$ git init

把目录变成仓库

自动创建master分支

$ git add readme.txt

readme.txt是目录下的一个文件，这样git才能找到

$ git commit -m "wrote a readme file"

-m后面是注释

* 如果提交的是一个.txt文件，显示

$ git commit -m "first bbb"

[master (root-commit) 974aa82] first bbb

1 file changed, 2 insertions(+)

create mode 100644 bbb.txt

* 如果提交一个二进制文件

$ git commit -m "first commit"

On branch master

Initial commit

Untracked files:

Ajax.docx

nothing added to commit but untracked files present

git status

命令查看当前状态

git diff readme.txt

看txt文件哪儿被修改了

$ git add file1.txt

$ git add file2.txt file3.txt

$ git commit -m "add 3 files."

可以一次commit很多文件

* Untracked files未跟踪文件

在目录里新建一个文件啥都不干就是这个状态

* Changes to be committed

untracked files执行add命令之后的状态

add命令就是把文件修改添加到暂存区stage

可以用git rm --cached README.txt回到untracked files状态

* changes not staged for commit

上面这个状态，不commit，修改.txt文件后变成这个状态

* commit

把暂存区的所有内容提交到当前分支

$ git log

$ git log --pretty=oneline

commit a6554cb22dc6b92e972174a1bd9706dac60685ac (HEAD -> master)

Author: Nfboys <1033276810@qq.com>

Date: Sat Nov 3 12:49:55 2018 +0800

add GPL

commit 43784d6debff264fb9e1a1d0bb433a90cb580fbe

Author: Nfboys <1033276810@qq.com>

Date: Sat Nov 3 12:48:52 2018 +0800

bbb2

commit 974aa828f5a35afc6d85c4971dcf0dc35f223017

Author: Nfboys <1033276810@qq.com>

Date: Sat Nov 3 12:35:39 2018 +0800

first bbb

最上边是commit id版本号

git reset --hard HEAD^

回退到上一个版本

git reset --hard 1094a

回退到某个版本，a6554是版本号前5位

$ git reflog

记录每一次命令

git checkout -- readme.txt

这个文件回到最近一次git commit或git add时的状态

git reset HEAD <file>

把暂存区的修改撤销掉（unstage），重新放回工作区

git rm bbb.txt

删除

git checkout -- test.txt

如果你用文件管理器删的，还想恢复，则用这个命令

在github上新建一个repository

假如叫temp

$ git remote add origin [git@github.com:Nfboys/temp.git](mailto:git@github.com:Nfboys/temp.git)

把远方的repository关联到当前的库，远程库的名字就是origin

* $ git push -u origin master

把本地库所有内容推送到origin

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。也就是说第一次推送加上-u参数，以后不加。

$ git push origin master

如果先有远程库

git clone git@github.com:Nfboys/temp.git