## 创建版本库

$ git init

把当前目录变成仓库，多了一个.git的目录

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 add -A

add所有

$ git commit -m "wrote a readme file"

-m后面是注释

* 如果commit一个.txt文件，显示

[master 01671b3] wrote a readme file

1 file changed, 1 insertion(+)

表示文本文件增加了一行

* 如果提交一个二进制文件

[master 888cc49] commit a picture

1 file changed, 0 insertions(+), 0 deletions(-)

git status

命令查看当前状态

* 所有文件已commit

On branch master

nothing to commit, working tree clean

* 有文件被修改，没有add也没有commit

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: aaa.txt

no changes added to commit (use "git add" and/or "git commit -a")

显示 未经任何处理的changes，也就是没有add过的changes

* 有文件被修改，已经add，没有commit

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: aaa.txt

显示 待commit

git diff readme.txt

修改过但没有add的文件，可以通过diff命令查看具体修改了什么内容

* 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

* changes not staged for commit

修改文件后未经任何处理的状态

* commit

把暂存区的所有内容提交到当前分支

$ git log

git log命令显示提交日志，也就是历史commit的版本号、时间、作者、注释

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

## 版本回退

HEAD表示当前版本，也就是最新的提交a6554cb22dc6b92e972174a1bd9706dac60685ac。上一个版本是HEAD^，上上一个版本是HEAD^^，往上100个版本是HEAD~100。

git reset --hard HEAD^

reset命令回退到上一个版本。亲测不行，报错，git reset –hard HEAD^^可以退回到上一版本

git reset --hard 1094a

回退到某个版本，1094a是版本号前5位，至少写版本号前5位

$ git reflog

reflog记录每一次命令

24e1c23 (HEAD -> master) HEAD@{0}: reset: moving to HEAD^

01671b3 HEAD@{1}: reset: moving to HEAD^

888cc49 HEAD@{2}: reset: moving to HEAD^

e4adb6b HEAD@{3}: commit: addBBB

前面e4adb6b是版本号的前7位，可供reset

* 工作区（Working Directory）

就是根目录，.git文件的父目录

* 版本库（Repository）

.git目录

Git的版本库里存了很多东西，其中最重要的就是称为stage/index的暂存区，还有Git为我们自动创建的第一个分支master，以及指向master的一个指针HEAD。

把文件往Git版本库里添加的时候分两步执行

第一步git add把文件添加进去，实际上就是把文件修改添加到暂存区；

第二步git commit提交更改，实际上就是把暂存区的所有内容提交到当前分支。

修改——add——暂存区——commit——分支

## 撤销修改

git checkout -- readme.txt

命令git checkout -- readme.txt意思就是，把readme.txt文件在工作区的修改全部撤销，这里有两种情况：

* readme.txt修改后还没有被放到暂存区，也就是没有git add，则撤销修改就回到和版本库一模一样的状态；
* readme.txt已经通过git add添加到暂存区后，又作了修改，则撤销修改就回到添加到暂存区后的状态。

总之，git checkout命令就是让这个文件回到最近一次git commit或git add时的状态。

* 情况1
* 新建文件readme.txt
* 加一行aaaaaa
* git checkout
* 文件变为空
* 情况2
* 新建文件readme.txt
* 加一行aaaaaa
* git add
* 加一行bbbbb
* git checkout
* 文件变为aaaaa，没有bbbbb

git reset HEAD readme.txt

git reset HEAD <file>可以把暂存区的修改撤销掉（unstage），重新放回工作区：

git reset命令既可以回退版本，也可以把暂存区的修改回退到工作区。HEAD表示最新版本。

* 新建readme.txt
* 加一行aaaaaaaa
* git add
* git reset HEAD readme.txt
* 现在readme.txt里还有aaaaaaa，但是暂存区是空的，也就是没有经过git add的状态

## 删除文件

直接在文件管理器中删除一个a.txt之后，要用命令git rm删掉，并且git commit：

git rm a.txt

还可以还原误删的文件在此不解释，有兴趣回头看

## 远程仓库

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 pull origin master // 把本地仓库的变化连接到远程仓库主分支

有时候远程库里已经有内容，直接push会报错，所以在add origin和push操作之间加一步pull

git remote

查看远程库的信息

git remote -v

显示形如

origin https://gitlab.com/tjuse/tjqs3.git (fetch)

origin https://gitlab.com/tjuse/tjqs3.git (push)

可以抓取fetch和推送push的远程库地址。如果没有push权限，就看不到push地址

$ 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

把远程库下载到本地并关联