git 笔记

一、windows:

我的gitHub主页 :https://github.com/CiaraGan ; https://gitee.com/CiaraGan

二、Linux

1.git创建仓库

(1)git init :初始化一个Git仓库

使用当前目录作为git仓库 :git init ---初始化

指定目录作为git仓库 :git init new\_repository ---会在new\_repository下出现.git目录

(2)git clone :从现有Git仓库中拷贝项目

git clone url direct

(3)git add :将文件添加到缓存区, 对其跟踪

git add . ---文件内容修改(modified)以及新文件(new)，但不包括被删除的文件

git add -A ---监控已经被add的文件

git add -u ---以上两个功能的合集

(4)git status :查看上次提交之后是否有修改

git status -s ---简短结果输出

(5)git diff :显示已写入缓存与已修改但尚未写入缓存的改动的区别

尚未缓存的改动 :git diff

查看已缓存的改动 :git diff --cached

查看已缓存的与未缓存的所有改动 :git diff HEAD

显示摘要而非整个diff :git diff --stat

(6)git commit -m :将缓存区内容添加到本地仓库中

(7)配置user name和 user email

git config --global user.name=”CiaraA” ---标记身份

git config --global user.email=”924417727@qq.com”---匹配GitHub账户

(8)git reset HEAD :取消已缓存的内容

git reset file

(9)git rm

git rm file ---从Git中移除某个文件

git rm -f file ---删除之前修改过并且已经放到暂存区域

git rm --cached file ---把文件从暂存区域移除, 但仍然希望保留在当前工作目录中

(10)git mv :移动或重命名一个文件/目录/软连接

git mv README README.md

(11)git branch :列出分支

git branch ---列出本地的分支

git branch branch\_name ---手动创建一个分支

git checkout branch ---切换分支

git checkout -b branch\_name ---创建新分支并立即切换到该分支下, 在该分支中操作

git branch -d branch\_name ---删除分支

git merge ---将任何分支合并到当前分支

(12)git log :查看历史提交记录

git log --oneline ---历史记录简洁版本

git log --oneline --graph

git log --reverse --oneline

(13)tag :标签

git tag -a v1.0 ---带注解的标签

git tag -a v0.9 85fc7e7 ---追加标签

git tag ---查看所有标签

git tag -a tag\_name -m "tag\_message" ---指定标签信息

git tag -s tag\_name -m "tag\_message" ---PGP签名标签

git tag -d v1.1 ---删除标签

git show v1.0 ---查看此版本修改内容

(14)git remote add :添加远程仓库

git remote add short\_name url

git remote ---查看当前远程库

git remote -v ---可以看到别名的实际链接地址

git fetch alias ---从远程仓库下载新分支与数据, 该命令执行完后需要执行git merge远程分支到所在的分支

git merge alias/branch ---从远端仓库提取数据并尝试合并到当前分支, 执行 git fetch 之后紧接着执行 git merge 远程分支到你所在的任意分支

git remote rm alias ---删除远程仓库

(14)git pull

git push alias branch ---

git pull --rebase origin master

(15)git push

git push alias branch ---推送新分支与数据到远端仓库

git push -u origin master