**

查看远程仓库名：

git remote (默认origin)

查看上传地址和下载地址是否为一个地址

git remote -v

创建远程仓库

git remote add 名字 github的地址

上传远程仓库

git push origin(默认) master

**协作：**

需要创建项目者给协作者权限

进入项目 -> settings ->

最左边 Collaborators -> 添加协作者名字

等待协作者确定

协作者：

1.确定协作

2.clone项目

3.参与项目开发

4.提交上传

有可能会遇到冲突:

git pull (直接把远程的代码覆盖到本地)（不太推荐）

git fetch (把远程仓库的代码拉取下来不覆盖)

**查看**哪里有**冲突**

git diff master origin/master

**合并冲突**

git merge origin/master (你会发现master变成了matser|MERGEING)

人为判断冲突，把冲突内容删除

删除完成之后（代码被修改了，需要重新提交）

保存到版本区之后，继续push （matser|MERGEING就变成了master）

**没权限协作：**

1.找到想参与的项目fork

2.把项目克隆到本地 -> 修改 -> 提交

3.点击 Pull requests

4.点击 new pull request

5.点击 Create pull request

6.对话点击Create pull request

7.等待

收到**修改消息**

1.点击修改的消息

2.点击files changed（查看修改内容）

3.merge pull request(可以回复别人)

**拷课件步骤:**

先删除fork的项目

在fork课件项目

到本地使用git pull

**分支：**

**新建分支：**

git branch 分支名

**查看分支:**

git branch

**切换分支:**

git checkout 分支名

**快速创建 + 切换**

git checkout -b 分支名

**合并分支:**

git merge 分支名

可能会出现冲突:

人为修改文件 -> 提交

**删除已经合并的分支:**

git branch -d 分支名

**删除未合并的分支:**

git branch -D 分支名

**查看已经合并的分支:**

git branch --merged

**查看未合并的分支:**

git branch --no-merged