

create a new repository

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/javadev121/jvm.git

git push -u origin master

push an existing repository

git remote add origin https://github.com/javadev121/jvm.git

git push -u origin master

* git clone

1. *git clone <版本库的网址>*
2. *git clone <版本库的网址> <本地目录名>*
3. $ git clone **http**[s]:*//example.com/path/to/repo.git/*
4. $ git clone **ssh**:*//example.com/path/to/repo.git/*
5. $ git clone **git**:*//example.com/path/to/repo.git/*
6. $ git clone /opt/git/project.git
7. $ git clone **file**:*///opt/git/project.git*
8. $ git clone **ftp**[s]:*//example.com/path/to/repo.git/*
9. $ git clone **rsync**:*//example.com/path/to/repo.git/*

**常来说，Git协议下载速度最快，SSH协议用于需要用户认证的场合。**

* git remote

***git remote***  *列出所有远程主机。*

origin

***git remote -v*** *查看主机网址*

origin git@github.com:jquery/jquery.git (fetch)

origin git@github.com:jquery/jquery.git (push)

注意：克隆版本库的时候，所使用的远程主机自动被Git命名为origin。

如果想用其他的主机名，需要用git clone命令的-o选项指定。

$ git clone -o jQuery https://github.com/jquery/jquery.git

$ git remote

jQuery

***git remote show***  *命令加上主机名，可以查看该主机的详细信息。*

$ git remote show <主机名>

***git remote add*** *命令用于添加远程主机。*

$ git remote add <主机名> <网址>

***git remote rm****命令用于删除远程主机。*

$ git remote rm <主机名>

*git remote rename命令用于远程主机的改名。*

$ git remote rename <原主机名> <新主机名>

* git fetch

***git fetch****将某个远程主机的更新，全部取回本地。*

**$ git fetch <远程主机名>** #取回全部分支branch的更新

**$ git fetch <远程主机名> <分支名>**  #取回特定分支更新。

**$ git fetch origin master**   **#取回origin主机的master分支。**

* git pull

***git pull*** *取回远程主机某个分支的更新，再与本地的指定分支合并.*

**$ git pull <远程主机名> <远程分支名>:<本地分支名>**

**$ git pull origin next:master** #取回origin主机的next分支，与本地master分支合并

**$ git pull origin next** #取回origin主机的next分支，与当前分支合并

=$ git fetch origin $ git merge origin/next

* git push

***git push*** *将本地分支的更新，推送到远程主机*

**$ git push** <**远程主机名**> **<本地分支名>:<远程分支名>**

git push -u origin master #如果当前分支和多个主机之间存在追踪关系，可以使用这个命令来设置一个默认的主机

**git clone: 从远程主机克隆一个版本库**

**$ git clone [url] #下载一个项目**

**$ git add [file1] [file2] ... # 添加指定文件到暂存区**

**$ git add .**  **# 添加当前目录的所有文件到暂存区**

**$ git commit -m [message] # 提交暂存区到仓库区**

**$ git commit [file1] [file2] ... -m [message] # 提交暂存区的指定文件到仓库区**

**$ git branch** **#列出所有本地分支**

**$ git branch -a**  **#列出所有本地分支和远程分支**

**$ git branch -r**  **#远程分支**

**$ git branch [branch-name] #新建一个分支**

**$ git branch -d [branch-name] # 删除分支**

**本地分支与远程分支之间，建立一种追踪关系（tracking）**

**$ git branch --set-upstream master origin/next** #master分支追踪origin/next分支

**$ git checkout -b [branch] = git branch aaa 和git checkout aaa #新建一个分支，并切换到该分支**

**$ git checkout -b newBrach origin/master #在远程分支origin/master的基础上，创建一个分支**

**$ git checkout [branch-name]** #切换到指定分支，并更新工作区

**$ git checkout -** #切换到上一个分支

**$ git merge origin/master 或者 $ git rebase origin/master** #在本地分支上合并远程分支