**一、安装git**

**（1）Windows系统安装**

从Git官网下载安装程序，默认选项安装即可

官网下载地址：

安装完成后，在开始菜单里找到“Git”-->“Git Bash”，蹦出一个类似命令行窗口的东西，说明Git安装成功。

安装完成后，进行最后一步设置，输入：

$ git config --global user.name "Your Name" $ git config --global user.email "email@example.com"

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：你的名字和Email地址

**（2）Linux系统安装**

首先，可以试着输入 git，看看系统有没有安装git：

$ git The program 'git' is currently not installed. You can install it by typing: su**do** apt-**get** install git

像上面的命令，有很多Linux会友好地告诉你Git没有安装，还会告诉你如何安装Git。

如果没有安装，通过一条sudo apt-get install git就可以直接完成Git的安装，非常简单。

**二、创建版本库**

首先找到一个空文件夹

然后，初始化一个Git仓库，使用git init命令。

$ git init

Initialized empty Git repository in E:/learngit/.git/

提示初始化成功。

添加文件到Git仓库，分两步：

1. 使用命令git add <file>，注意，可反复多次使用，添加多个文件；
2. 使用命令git commit -m <message>，完成。

message 里用来描述此次提交的描述信息（一定要如实，准确的描述）

**三、使用Git**

要随时掌握工作区的状态，使用git status命令。

如果git status告诉你有文件被修改过，可以用git diff查看修改内容。（git只能对比二进制文件，对于图片，Word文件对比不了）

**（1）版本回退**

git log命令可以查看文件提交历史记录。

如果先输出信息太多，看的眼花缭乱的，可以加上 --pretty=oneline参数。

即： git log --pretty=oneline

显示效果为：

e67a334cb96565d704ef4625cf7e23c961ffd6ce (HEAD -> master) append GPL

49e772656ccd25957ae2af3bcee9eda8be6c2078 add distributed

1bfe975300924f5b0640bd86dd62a543afb8990f wrote a readme file

（前边为版本号，由SHA1计算出来的十六进制数字）

如果想要回退的话，可以使用 git reset命令

例如： git reset --head HEAD^ #表示回退到上一个版本。

如果想要撤销掉回退操作，那么可以根据版本号操作（只需要前几位版本号即可）

例如：git reset --HEAD e67a

如果忘记了前边操作的版本号了，可以通过git reflog命令查看之前的每一次操作命令。

例如songwanyang@DESKTOP-DG6B04L MINGW64 /e/learngit (master)

$ git reflog

e67a334 (HEAD -> master) HEAD@{0}: reset: moving to e67a

49e7726 HEAD@{1}: reset: moving to HEAD^

e67a334 (HEAD -> master) HEAD@{2}: reset: moving to HEAD

e67a334 (HEAD -> master) HEAD@{3}: commit: append GPL

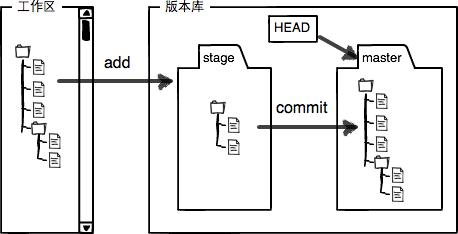
49e7726 HEAD@{4}: commit: add distributed

1bfe975 HEAD@{5}: commit (initial): wrote a readme file

**（2）工作区和暂存区**

工作区就是电脑里能看到的目录，比如我的learngit文件夹就是一个工作区。

工作区有一个隐藏目录.git，这个不算工作区，而是git的版本库。。Git的版本库里存了很多东西，其中最重要的就是成为stage（或者叫index）的缓存区，还有Git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。



前面讲了我们把文件往Git版本库里添加的时候，是分两步执行的：

第一步是用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；

第二步是用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。

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，现在，git commit就是往master分支上提交更改。

Untracked 表示还从来没有被添加过

**（3）管理修改**

Git跟踪并管理的是修改，而不是文件。

每次修改，如果不用git add到暂存区，那就不会加入到commit中。

**（4）撤销修改**

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

git checkout -- file可以丢弃工作区的修改：

例如：

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：

一种是readme.txt自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；

一种是readme.txt已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。

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

现在，看看readme.txt的文件内容：

(注意：git checkout -- file命令中的--很重要，没有--，就变成了“切换到另一个分支”的命令，我们在后面的分支管理中会再次遇到git checkout命令。)

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD <file>，就回到了场景1，第二步按场景1操作。

例如：

git reset HEAD readme.txt

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。当我们用HEAD时，表示最新的版本。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考[版本回退](https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013744142037508cf42e51debf49668810645e02887691000)一节，不过前提是没有推送到远程库。

**（5）删除文件**

一般情况下，你通常直接在文件管理器中把没用的文件删了，或者用rm命令删了：

$ rm test.txt

这个时候，Git知道你删除了文件，因此，工作区和版本库就不一致了，git status命令会立刻告诉你哪些文件被删除了：

现在你有两个选择，一是确实要从版本库中删除该文件，那就用命令git rm删掉，并且git commit：

另一种情况是删错了，因为版本库里还有呢，所以可以很轻松地把误删的文件恢复到最新版本：

$ git checkout -- test.txt

※※※

命令 git rm用于删除一个文件。如果一个文件已经被提交到版本库，那么你永远不用担心误删，但是要小心，智能恢复文件到最新版本，会丢失最近一次提交后修改的内容。

※※※

四、应用远程仓库

※※※

git是分布式版本控制系统，同一个Git仓库，可以分布到不同的机器上-->最早只有一台机器有一个原始版本库，此后，别的机器可以“克隆”这个原始版本库，而且每台机器的版本库都是一样的，并没有主次之分。

※※※

**1、创建远程库，以GitHub为例**

第一步，创建SSH Key。在Git Bash窗口下输入：

$ ssh-keygen -t rsa -C "youremail@example.com"

需要把邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可，由于这个Key也不是用于军事目的，所以也无需设置密码。

如果一切顺利的话，可以在用户主目录里找到.ssh目录，里面有id\_rsa和id\_rsa.pub两个文件，这两个就是SSH Key的秘钥对，id\_rsa是私钥，不能泄露出去，id\_rsa.pub是公钥，可以放心地告诉任何人。

第二步，登录GitHub，在Setting-->SSH and GPG keys处，创建New SSH Key



点“Add Key”，你就应该看到已经添加的Key：



※※※

（1）为什么GitHub需要SSH Key呢？

因为GitHub需要识别出你推送的提交确实是你推送的，而不是别人冒充的，Git支持SSH协议，所以，GitHub只要知道了你的公钥，就可以确认只有你自己才能推送了。

当然，GitHub允许添加多个Key。只要把每台电脑的Key都添加到GitHub，就可以在每台电脑上往GitHub推送了。

（2）GitHub上免费托管的Git仓库，任何人都可以看到（但只有自己才能修改）。所以不要把敏感信息放进去。

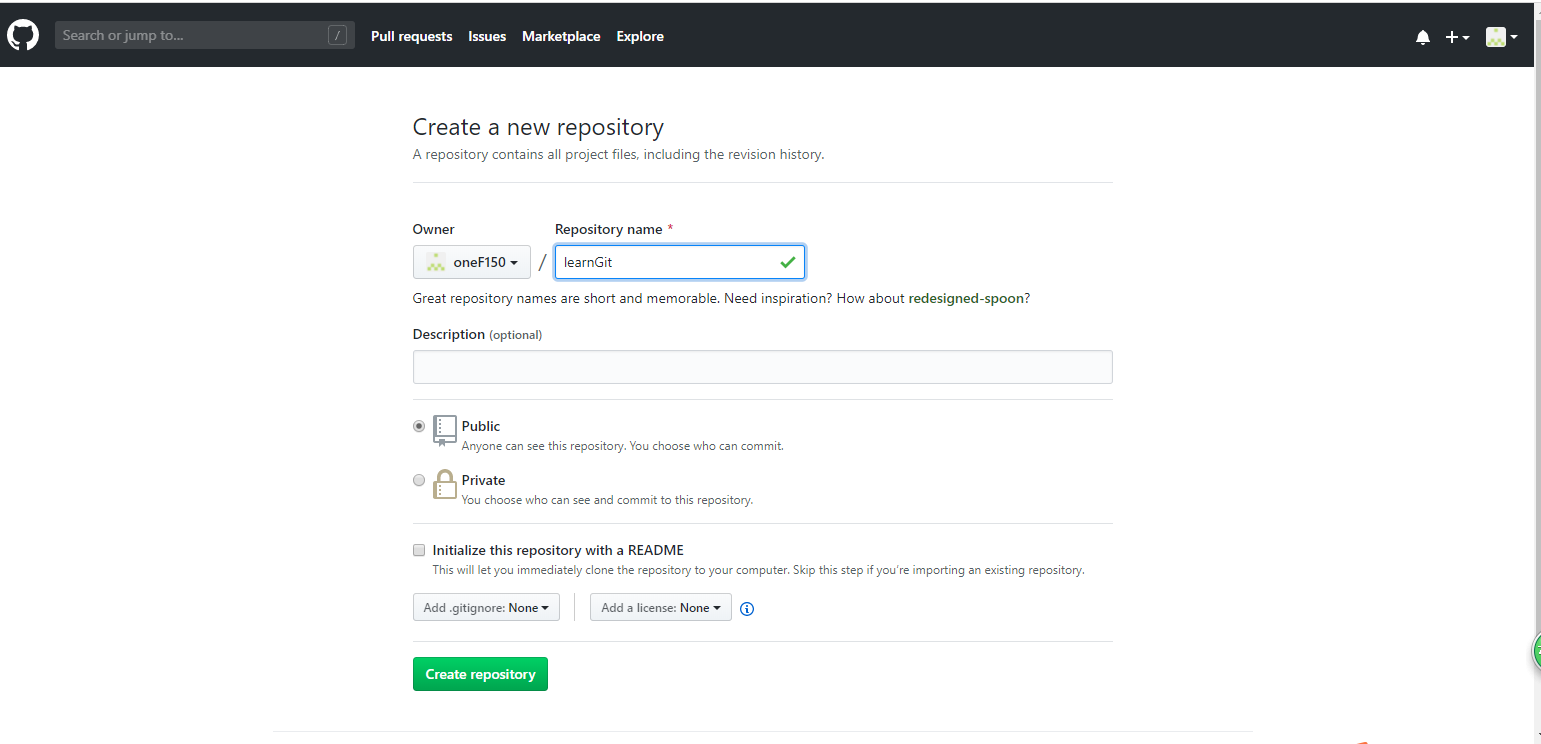
如果不想让别人看到Git库。两个办法，一个是交点保护费，让GitHub把公开的库变成私有的；另一个办法是自己动手，搭建一个Git服务器，因为是自己的Git服务器，所以别人是看不见的。

※※※

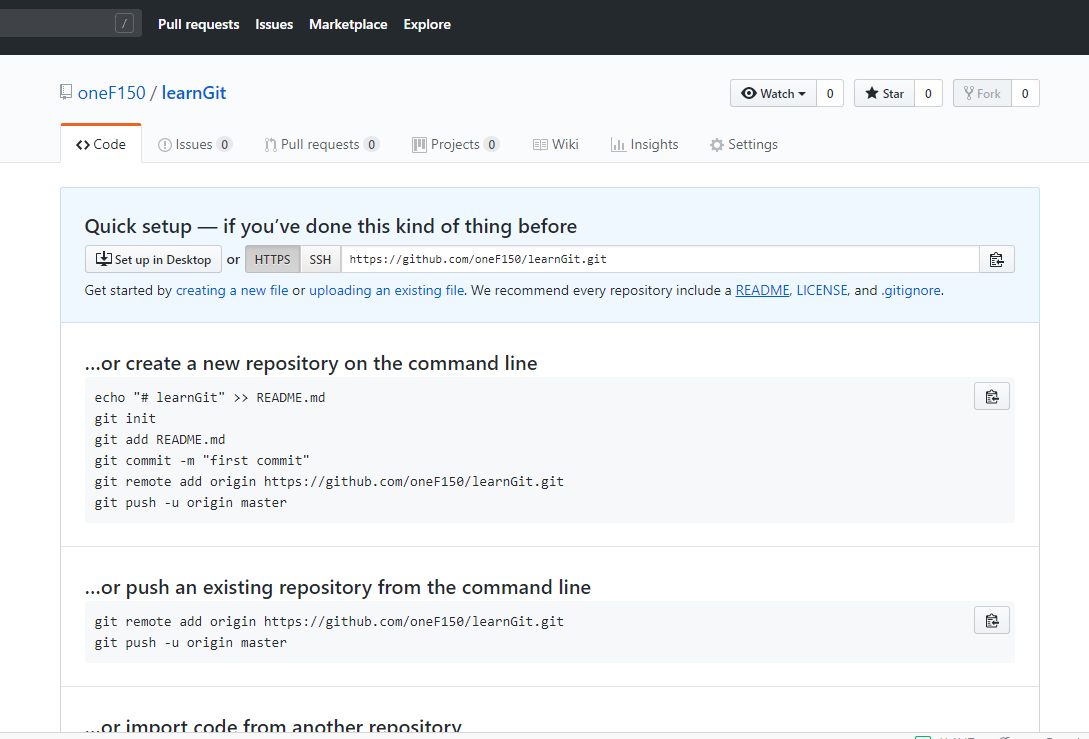
2、添加远程库

本地有一个Git仓库了，又想在GitHub上创建一个Git仓库，并且让这两个仓库进行远程同步。这样，GitHub上的仓库既可以作为备份，又可以让别人通过该仓库来协作。操作步骤为：

（1）登录GitHub，在右上角“+”-->"New repository"按钮，创建一个新的仓库：



在Repository name填入learngit，其他保持默认设置，点击“Create repository”按钮，就成功地创建了一个新的Git仓库：



目前，在GitHub上的这个learnGit仓库还是空的，GitHub告诉我们，可以从这个仓库克隆出新的仓库，也可以把一个已有的本地仓库与之关联，然后，把本地仓库的内容推送到GitHub仓库。

（2）现在，我们根据GitHub的提示，在本地的learnGit仓库下运行以下命令：

参考：<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013758392816224cafd33c44b4451887cc941e6716805c000>