Git

1.克隆仓库

命令行

cd返回主页?

cd Desktop

mkdir 文件名(myRepo) 建立文件夹

cd myRepo (进入该文件夹)

git init(建立本地仓库)

打开Github,建立新库,复制url

回到终端

git clone url

输入密码

ls查看新目录

echo hello>hello.txt 将字符串写到该文件,文件重名覆盖,不重名新建

git add. 添加到暂存区

git commit -m”first commit” -m允许在一行命令中直接提供提交信息

git ls-files查看仓库状态 可以看到多了一个hello.txt文件

下面使用push命令将本地仓库的修改内容推送给远程仓库

git push

打开远程仓库ctrl+shift+R刷新

1. 关联本地仓库和远程仓库

远程仓库建立新库,复制url

cd myRepo 进入本地仓库

查看github提示,这里是git remote add origin git@github.com:shieldforgrey/first-repo.git,复制粘贴 origin为仓库默认别名

git remote -v 查看当前仓库对应的远程仓库的别名和地址,这里显示

origin git@github.com:shieldforgrey/first-repo.git (fetch)

origin git@github.com:shieldforgrey/first-repo.git (push)

表示origin为别名,后面为地址

提示第二行git branch -M main表明指定分支名称为main,因为默认分支就是main,所以可以省略该行

提示第三行git push -u origin main表明把本仓的main分支和远仓的main分支关联,全称应该是git push -u origin main:main u是upstream缩写

回车后输入密码,回车,成功

远程仓库修改内容,pull到本地仓库

git pull <远程仓库名> <远程分支名>:<本地分支名>如git pull origin main

git会自动为我们进行一次合并操作,如果不冲突就是成功,否则需要手动

git fetch类似git pull,但是不会合并

可以看到本仓有了新修改文件,可以使用cat filename 查看

3.查找文件或者目录

find / -type d -name “项目文件名称” 2>/dev/null

解释

find / 表示从根目录/查找

-type指明要查找的文件类型为目录(d)

-name 指定要查找的文件或目录的名称

1. /dev/null 是将标准错误输出重定向的语法,/dev/null是一个特殊的设备文件,接受所有写入但不执行任何操作,这就相当于将错误信息丢弃,这样做是为了忽略对某些目录没有访问权限的错误,以保持输出的整洁性

git init

git add . 将所有文件添加到暂存区

git commit -m”初始化工程” 提交到本地仓库

ls ~/.ssh/id\_rsa 检查本地是否有ssh秘钥

解释:列出指定路径下的文件或目录, ~表示主目录,后面表示主目录下的.ssh文件夹中的id\_rsa文件, .ssh中的. 表示是隐藏文件或者隐藏文件夹

配置秘钥

ssh-keygen -t rsa -b 4096 -C “dong661018@gmail.com”

输入文件位置，直接回车就默认位置 ~/.ssh/id\_rsa

输入密码（我的是6位+全小，和其他密码一样）

打开秘钥文件

cat ~/.ssh/id\_rsa.pub

复制ssh-rsa开头的ssh公钥

常见命令行

ls 查看