简单教程

Git是分布式版本控制系统，那么它就没有中央服务器的，每个人的电脑就是一个完整的版本库，这样，工作的时候就不需要联网了，因为版本都是在自己的电脑上。既然每个人的电脑都有一个完整的版本库，那多个人如何协作呢？比如说自己在电脑上改了文件A，其他人也在电脑上改了文件A，这时，你们两之间只需把各自的修改推送给对方，就可以互相看到对方的修改了。推送一般通过远程仓库来实现，而gitHub就是一个远程仓托管平台。

1. 建仓库

Git在windows环境下有两个命令输入窗口

（在开始搜索可以找到）

Git GUI—可视化图形界面

Git CMD—我也不知道是啥

Git Bash—命令行窗口

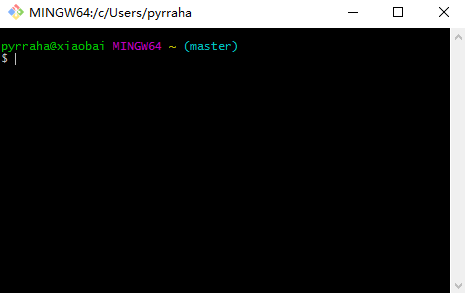
（后来我又下载了一个中文图形化的git界面软件，SourceTree，感觉也不错，这个工具就是用来简化操作的，不用天天手动输入命令）



命令输入窗口是用来建仓库用的



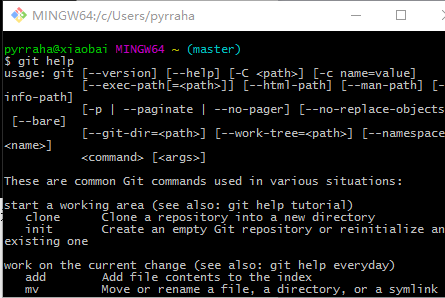
打开Git Bash



通过输入命令可以建库和对仓库里面的东西做一系列不可描述的事情

刚刚安装好git的时候还要对git进行检验 就是像java那样 看看有没有安装成功

即输入git help加回车可查看一下git命令参数



安装成功之后呢，还需要最后一步设置，在命令行输入（其实那个$都会自动有的，就不用输了）：

$ git config --global user.name "Your Name"（“”里面填用户名我的用户名好像是自动生成的）

$ git config --global user.email "email@example.com"（填邮箱，尽量不要用qq邮箱，用163或者其他会好一点）

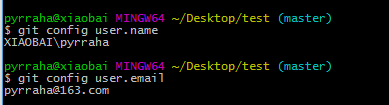
因为Git是分布式版本控制系统，所以，每个机器都必须自报家门，设置用户名跟邮箱。注意，git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。

设置好后可以再输入

$ git config user.name

$ git config user.email

来查看刚刚输入的信息



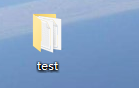
然后来愉快地建本地仓库

有一种方法是通过cd 啥的把一个目录路径找出来创建仓库

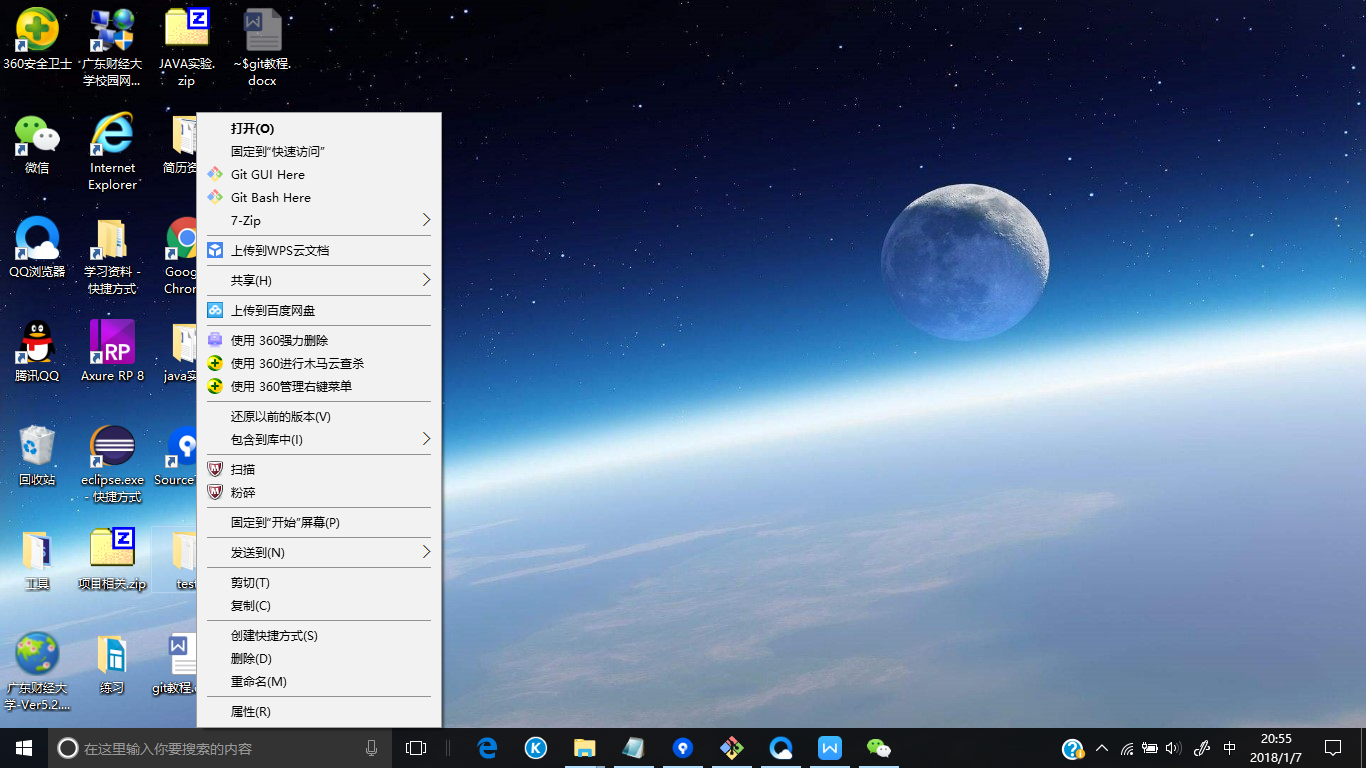
输入$ pwd会显示你的仓库的路径

另一种方法更简单一点

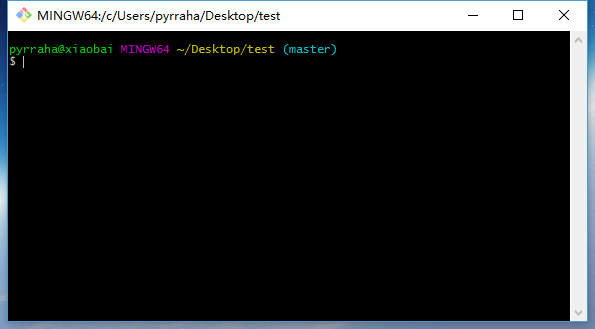
先在桌面上新建个test文件



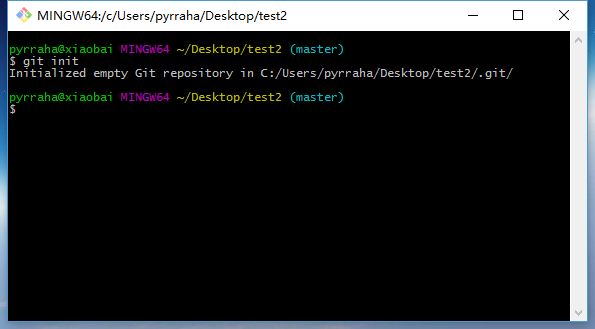
然后按红色框框那里



然后弹出下面这个东西

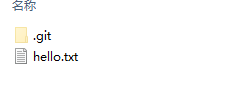


然后输入$ git init，出现下面那行就是已经建立了一个空仓库



在创建的test文件夹里面 创建一个hello.txt

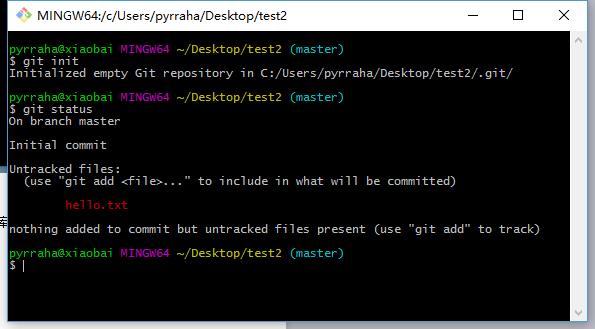
.git那个隐藏文件不能随便更改里面的内容 否则会破坏仓库



接下来是一系列骚操作

①检查状态 – git status

Git status是另一个非常重要的命令，它反馈给我们仓库当前状态的信息：是否为最新代码，有什么更新等等。在我们新建的仓库中执行git status会得到以下内容:



反馈信息告诉我们，hello.txt尚未跟踪，这是说这个文件是新的，git不知道是应该跟踪它的变动还是直接忽略。为了跟踪我们的新文件，我们需要暂存它。

②暂存 – git add

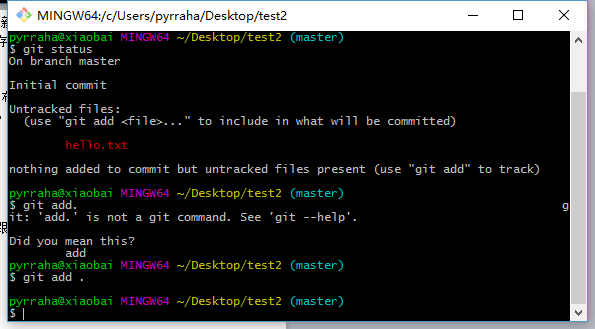
Git有个概念叫“暂存区“，你可以把它看成一块空白的画布，包裹着所有你可能会提交的变动。它一开始是空的，可以通过 git add 命令添加内容，最后使用 git commit 提交（创建一个快照）。

这个例子中只有一个文件，让我们可以先add它：

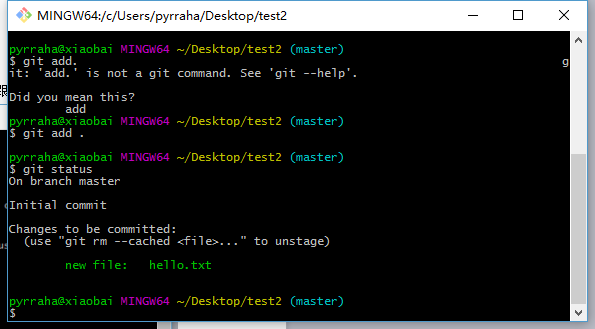
$ git add hello.txt

如果需要提交目录下的所有内容，也可以这样做：

$ git add .(注意这里还有个点，就是add后面跟个空格再跟个点，刚刚就不小心打错了)



再次使用git status查看状态试试：



这说明我们的文件已经准备好可以提交了。状态信息还告诉我们暂存区文件发生了什么变动，这里我们新增了一个文件，同样可以做修改和删除。取决于我们在上一次git add之后发生了什么。

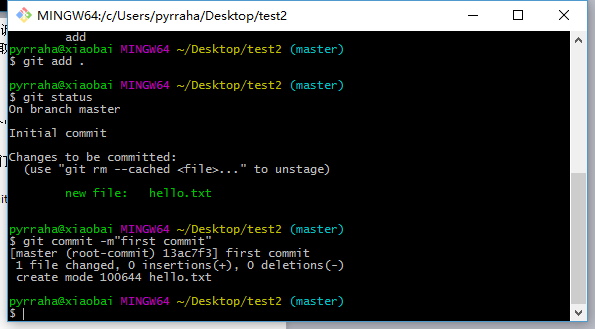
③提交 – git commit

一次提交代表着我们的仓库到了一个新的状态，就像是一个快照，允许我们像使用时光机一样回到之前的某个时间点。

创建提交，需要我们至少在到暂存区有一次修改（刚才我们做了git add），然后输入命令：

$ git commit -m "Initial commit."

这就创建了一次从暂存区的提交（加入hello.txt），-m “Initial commit.”是用户对这次提交的描述，建议写成有意义的描述性信息。



还有很多其他操作如版本回退撤销等，详见https://www.jianshu.com/p/ea9b7e44d731和其他教程

### 二、远程仓库

### 到目前为止，我们的操作都是在本地的——只存在于.git文件中。为了能够协同开发，我们需要把代码部署到远程仓库服务器上，这时候要用到github，先上官网注册一个账号。

### 创建GitHub账号

### 1. 打开GitHub官网。

### 2. 点击绿色按钮Sign up for GitHub，填写用户名，邮件地址和密码。

### 注意：

### 用户名只能包含字母和"-"字符

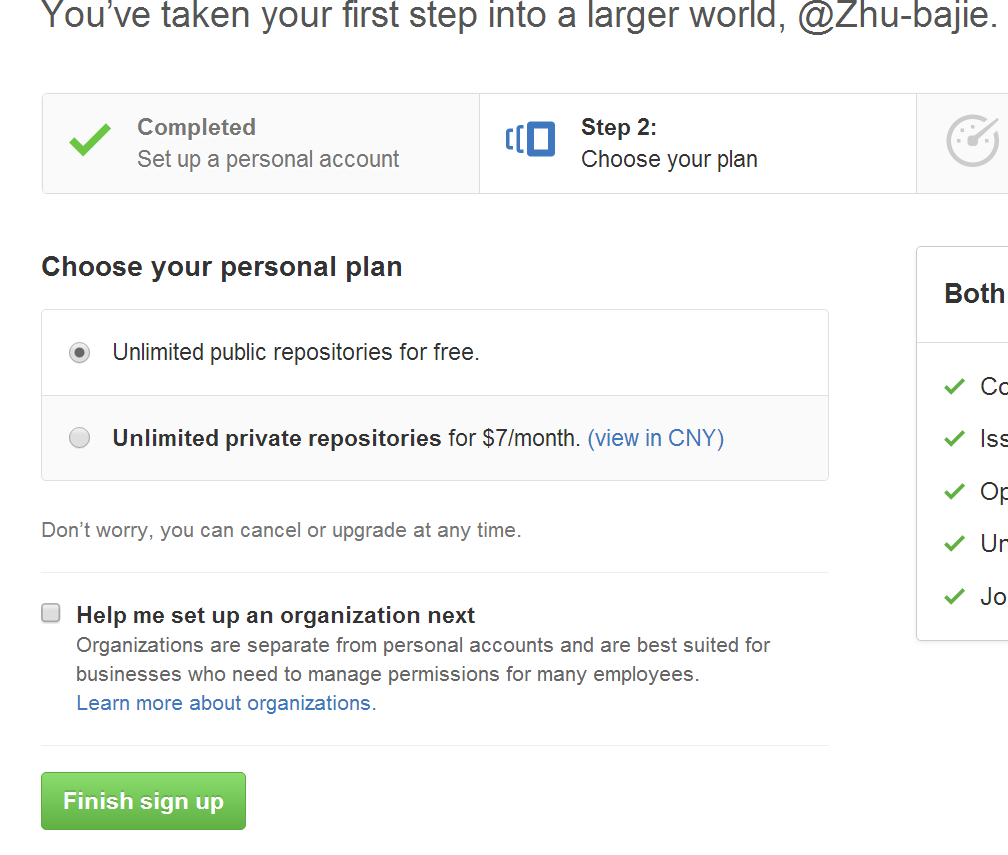
### 密码至少包含一个字母，一个数字和七个字符

你填的邮箱要跟Git的那个邮箱一致

### 点击绿色按钮Create an account，创建账号

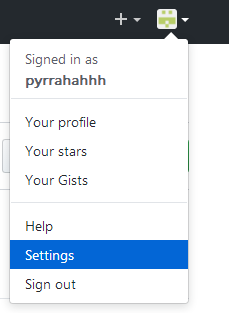
#### 4. 在下一个页面点击绿色按钮Finish sign完成注册

### IMG_256

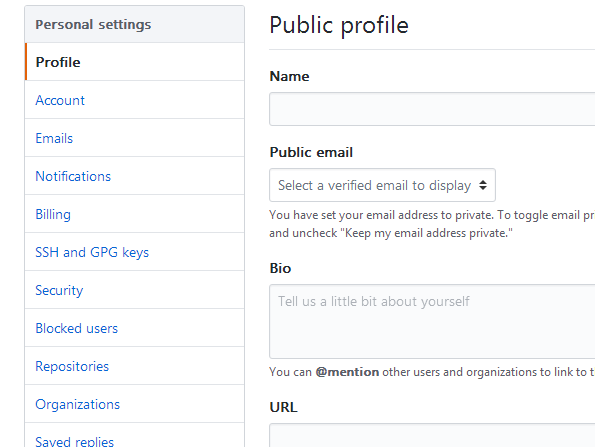


现在，你的GitHub账号就创建好了。请先在你的注册邮箱中完成邮件验证，然后可以看看GitHub提供的[新手文档](https://guides.github.com/activities/hello-world/" \t "http://blog.csdn.net/javaandroid730/article/details/_blank)（嗯，全是英文）

登陆之后按这里

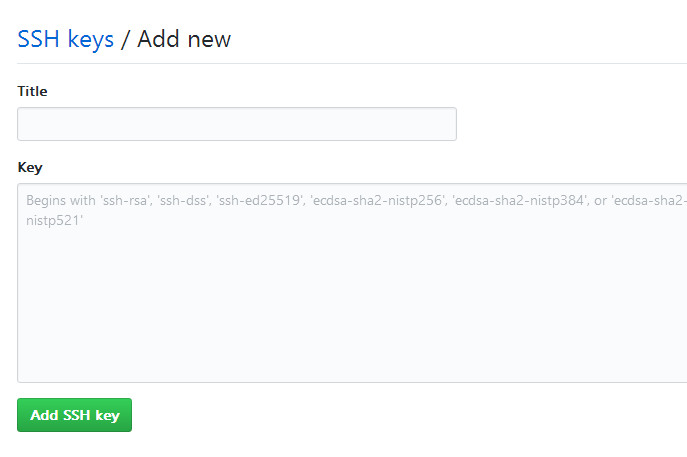


然后按这里，现在要配置一下ssh公钥，用于与本地git建立连接



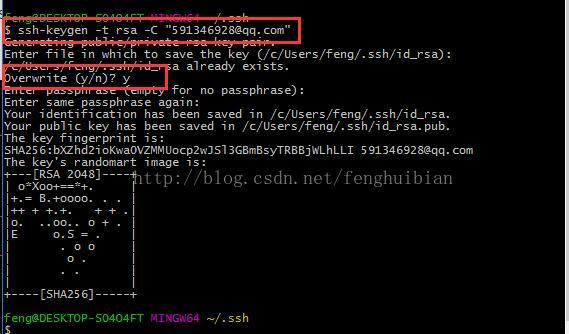
Title里面是可以自己随便起名字的

Key那里面是要粘贴东西的 那个东西我们要去git生成

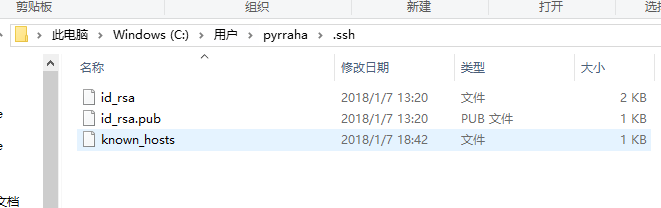


打开git bash，输入ssh-keygen -t rsa -C "email@email.com"命令，其中"email@email.com"是你在GitHub的账号（也就是邮箱）

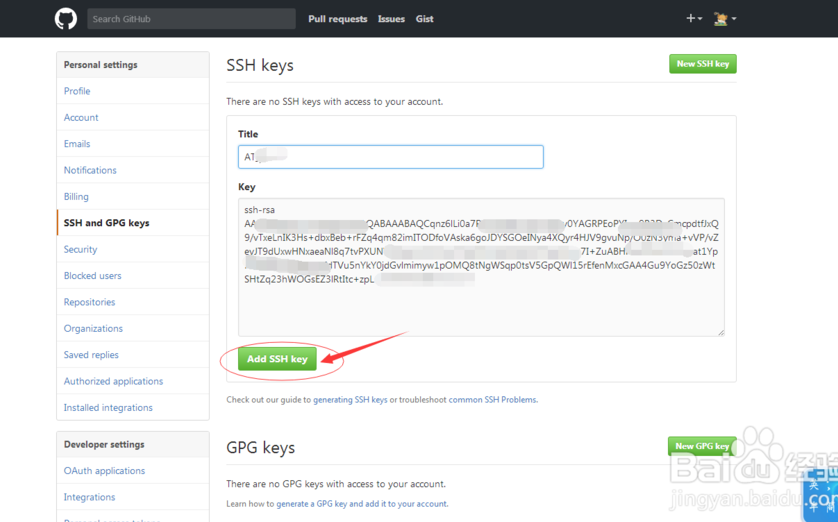
遇到要输（y/n）的地方就输y，其他地方回车就行

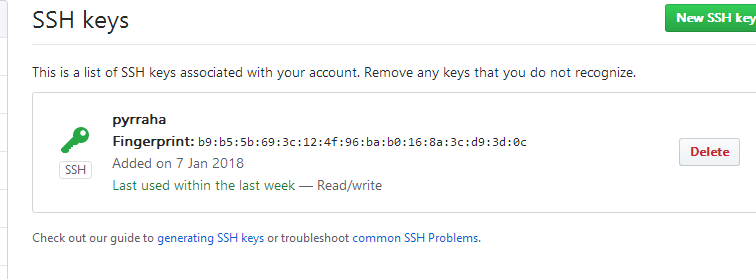


然后打开我的电脑，找到红框内的路径

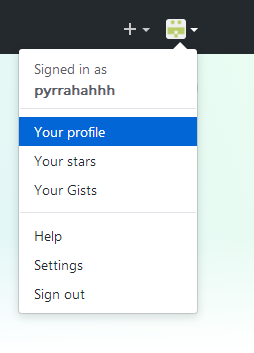


看到有id\_rsa和id\_rsa.pub两个文件，其中id\_rsa是私钥，不能对外公布，id\_rsa.pub是公钥，我们复制公钥（用记事本打开复制）

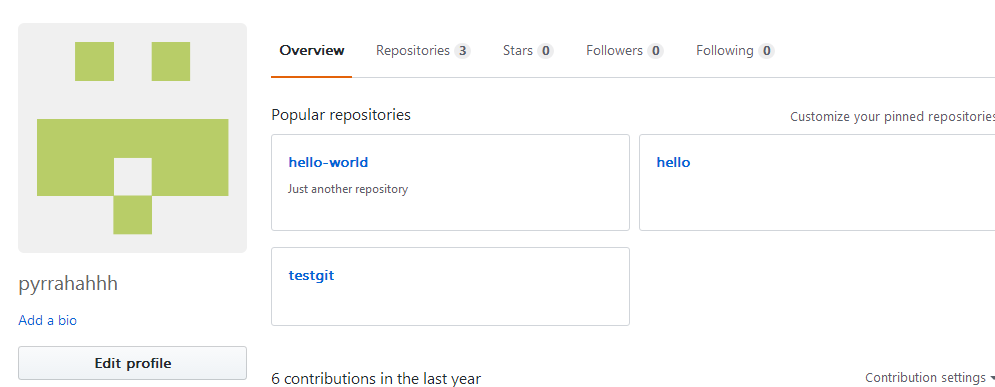


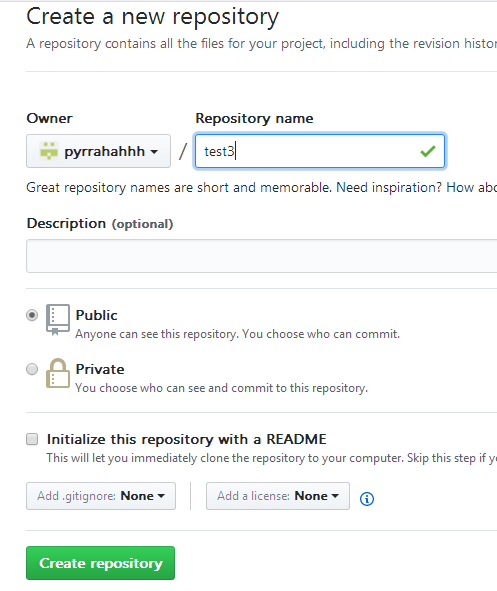


然后回到右上角，按这里



再按下面的地方，new一个远程仓库





复制框住的地方



再回到git bash，输入以下命令

$ git remote add origin git@github.com:pyrrahahhh/test3.git（origin后面就是框里面的东西）



出现上图即为已经建立连接

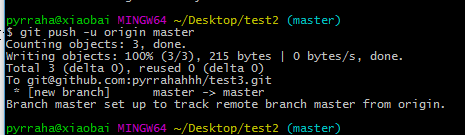
上传到服务器 – git push

把本地的提交传送到服务器的动作叫做push。每次我们要提交修改到服务器上时，都会使用到git push。

git push命令有两个参数，远程仓库的名字，以及分支的名字：

第一次push时 需要输入$ git push -u origin master

之后的push可以直接输入$ git push origin master



如果没有出差错，现在用浏览器看你的远程仓库上已经有hello.txt了。

