协议 http/ssh

# git命令

git init

git status [-s]

git clone <https://.git>

git clone –b //签出特定分支

git add file

## git rm file//删除文件

git commit -m "description"

git push

git pull

git diff file

git log [--pretty=oneline]

git reflog//历史命令

git中HEAD表示当前版本，上一版本是HEAD^，上上版本HEAD^^,或者HEAD^2，或者commit\_id

git reset --hard HEAD^

git checkout -- file 就是让这个文件回到最近一次git commit或git add时的状态,没有--，就变成了“切换到另一个分支”的命令

git reset HEAD file 可以把暂存区的修改撤销掉

## git remote –v //查看远程仓库地址

git remote

git remote add origin git@github.com:zhuanghu/learngit.git

git push -u origin master 由于远程库是空的，我们第一次推送master分支时，加上了-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。以后可以使用git push origin master

(要关联一个远程库，使用命令git remote add origin git@server-name:path/repo-name.git；

关联后，使用命令git push -u origin master第一次推送master分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令git push origin master推送最新修改；)

使用https除了速度慢以外，还有个最大的麻烦是每次推送都必须输入口令

git checkout -b dev 创建并切换分支

git checkout -b dev origin/dev

git branch dev 创建分支

git branch -d dev删除分支

git branch -D <name> 强行删除一个没有合并的分支

git checkout dev 切换分支

git branch 查看当前分支。

git branch --set-upstream branch-name origin/branch-name 设置dev和origin/dev的链接

git pull

git merge dev 合并指定分支到当前分支

用git log --graph命令可以看到分支合并图

合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出来曾经做过合并。

git stash 储存当前工作区

git stash apply [stash@{0}] 恢复工作区储存【指定版本】

git stash drop 删除存储

git stash pop 恢复并删除

git tag <name> 打一个新标签

git tag 查看所有标签

git tag v0.9 6224937 对commit\_id打标签

git tag -a v0.1 -m "version 0.1 released" 3628164 用-a指定标签名，-m指定说明文字

git show <tagname>可以看到说明文字

-s用私钥签名一个标签，签名采用PGP签名，因此，必须首先安装gpg（GnuPG），如果没有找到gpg，或者没有gpg密钥对，就会报错

git tag -d v0.1

git push origin <tagname> 推送某个标签到远程

git push origin --tags 一次性推送全部尚未推送到远程的本地标签

如果标签已经推送到远程，要删除远程标签就麻烦一点，先从本地删除。然后，从远程删除。删除命令也是push，git push origin :refs/tags/v0.9

git config --global color.ui true 让Git显示颜色

README.MD markdang语法

## git本地代码关联远程仓库

git init

git remote add origin <https://github.com/jwei2006/grpc-n1.git>

## 提交代码

git add .

git commit –m “remark”

git push [origin master] git push origin master的意思就是上传本地当前分支代码到master分支

### 提交时代码没有关联解决

git push --set-upstream origin master

# git免密码push

<http://www.cnblogs.com/ballwql/p/3462104.html>

Linux或者Mac下方法：

创建文件，进入文件，输入内容：

cd ~

touch .git-credentials

vim .git-credentials

https://{username}:{password}@github.com

在终端下输入：

git config --global credential.helper store

打开~/.gitconfig文件，会发现多了一项:

[credential]

helper = store

Windows方法：

方法同上面，只是第一步创建git-credentials有点不同。在%HOME%目录中，一般为C:\users\Administrator，也可以是你自己创建的系统用户名目录，反正都在C:\users\中。文件名为.git-credentials,由于在Window中不允许直接创建以"."开头的文件，所以需要借助git bash进行，打开git bash客户端，进行%HOME%目录，然后用touch创建文件 .git-credentials, 然后后面的操作同上面。