命令行方式：

任意目录（建议开发根目录） 右键 > Git Bash Here

# 基本操作

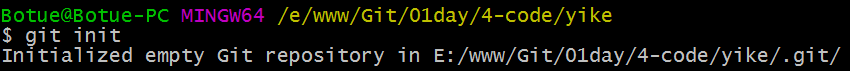
## 初始化和建立项目

### 初始化仓库

我们如果想要利用git进行版本控制，需要将现有项目初始化为一个仓库，或者将一个已有的使用git进行版本控制的仓库克隆到本地。

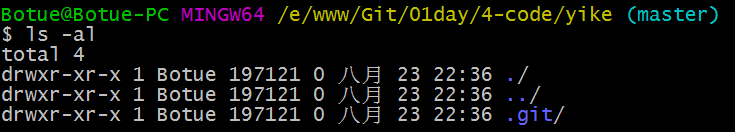
#### git init

**git init**



git init 只是创建了一个名为 .git 的隐藏目录，这个目录就是存储我们历史版本的仓库，

**ls -al 可以查看。**

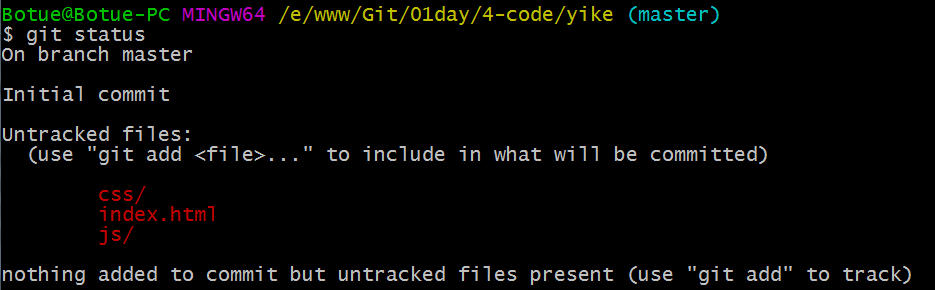


### 查看文件状态

初始化仓库后便可以进行开发了，进入到刚刚创建好并初始为仓库的目录，添加我们开发需要的文件。

git status 可以检测当前仓库文件的状态

**git status**



注：

git----会忽略空的目录

 git status -----列出当前目录所有还没有被git管理的文件和被git管理且被修改但还未提交(git commit)的文件，也就是所有改动文件，红色字体标出。

### 添加文件到暂存区

假设经过一段时间的开发后，需要把已开发的部分存起来，使用git add 添加到暂存区。

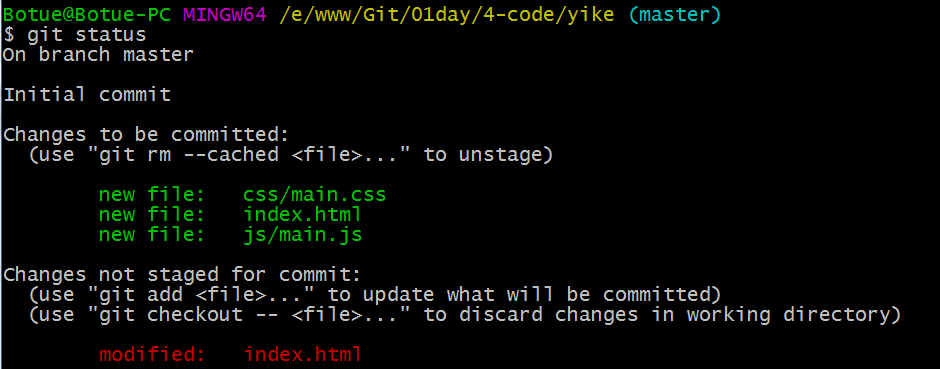
git add 文件名/文件路径 **“\*”/ -A / .**  代表所有

**git add .**

再次，**git status**

放到暂存区的文件被标记成了绿色，等待提交。

注：颜色是工具给添加的，目的是增加可读性并不是git统一的。



如果index.html再次被修改了，并被标记了红色。

### 撤销文件

又经过一段时间后发现新开发的部分有Bug，想要回到之前状态，可以使用

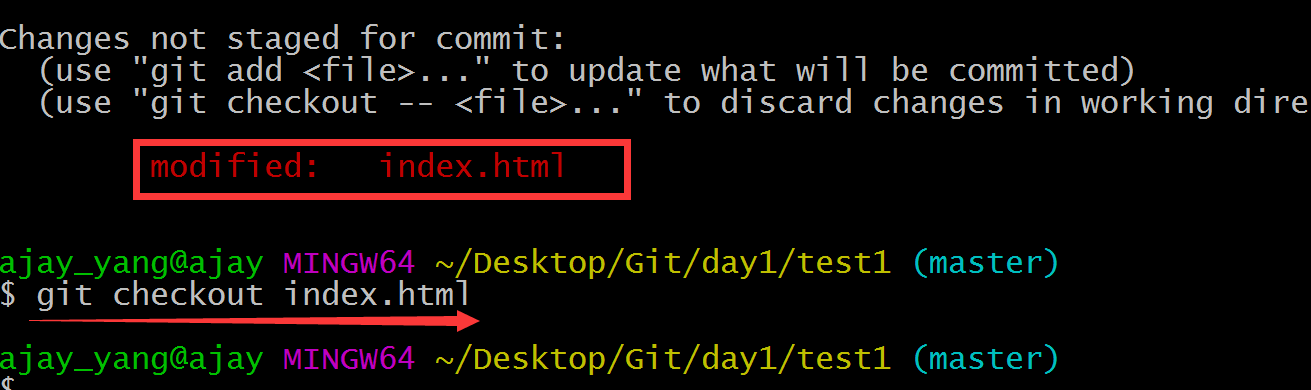
**git checkout 文件名**





**git status**

**git checkout index.html**



回到上次的工作内容上

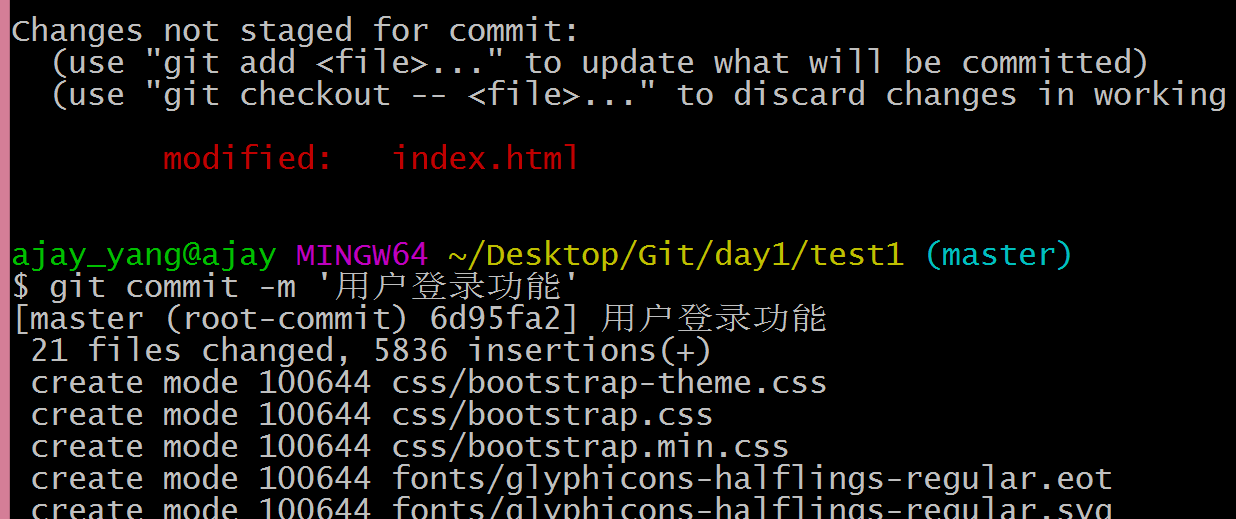


注：从暂存区还原原到工作区

### 提交文件

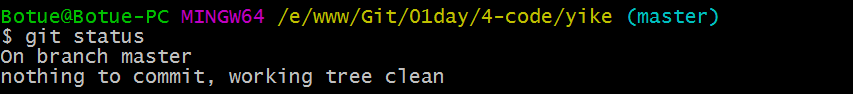
经过一个相对较长阶段开发或者一个功能开发完成了，就可以提交到本地仓库了，永久保存了。

**git commit -m '备注信息'**



将暂存区被标记成绿色的文件，全部提交到本地仓库存储。

这时git status查看状态



没有什么可提交的，变的很干净。

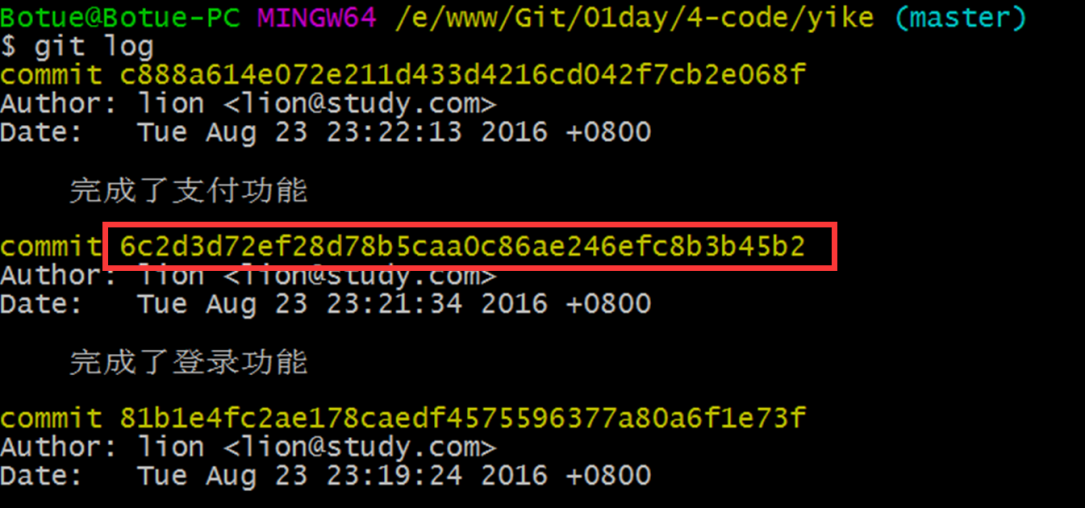
然后就可以进行反复的修改提交了。



### 查看提交历史

反反复复开发了很多的功能了，通过git log查看一下提交的历史。

**git log**



### 查看提交记录

**commit 81b1e4fc2ae178caedf4575596377a80a6f1e73f**

代表一次提交的唯一ID，一般称为SHA值。

傻

这时可以查看到更多提交历史。

这时关掉所有目录甚至关机！

### 恢复上一次提交的状态

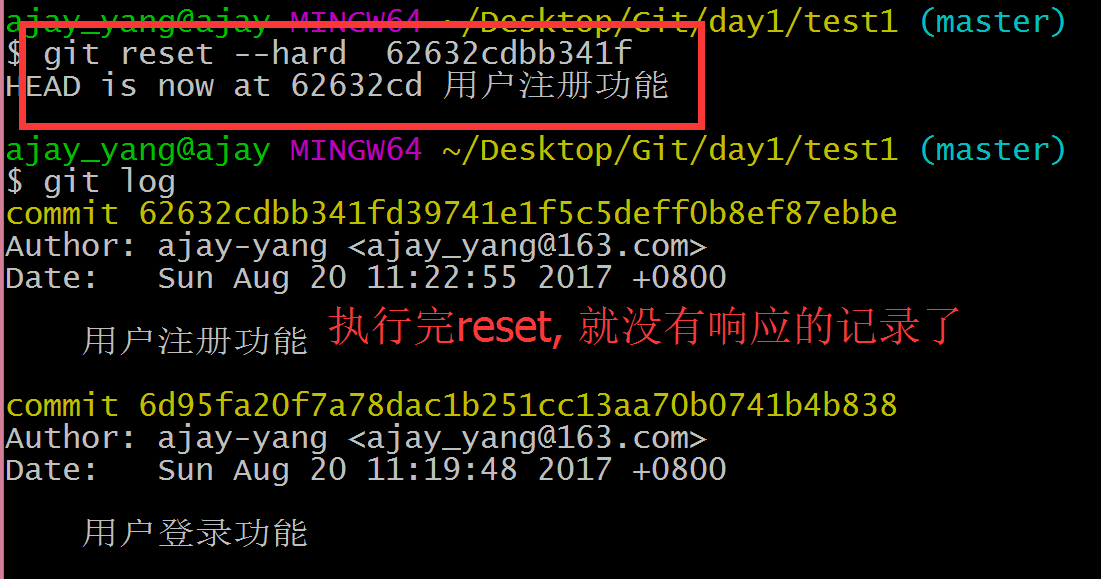
通过SHA值可以回到之前某一次的提交（时光倒流）

**git reset --hard c888a614e072e2**

git log再次查看发现最后一次提交成了支付功能了



### git checkout 与git reset 的区别：



git checkout -- file；

撤销对工作区修改；这个命令是以最新的存储时间节点（add和commit）为参照，覆盖工作区对应文件file；暂存区还有响应的记录。   
　　git reset --hard -- file；

清空add命令向暂存区提交的关于file文件的修改（Ustage）；

这个命令仅改变暂存区，也改变工作区，原来的记录就彻底没有了。

**git init**

**//--修改添加文件**

**git status**

**git add .**

**git commit -m ‘备注信息’**

**git status**

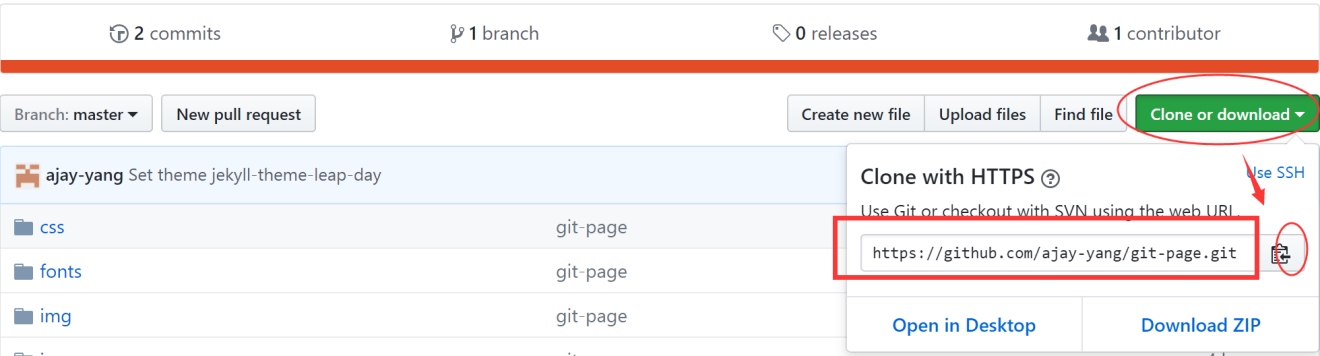
**git add .**

**.......**

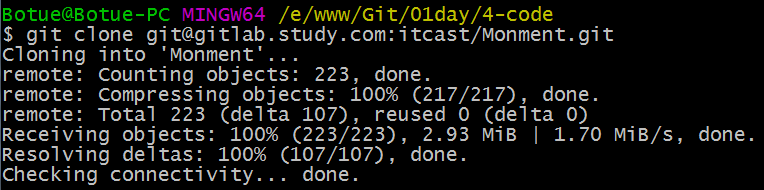
## 克隆创建项目

### 假如公司已有项目用了Git clone，那我们就利用克隆

新建文件夹Git用来存放github上的仓库，



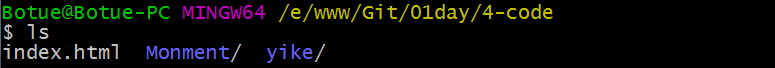
**git clone 仓库地址**



执行完这个命令，这个便是已有一个使用Git管理的项目。就当于github仓库跟你的电脑本地仓库建立一个连接。

ls 查看项目文件

**ls**



**添加文件**

**修改文件**

**git status**

**git add .**

**git commit –m ‘备注信息’**