git init //得到一个git仓库。被git初始化过的这个仓库的

作用：便于git对这个仓库（目录）中的文件进行版本管理

git add 文件名 //把文件添加到提交列表

git commit -m “添加修改说明” //把文件的版本修改提交到仓库

git push //把本地仓库推送到github远程云端上，新仓

库第一次push时，需要把本地git仓库和GitHub账号上创建的仓库进行相关联

一些概念：

git：分布式版本控制系统（抽象出来的一个系统，比如你自己搭建的多人协作，多备份的系统）

git-scm.com：git官方网站

GitHub：是一个公开的，免费代码托管平台，你可以把你的代码托管在上面，类似于CSDN的码云，或者是你公司内部的server

git在windows操作系统上的客户端名字：Cygwin、msysGit、GitHub for windows

git在linux操作系统上的客户端名字：git

TortoiseGit：是一个GUI，基于msysGit的图形化界面，在该图形界面上，调用的是msysGit的内部命令，所以，在使用它之前，需要先安装msysGit

如果我仅对本地仓库中的文件进行更改，而不add，会怎样呢，add后，不commit，会怎样呢？



一张图理解，把本地目录创建为git仓库后，该git仓库如何管理更改的各版本。

把本地目录创建为git仓库，本地该目录中可见的文件为工作区，同时该目录下有一个隐藏的.git目录（随本地目录init为git仓库而创建），该.git目录，名字称为版本库，顾名思义，这个版本库中保存的信息，即为文件各版本更改信息。

在git bash中，切换到工作区目录，对工作区中文件进行更改，修改完成后，经过git add把更改的文件提交到版本库中的stage（暂存区），再经过git commit把修改的文件信息提交到当前分支上（把提交的修改信息，按照时间线串联起来，即，该分支为按照时间排列的，被提交修改内容的串联，时间为横轴，把修改内容依次往后排列的一条线）

而，为什么要设置这个暂存区stage呢？

如果我把本地仓库和远程仓库相关联了，本地版本库中的每一次修改是不是都会被同步到远程仓库中，不管是添加操作还是删除操作？

看来额外分支是有非常大作用的，还有暂存区可以无记录地撤销更改，并可以简缩多次繁碎的commit （额外分支好像对这个问题并没有多大的帮助）

同时，git中有很多支持stage的指令，以便更好的利用这个区间

还有，这些都是先在本地创建一个git仓库，然后把内容推到远程仓库中，每一次新建一个git仓库，并执行第一次推送时，都要与远程仓库做关联，在之后的推送到远程仓库操作可以直接用git push，表示把本地版本库中的master分支，提交到远程仓库。

那么，如果我是从远程仓库中克隆的一个本地版本库，操作是不是和上面一样，还需不需要额外git push，还是自动同步？

第一次关联时可能出现的问题及解决办法：

<http://blog.csdn.net/sssssuuuuu666/article/details/78565381>

git：分布式版本控制系统

虽然分布式版本控制系统同样有一个充当中央服务器的电脑，但是和集中式不同的是，每个人的本地仓库都有完整的版本修改信息，即，中央服务器挂了，仍然有大量的本地记录保存，完全可以再设定一个中央服务器，再次执行推送，而这些是集中式版本控制所不具备的，集中式版本控制系统，一旦服务器挂了，所有的都gg。当然分布式版本控制系统的中央服务器可以由你自己搭建git server，用于保存提交的文件和修改，同时这个git server也可以是公司的server，24小时开机，能够让你从这个server中实时push，pull或clone到本地仓库；当然，github也是一个不错的server。虽然上面的东西都只能公开，并有一定的容量限制。

git还有一个很大的有点是：分支管理

作用：