

Git and GitHub



June 25, 2021

Jermy Peng

Study record

# Git是什么？GitHub又是什么?

Git是安装在电脑本地的一个版本控制软件，可以对文件不同版本的修改进行保存和追溯。

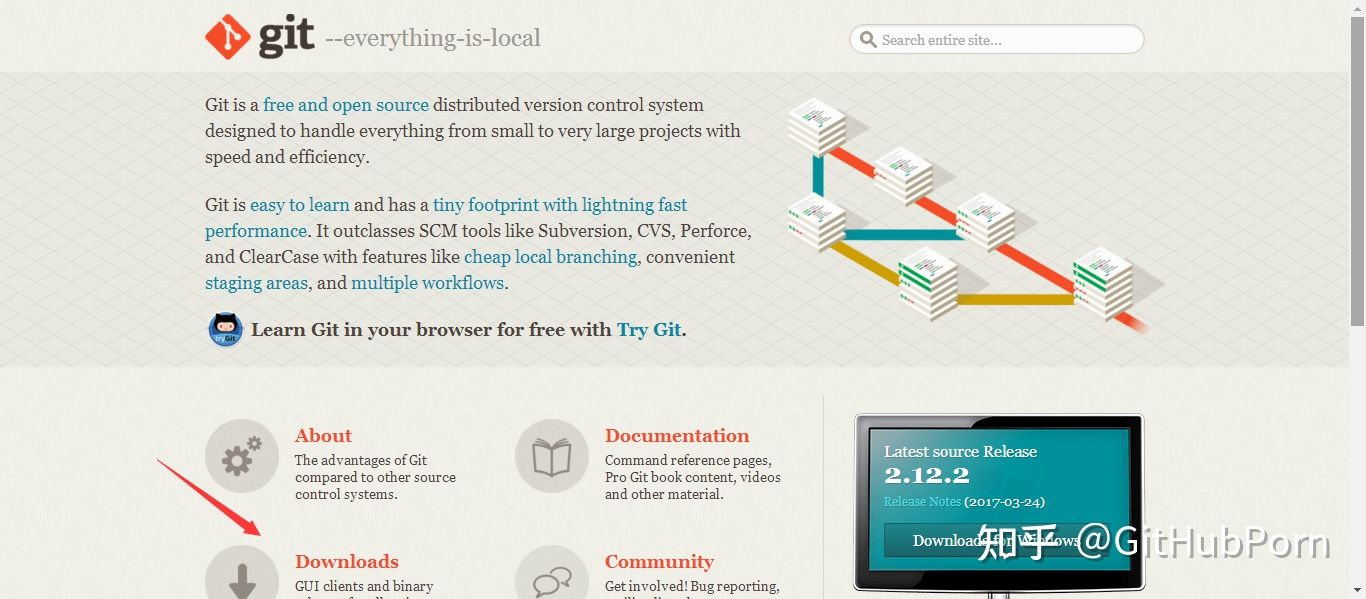
GitHub则是一个开源的版本控制网站，通过Git管理的本地文件，可以上传到GitHub服务器上，并被GitHub上的所有用户看到。GitHub上的用户可以对别人的代码进行复制，修改，提交合并请求等操作；也可以对别人的代码做出评论，请教问题等等。GitHub既是一个开源的代码托管网站，也是一个程序员社交平台。

# Git要如何使用？

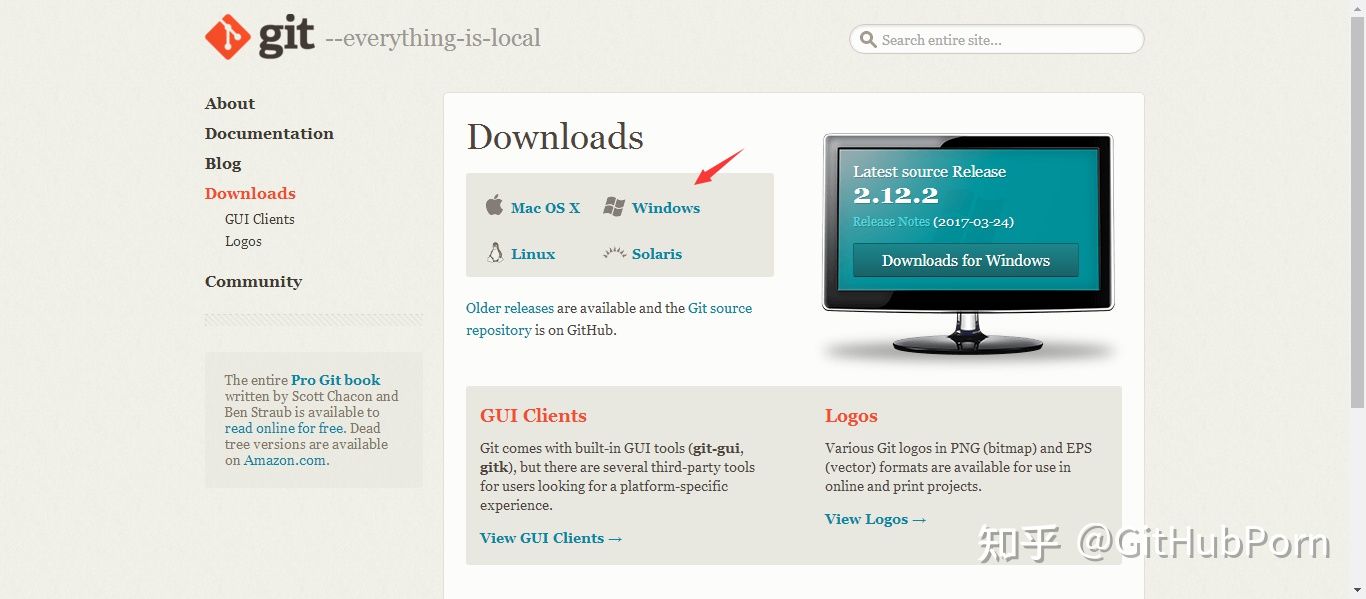
## 下载

首先，安装Git软件。Windows系统上的安装方式，是到Git下载: https://git-scm.com/downloads地址下载软件包，在本地安装即可。具体的安装步骤如下，以下答案引用自知乎答主GitHubPorn。

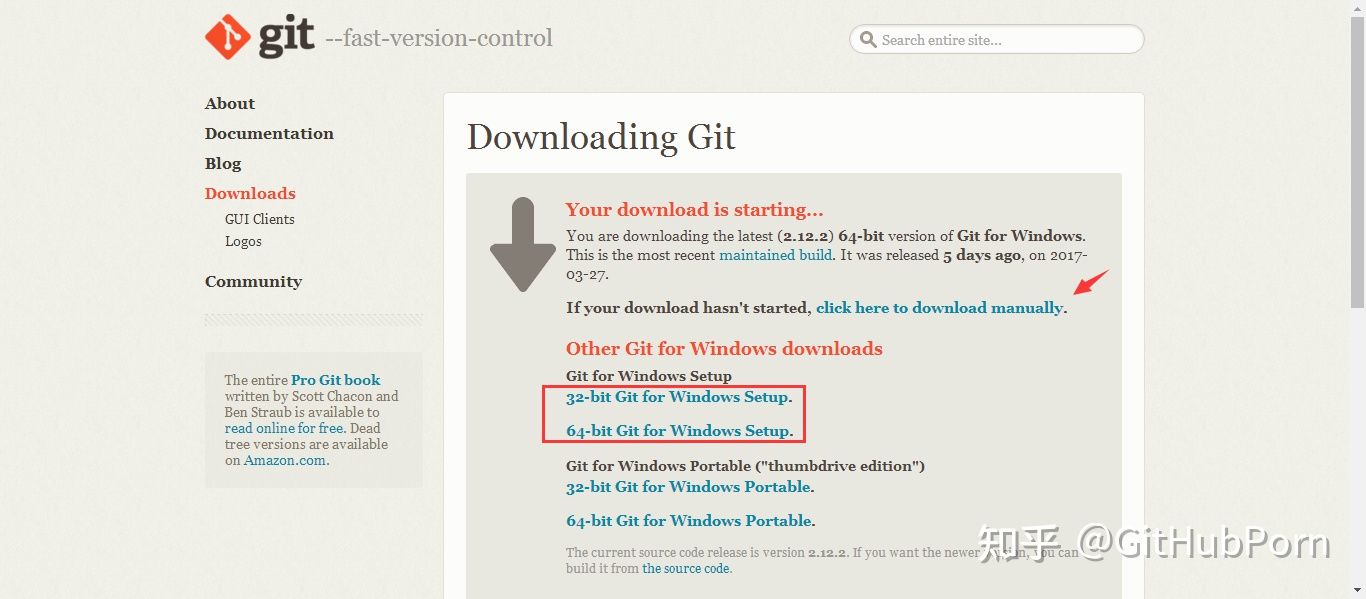
首先，进入 Git 的官网：git - -fast-version-control



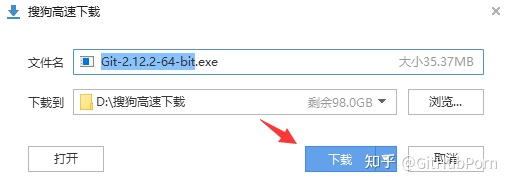
如上图所示，在 Git 的官网中点击Downloads，进入如下页面：



如上图所示，选择对应的操作系统，以博主为例，点击Windows，进入如下页面：



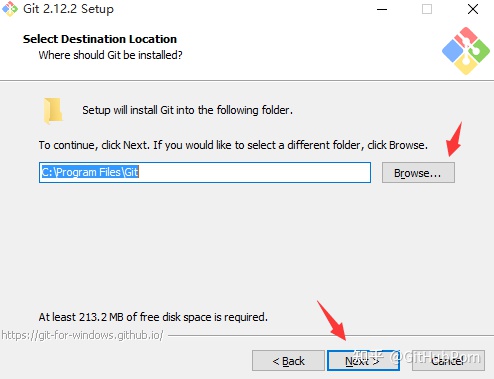
如上图所示，正常情况下，会自动弹出下载框，否则的话，手动点击红色箭头所示的**click here to download manually**亦可进入如下界面：



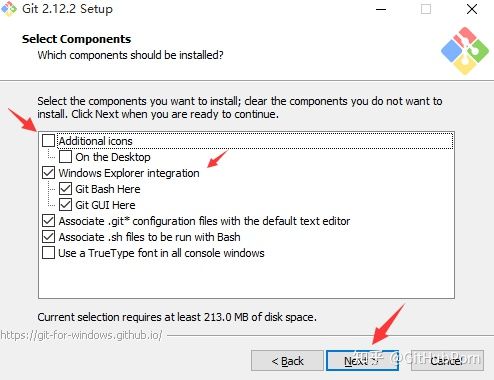
如上图所示，直接点击 下载 即可，下载完成后，双击打开，进入如下界面：



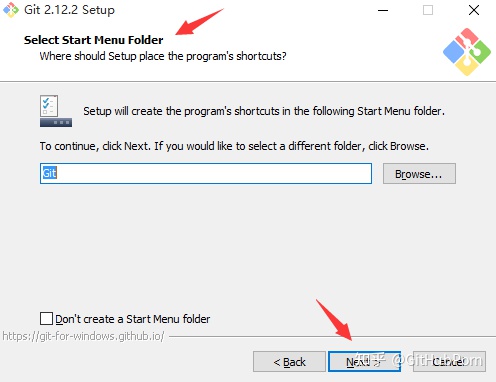
如上图所示，这是 Git 的安装界面，点击Next，进入如下界面：



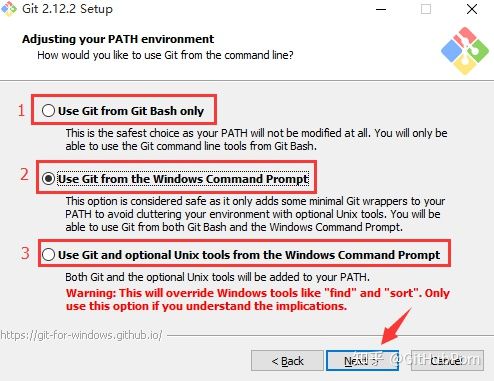
如上图所示，选择 Git 的安装目录，默认安装到C盘的Program Files目录下，想换的话，点击Browse进入更换。在这里，我们选择将其安装到D盘的Program Files目录下，选择完成后，点击Next，进入如下界面：



如上图所示，这里有一些可勾选的项，我们可以按自己的实际需求进行选择（后面同样如此），例如勾选Additional icons，将在 Git 安装完成后，在桌面创建一个图标，也就是打开 Git 的快捷方式。在这一步，建议大家选择默认即可，例如默认勾选的Windows Explorer integration，就可以让我们在点击鼠标右键的时候，快速选择打开Git GUI或者 Git Bash。选择完成后，点击Next，进入如下界面：



如上图所示，选择 开始菜单文件夹，默认即可，点击Next，进入如下界面：

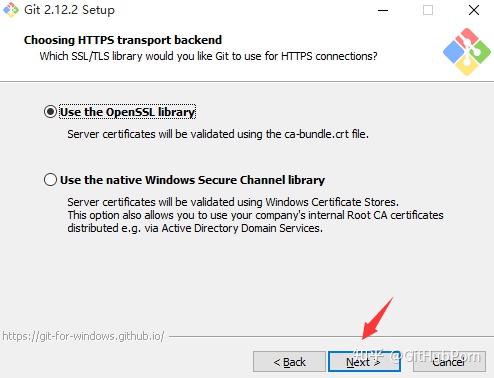


标注 1：仅使用 Git Bash 进行操作；

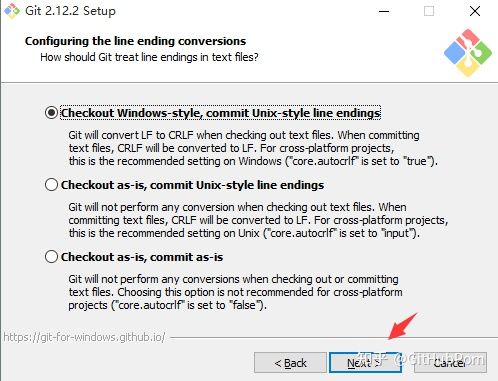
标注 2：在选择使用 Git Bash 进行操作的同时，也可以使用 Windows 命令行操作，建议选择此项；

标注 3：在选择使用 Git 的同时，也把 Unix 工具加入到了我们的配置之中，而且此操作会覆盖 Windows 的一些工具，强烈不建议选择此项。

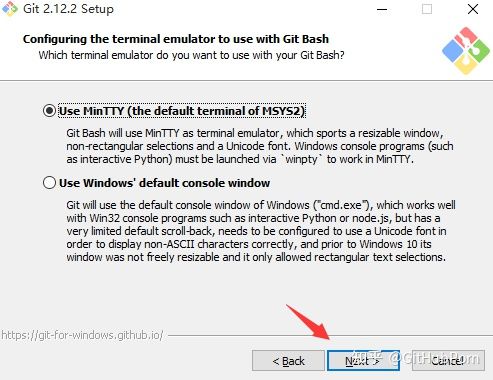
如上图所示，我们选择 标注2 所示的Use Git from the Windows Command Prompt，点击Next，进入如下界面：



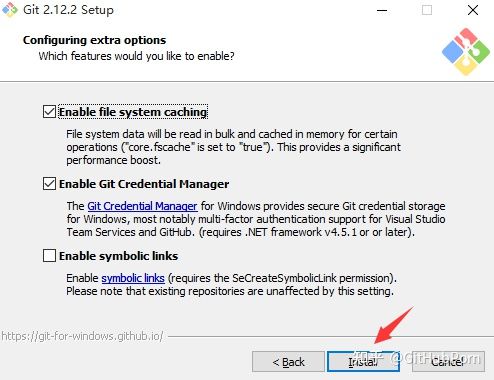
如上图所示，选择 HTTPS 传输后台，默认即可，点击Next，进入如下界面：



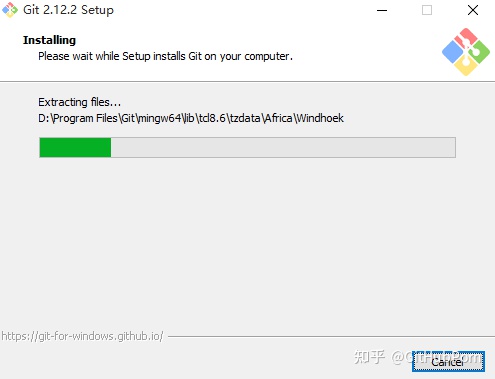
如上图所示，配置行结束标记，默认即可，点击Next，进入如下界面：



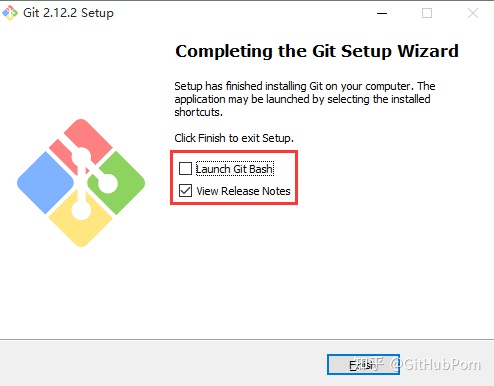
如上图所示，配置 Git Bash 的终端模拟器，默认即可，点击Next，进入如下界面：



如上图所示，配置补充功能，默认即可，点击Next，进入如下界面：



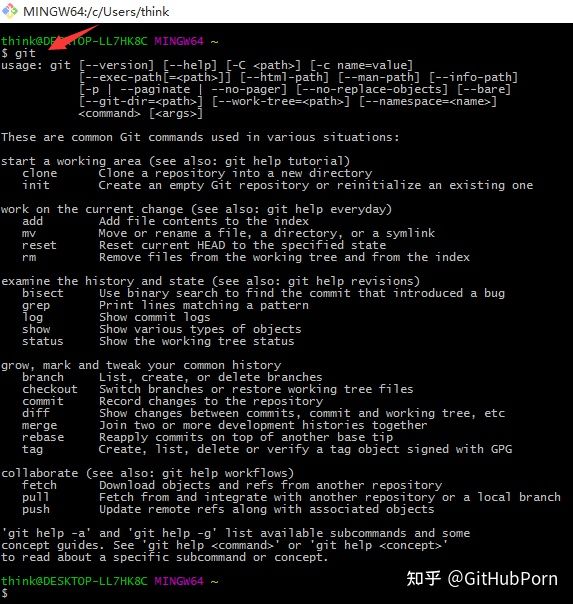
如上图所示，展示了 Git 安装中的界面，安装完成后，弹出如下窗口：



如上图所示，这表示 Git 已经安装完成了，至于图中的两个选择，则分别表示 打开 Git Bash 和 浏览 Git 版本信息，可以都选，也可以都不选，在这里，我们选择Launch Git Bash，进入如下界面：



如上图所示，我们打开了 Git Bash，输入git命令，将显示如下结果：



如上图所示，Git 已经准备就绪啦，接下来就是你的 show time 啦！

## **初始设置**

其次，要对Git做一些初始设置，包括设置我们的GitHub用户名和邮箱地址，以及Git连接到GitHub时需要认证的SSH Key.

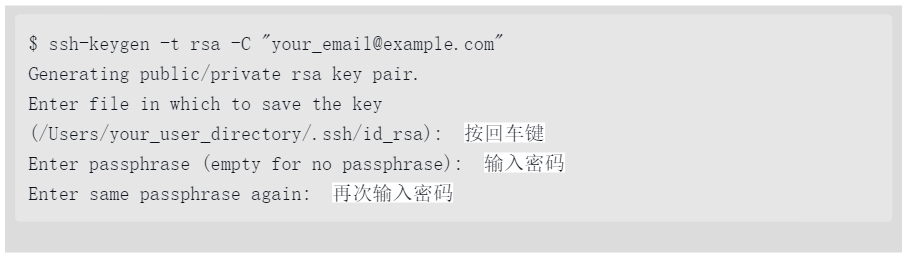
以下是设置用户名和邮箱的git命令。

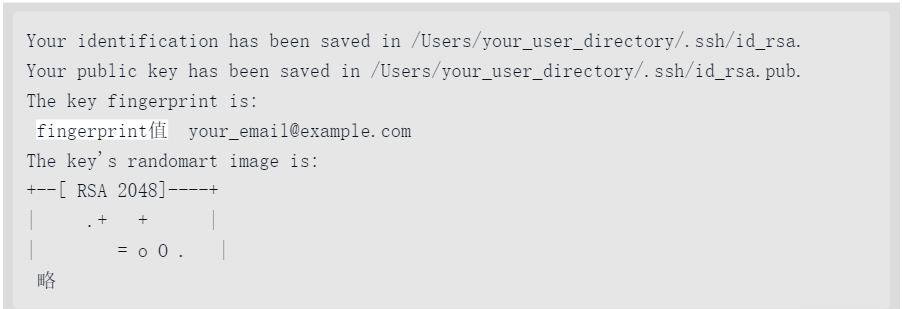


以下是设置SSH key的步骤：

●设置SSH Key

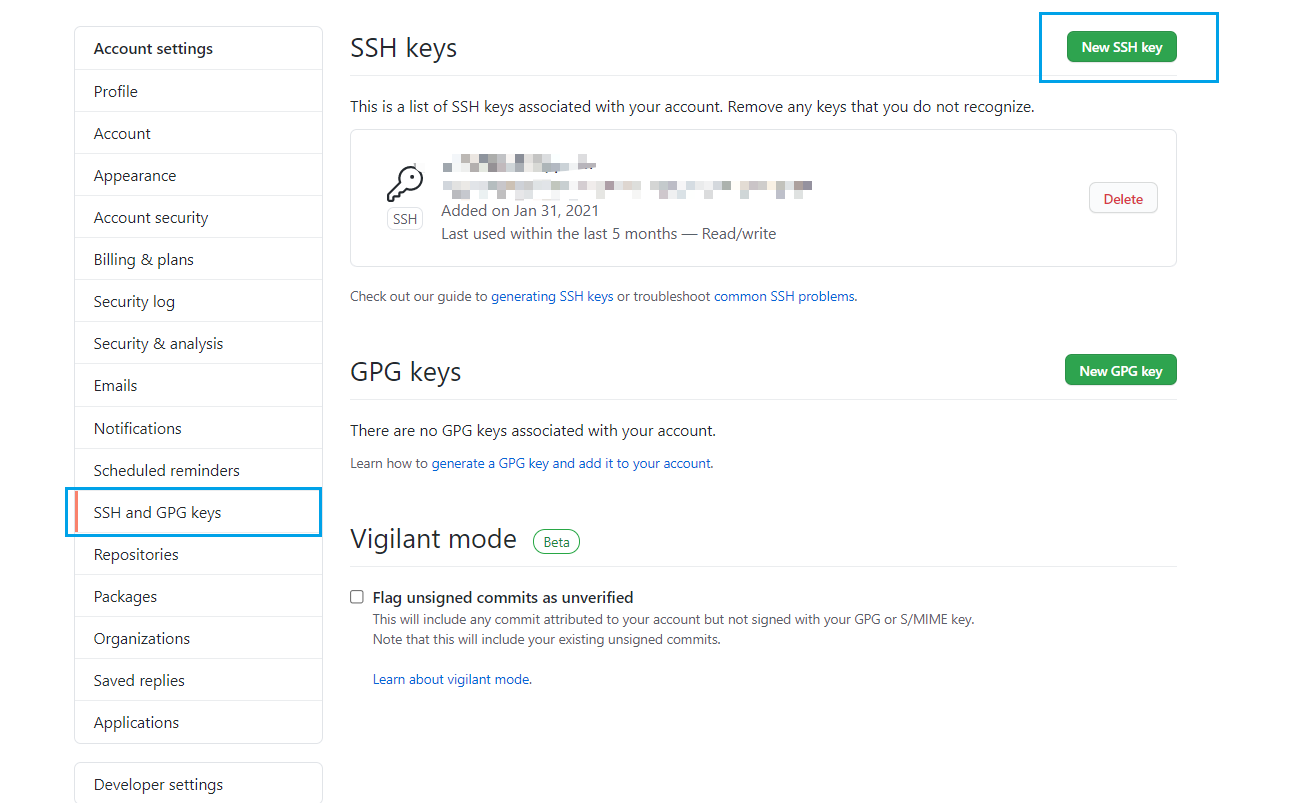
设置SSH KeyGitHub上连接已有仓库时的认证，是通过使用了SSH的公开密钥认证方式进行的。现在让我们来创建公开密钥认证所需的SSH Key，并将其添加至GitHub。运行下面的命令创建SSH Key。

 your\_email@example.com”的部分请改成您在创建账户时用的邮箱地址。密码需要在认证时输入，请选择复杂度高并且容易记忆的组合。输入密码后会出现以下结果。

 id\_rsa文件是私有密钥，id\_rsa.pub是公开密钥。

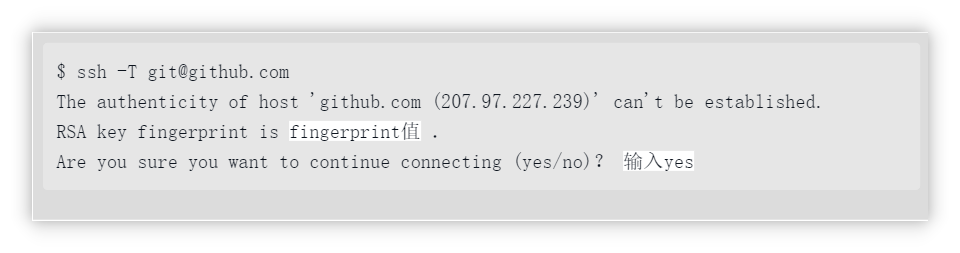
●添加公开密钥

在GitHub中添加公开密钥，今后就可以用私有密钥进行认证了。点击右上角的账户设定按钮（Settings），选择SSH Keys菜单后，就会出现如下图所示的界面。

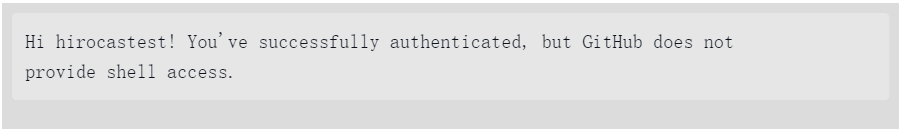
点击Add SSH Key，会出现Title和Key两个输入框。在Title中输入适当的密钥名称。Key部分请粘贴id\_rsa.pub文件里的内容。

id\_rsa.pub的内容可以用如下命令查看。

 公开密钥添加成功之后，创建账户时所用的邮箱会接到一封提示“公共密钥添加完成”的邮件。完成以上设置后，就可以用手中的私人密钥与GitHub进行认证和通信了。

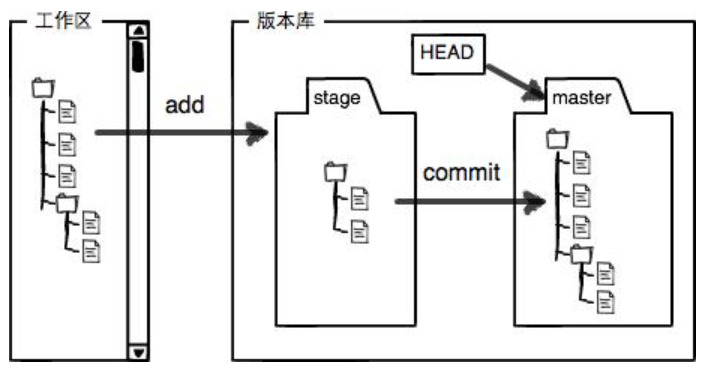


出现如下结果即为成功。



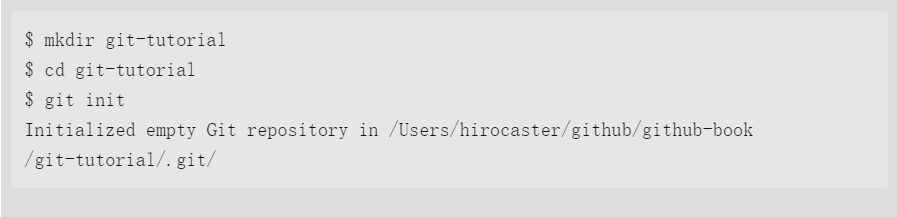
## **Git基本操作指令**

接着，学会使用一些Git指令。学习指令之前，我们先了解几个与Git版本控制相关的名词：工作区(Working directory)，版本库(Repository)，暂存区(stage)，默认的分支(master),指向master的指针(head).它们的关系如下图所示：



工作区就是指我们需要保存文件的文件夹。**在需要进行版本控制的文件目录下，**打开git bash窗口，可以使用以下指令操作：

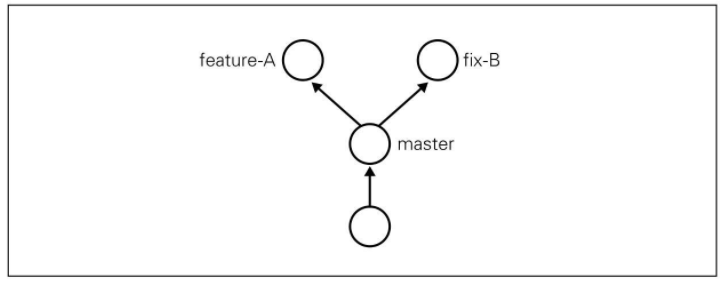
1. git init :初始化仓库，将当前的文件夹初始化出一个仓库，使其中的文件可以进行版本控制，**该文件夹目录下会生成一个.git文件，该文件就是Git的版本库，**并且该文件有可能隐藏。



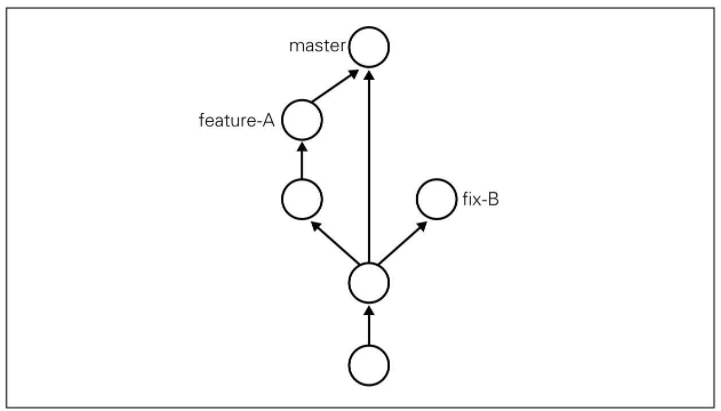
1. git status: 查看文件状态，可以看到该文件夹内的文件都处于什么状态。
2. git add:把文件添加到暂存区，暂存区是正式提交前的临时区域。
3. git commit -m”message”: 提交修改。-m”message”用于添加对这次修改的描述。如果不加-m,直接执行git commit命令，执行后会自动启动编辑器，可以在编辑器中添加更详细的描述，如果不添加任何描述直接关闭编辑器，则提交操作被取消。
4. git log:查看文件修改记录。git log是直接查看所有的变更记录。git log –pretty=short只显示一行简述，“git log 文件名“可查看指定文件的变更记录，git log -p 可显示文件前后的差异，再加上文件名可显示指定文件的前后差异。还可以使用git log –graph，通过图表形式查看修改记录。
5. git diff：该命令可以查看一个文件在工作区，暂存区和master之间的差异，git diff可查看工作区和暂存区的区别，若是暂存区没有内容，则会对比工作区和master中最新版本的区别。通过git diff head可直接查看工作区和最新提交版本的差别。

## Git分支操作指令

分支(branch): 分支是可以实现并行开发的功能。master分支是Git默认生成的第一个分支。其他的分支基本上都是从master分支衍生而来。



在不同分支中，可以同时进行完全不同的作业，等到该分支完成了作业，再合并到master分支，从而实现并行开发。灵活使用分支功能，可以许多人合作完成一个项目，大大提高开发效率。



对分支操作的命令如下：

1. git branch: 显示分支列表，并用\*标识我们当前所在的分支。
2. git checkout -b 分支名：该命令其实相当于两个命令，创建一个分支：git branch 分支名，再切换到该分支：git checkout 分支名。这里直接使用一个命令就实现了创建和切换分支。
3. git merge –no-ff 分支名：先用git checkout切换回主分支，再执行该命令可将其他分支合并到主分支。
4. git reset –hard 哈希值：将文件恢复到哈希值对应时间点的版本，通过git log命令可以查看到不同提交时间的哈希值
5. git commit –amend：修改上次提交的备注信息
6. git commit –am “备注信息”：相当于git add和git commit两个命令
7. git rebase -I ：用于压缩修改的历史记录，常用语错别字的修改

## Git远程仓库操作

1. 先在GitHub上创建一个远程仓库
2. git remote add：将GitHub上新建的仓库设置为本地仓库的远程仓库
3. git push -u 远程仓库标识符 分支：可以将本地仓库的内容推送至远程仓库中的某个分支
4. git clone：将远程仓库克隆到本地
5. git pull：获取远程仓库最新的状态