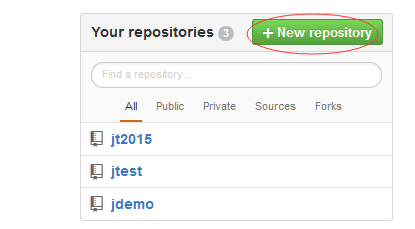
Git 学习笔记

1. 注册账号：<https://github.com/>
2. 在github上新建一个仓库



1. 安装Eclipse EGit插件，插件下载地址：[http://download.eclipse.org/egit/updates/](http://download.eclipse.org/egit/updates/" \t "_blank)
2. 安装git客户端



1. Git基础教程地址：

<http://git.oschina.net/progit/2-Git-%E5%9F%BA%E7%A1%80.html#2.1-%E5%8F%96%E5%BE%97%E9%A1%B9%E7%9B%AE%E7%9A%84-Git-%E4%BB%93%E5%BA%93>

GIT常用命令

|  |
| --- |
| $cd d:/workdir  ## 1>创建项目目录  $mkdir gitdemo  ## 2>项目初始化，创建版本库  $git init –git  ## 3>添加文件并提交  $git add –A  $git commit –m “initialized”  ## 4>为项目创建一个里程碑：v1  $git tag v1  ## 5>修改并提交文件  $git commit –a  ## 6> 将mirror版本库中的数据同步到本地master  $git pull mirror master  ## 7>提交到home版本库  $git push home  ## 8>文件查找  $git grep “git”  ## 9> 本地备份  $git clone gitbook gitbook-bak  ## 10> 差异比较  ## 显示工作区与提交任务（提交暂存区）之间的差异  $git diff  ## 显示工作区与HEAD（版本库）之间的差异  $git diff HEAD  ## 显示提交暂存区与HEAD（版本库）之间的差异  $git diff –cached  ## 显示版本A与版本B之间的差异  $git diff A B  ## 11> 撤销  ## 从暂存区撤销全部文件（或某些文件）  $git checkout . (or git checkout --filename)  ## 从版本库中撤销全部文件（或某些文件）注：该操作会同步撤销暂存区的内容  $git checkout HEAD .(or git checkout HEAD --filename)  ## 12>保存当前工作进度  $git stash  ## 12>恢复最新保存的工作进度  $git stash pop  ## 13> 查看提交日志  $git log –graph –oneline  ## 14>查看工作区的文件  $ls  ## 15>将版本库中最后一次提交的内容回滚  ## 硬回滚(注意：该命令十分危险，请慎用)  $git reset –hard HEAD^  # 软回滚，工作区和暂存区不改变，但是引用向前回退一次。当对最新的提交说明或者提交的更改不满意时，撤销最新的提交以便重新提交。  $git reset –soft HEAD^  ## 16>删除文件  $rm \*.txt //删除所有以txt为后缀的文件  $git rm demo.txt //暂存区中删除文件  ## 17>查看历史版本中的文件  $git ls-files –with-tree=HEAD^  $git cat-file –p HEAD^:demo.txt//查看历史版本中尚在的删除文件内容  ## 18>更新本地文件（修改、删除）到暂存区  $git add –u  ## 19>恢复删除的文件  $git cat-file –p HEAD^1:demo.txt > demo.txt  $git show HEAD^1:readme.txt > readme.txt  $git checkout HEAD^1 – detached-commit.txt  ## 20>查看里程碑的版本信息  $git describe //该命令有助于跟踪文件版本  ## 21> 文件归档  ## 对版本库建档  $git archive –o latest.zip HEAD  ## 将版本库中的src和 doc建档  $git archive –o partial.tar HEAD src doc  ## 基于里程碑v1.0建立归档  $git archive –format=tar –prefix=1.0/ v1.0 | gzip > foo-1.0.tar.gz  ## 22>图形化显示所有分支  $gitk –all  ## 23> 版本表示法：git rev-parse  ## 显示分支  $git rev-parse –symbolic –branches  ## 显示里程碑  $git rev-parse –symbolic –tags  ## 显示定义的所有引用  $git rev-parse –symbolic –glob=refs/\*  ## 显示HEAD对应的SHA1哈希值  $git rev-parse HEAD  ## 24>查看某个版本的所有历史提交  ## 查看版本A开始的所有历史提交  $git rev-list –oneline A  ## 25>文件追溯:git blame  ## 追溯README文件的所有变更记录  $git blame README  ## 只查看README的某几次的提交记录  $git blame –L 6,+5 README  ## 26> 二分查找：git bisect  ## 确认在master分支上  $git checkout master  ## 开始二分查找  $git bisect start  ## 查看当前分支是否存在doc/B.txt文件  $git cat-file –t master:doc/B.txt  ## 查看版本G下是否存在doc/B.txt文件  $git cat-file –t G:doc/B.txt  ## 假定当前分支存在该文件，G版本不存在。将当前分支标记为“坏提交”,G版本标记为“好提交”  $git bisect bad  $git bisect good G  ## master分支有问题，git自动定位到下个里程碑  ## describe 查看当前里程碑  $git describe    ## 查看C版本是否存在doc/B.txt文件  $git cat-file –t C:doc/B.txt  ## C版本下没有找到该文件，将当前版本标记为“好提交”  $git bisect good  ## 定位坏提交的版本  $git checkout bisect/bad  ## bug修复后，撤销二分查找  $git bisect reset  ## 27> 签出某个历史版本的某个文件  ## 如：git show A:README > README\_VA,< commit>为里程碑号  ## 如：git show 0cd7f2e:README > README\_VC< commit>为里版本号  $git show <commit>:<file> > new\_file |