git学习

1. 基本操作

常见版本管理工具：svn、cvs、bitkeeper

概念：

工作区：版本控制的目录（用户编辑的文件）

暂存区：未提交的代码 .git/index（进行了跟踪但未提交的文件，add操作产生的文件）

版本库：.git（提交后的文件，commit产生的文件）

特点：开源分布式版本控制器

操作：

git init 在任意文件夹下执行该初始化版本库，会产生一个.git的隐藏文件夹，该文件夹（即为版本库）下index文件对应暂存区

git add filename 添加文件到暂存区

git commit -m “message” 提交暂存区文件到版本库

git reset --hard HEAD^ 重置版本未上次提交的版本（取消最近一次提交）

git status 查看暂存区的状态

git diff 查看版本改动

git commit –amend修改上一次提交

git reset HEAD filename 取消暂存文件

git checkout – filename 取消对文件修改（危险操作）

git remote -v 查看远程仓库

git remote add name remoteurl 创建远程仓库别名

git clone remote -url 从远程仓库拉取数据

git fetch origin 从远程仓库抓取本地没有地的数据

git push origin master 将本地代码提交到远程仓库主分支上

git tag 列出版本标签

git tag -a tagname -m ‘message’ 添加标签

git show tagname 显示标签详细信息

git tag tagname-lw 创建轻量级标签，轻量级标签实际上就是一个保存着对应提交对象的校验和信息的文件

git branch branchname 创建分支

git branch 查看分支

git branch -v 查看分支最后一个提交信息

git checkout branchname 切换到其他分支上

git checkout -b branchname 创建分支并在切换到该分支

git merge branchname 合并分支到当前分支

git branch -d branchname （合并完成之后）删除分支

git mergetool 合并产生冲突时使用该命令引导解决冲突

git branch –merged 查看当前已合并分支

git branch –no-merged 查看未合并分支进度

git push origin remotebranchname 推送本地分支到远程仓库（在已经进行clone操作的情况下）

git pull origin master从远程主分区拉取数据

git push origin branchname 将本地分支推送到远程服务器上

git checkout --track origin/branchname 跟踪远程分支

git checkout -b localbranchname origin/remotebranchname 跟踪远程分支并在本地创建别名分支

git push origin :remotebranchname 合并远程分支到master

git rebase master 分支基变，与merge操作类似，将当前分支作为master分支的下游分支

git rebase --onto master server client 将client、server合并到master分支上

1. 在idea中的配置使用
2. GitHub的使用
3. GitLab