

cd :是进入了现在所在位置
mkdir XX:创建一个空目录 XX指目录名
pwd:显示当前目录的路径
git init:把当前的目录变成可以管理的git仓库,生成隐藏的.git文件
touch xx:新建xx文件文件
git add xx:把xx文件添加到暂存区
git commit -m "xx" a.txt :提交文件 -m后面的是注释
git status:查看仓库状态
git log:查看历史记录
git reset --hard HEAD^:网上回退一个版本
cat xx:查看xx文件内容
git reflog:查看历史记录的版本号id
git checkout -- xx:把xx文件在工作区的修改全部撤销,版本撤销
git rm xx:删除xx文件 之后要commit
git remote add origin https://github.com/qiuhaifeng01/a.git
关联一个远程库
git push -u(第一次要用-u以后不用) origin
master:把当前master分支推送到远程库
git clone https://github.com/xxxxxx 从远程库中克隆
git checkout -b dev:创建dev分支 并切换到dev分支上
git branch:查看当前所有的分支
git checkout master:切换回master分支
git merge dev:在当前分支合并dev分支
git branch -d dev:删除dev分支
git branch xxx:创建分支xxx
git remote:查看远程库信息
git remote -v查看远程库的详细信息
git push origin master:git会把master分支推送到远程库对应的分支上

创建版本库

cd /F/Git:先进如某个盘中

mkdir 名字:创建一个空目录 XX指目录名

git init Git:将Git初始生成可以管理的git仓库,生成隐藏的.git文件

创建文件:

git add a.txt / touch a.txt: 添加文件到库中

git commit -m “这里要打备注此次的行动” a.txt

修改文件:

vim a.txt:打开编辑框

:wq ! :退出编辑框

git add a.txt / touch a.txt: 添加文件到库中

git commit -m “这里要打备注此次的行动” a.txt

时光穿梭:

指向想要的版本

git log:先看最近的记录

git reflog:查看历史记录版本号id

git reset --hard HEAD^:网上回退一个版本

回到指定版本:

git reflog:查看历史记录版本号id

git reset --hard 69738e3:回到指定版本

在工作区修改的全部撤销

git checkout -- a.txt

1. 还没有放在暂存区,撤销就回到原来的样子
2. 已经添加到暂存区,又做了修改,撤销会回到暂存区后的状态
总之就是回到最近一次git commit 或者git add的状态

删除文件

`git rm a.txt` :删除文件

`$ git commit -m"ss"`要提交

理解工作区+版本库+暂存区

工作区(working Directory):就是你电脑本地硬盘目录

版本库(Repository):工作区有个隐藏目录.git, 他就是Git的本地版本库

暂存区(stage):一般存放在“git目录”下的index文件(.git/index)中, 所以
我们把暂存区有时也叫作索引(index)。

Git为我们自动创建的第一个分支master, 以及只想master的一个指针
叫HEAD

第一步是用“git

add”把文件纳入Git管理, 实际是把本地文件修改添加到暂存区

第二部使用“git

commit”提交更改, 实际上是把暂存区的所有内容提交到当前分支。

因为我们创建Git版本库时, Git自动为我们创建了唯一一个master分支,
所以commit就是往master分支上提交更改。

用‘git diff HEAD --

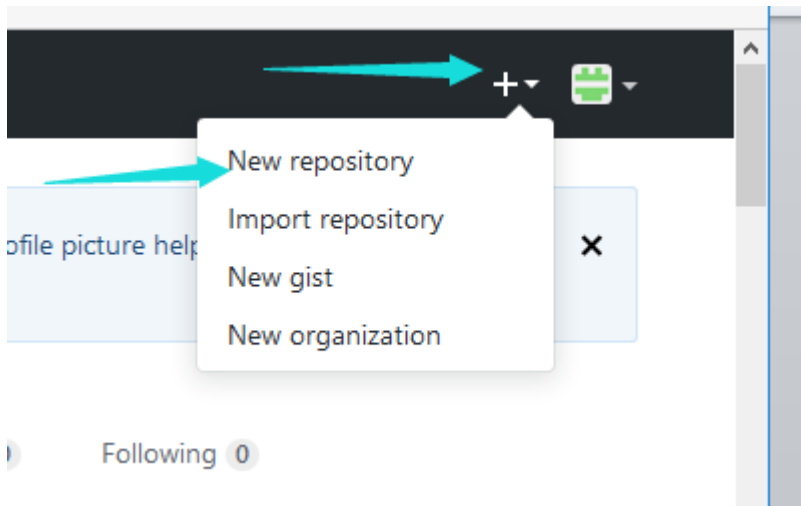
filename’命令可以查看工作区和暂存区里边最新版本的差别。

新建过撤销未add: `git checkout -- 文件名`

撤销已add未commit: 先`git reset HEAD 文件名` 再 `git checkout -- 文件名`

撤销已add已commit: `git reset --hard HEAD^`

push文件到github上



在加号加new一个repository

Owner: wxx083011 / Repository name:

Great repository names are short and memorable. Need inspiration? How about [expert-octo-giggle](#).

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

填写完整

Quick setup — if you've done this kind of thing before

or

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# wxx" >> README.md
git init
git add README.md
git commit -m "first commit"
```


创建之后，复制HTTPS的网址，

在这个文件里打开Git bush,

```
git remote add origin https://github.com/wxx083011/Git.git
```

```
$ git push -u origin master
```

输入账号密码



```
admin@DESKTOP-3OV7T6V /E/Git (master)
$ git push -u origin master
Username for 'https://github.com': wxx083011
Password for 'https://wxx083011@github.com':
Counting objects: 8, done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3) done.
Writing objects: 100% (8/8), 573 bytes | 0 bytes/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/wxx083011/Git.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
admin@DESKTOP-3OV7T6V /E/Git (master)
```

这样就成功了