git hub 笔记

cd 盘符切换

cd.. 返回上一个盘符

ls 查看整个目录下的结构

mkdir +文件名 创建文件夹

Tab键自动补全命令

1. Git 工具 版本控制
2. Github 远程仓库

命令 git clone 下载github的资源到本地

设置开发者的名字和邮箱：

git config --global user.name “github 的名字”

git config --global user.email “github 的email”

查看github的配置： git config --list

git的三个区：

1. 工作区
2. 暂存区

作为过渡区，避免误操作

（保护工作区和暂存区），分支处理

1. 版本区 主分支

命令 git status git 目前的状态

**提交：**

git add 文件名字 到暂存区

git add . 全部提交到暂存区

git commit 提交到版本区

git commit -m ‘可以写注释’

git commit -a -m ‘可以写注释’ 直接添加到版本区

git log 版本区的操作详情

**对比：**

git diff 工作区和暂存区的差别对比

git diff --cached 暂存区与版本区的对比

或者 git diff --staged 暂存区与版本区的对比

git diff master 工作区与版本区直接对比

**撤销：**

git reset HEAD “ 文件名” 暂存区撤销到工作区

git checkout --“文件名” 撤销工作区修改的内容

git commit --amend 撤回提交操作（误提交之后的撤回操作）

**删除：**

git rm “文件名” 删除工作区的文件 相对应的暂存区的也删掉

git rm -f “文件名” 删除工作区和暂存区的文件

git rm --cached “文件名” 只删除暂存区的文件 保留工作区的

**恢复：**

**回到过去的操作：**

不小心误删除之后的操作 通过git log 来查看之前的操作记录

找到之前的操作记录的id

git checkout id代码 文件名 对指定文件的恢复

或者 git reset --hard commit id 恢复整个版本 对整个版本的恢复

git reset --hard HEAD^ 恢复到指定的操作记录

比如：当前是change3 要恢复到change2的操作

再次恢复 需要再次提交

或者 git reset --hard HEAD~2 直接恢复到过去的版本 退回2步

**回到现在的操作：**

git reflog 找到要每次操作的记录 找到id

直接 git reset --head id 就可以恢复到之前的操作文件

**同步到远程仓库：**

git push 同步到远程仓库

git remote 可以查到远程仓库的名字 默认是origin

git remote -v 可以查到远程仓库的地址

git push 远程仓库的名字（默认origin） master(主分支名字)

只有一个文件时默认是master 例如： git push origin master

**多人开发：**

第一步 在git hub上 进入要多人开发的项目，点击settings之后

找到 collborators 点击进入之后 输入要协作的人员的github账号

第二步 协作人员的github 上确认即可；

第三步 协作人员要设置 名字和邮箱 已经到 github上git clone 要开发的文件

**多人协作 解决冲突：**

首选方法1

git fetch 同步更新代码 拉取不合并

git diff master origin/master 对比冲突代码

git merge origin/master 手动合并 需要在文件里删除不需要的冲突代码

git commit -a -m “命名(随便起个）”

git push origin master 再次提交到远程仓库

方法2

git pull 同步更新代码 拉取直接合并

不需要以下这两步: 直接合并

git diff master origin/master 对比冲突代码

git merge origin/master 手动合并

直接走这一步：

git commit -a -m “命名(随便起个）”

git push origin master 再次提交到远程仓库

**开源项目：**

功能负责的情况下 需要下载到本地修改

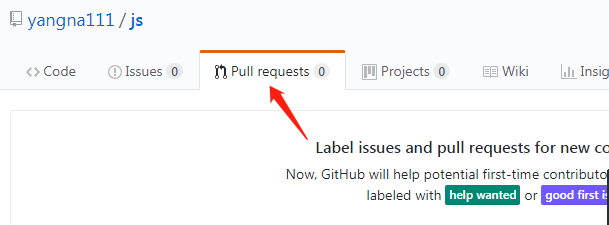
fork 开源项目 开了一个新的分支 镜像了一个新的版本

先git clone 到本地工作区 之后进行修改

提交 git commit -a -m “命名(随便起个）”

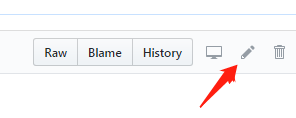
再 git push origin master push到远程仓库

在当前文件主页面 点击 pull request 发送



对方会根据情况来接受或者不接受

另一种是简单修改 直接在fork 后的git hub 文件上修改



修改后 在提交给对方来决定要不要修改

**删除fork 的远程仓库：**

点击主页面的settings设置里，点击Delete this repository选框输入要删除的文件名 进行删除

**git 分支处理：** 多人开发 互不影响

git branch 查看当前的分支

git branch +名字 创建分支 例如：git branch new1

git checkout +名字 切换分支 例如：git checkout new1

Git checkout master 切换到主分支

快速创建 并且切换到分支上：

git checkout -b +名字 例如：git checkout -b new2

把分支上的代码合并到主分支上：

在主分支master 下输入：git merge 名字 例如：git merge new1

git branch --merged 查看与master合并的分支

git branch --no-merged 查看没有与master合并的分支

git branch -d 名字 删除已经合并没有用的分支

如：git branch -d new1

没有合并的分支是不能直接执行删除 如要强制删除没有合并的分支，可以用git branch -D new1(名字)

打标签 ：记录现在的状态 也可以在git hub上直接打标签

比如一个完整的demo

git tag 查看标签

git tag+名字 如： git tag v1.0 打上标签

git psuh origin v1.0 push到远程仓库