**git介绍**

**一、Git概述**

**1.1 Git历史**

Git 诞生于一个极富纷争大举创新的年代。Linux 内核开源项目有着为数众多的参与者。 绝大多数的 Linux 内核维护工作都花在了提交补丁和保存归档的繁琐事务上（1991－2002年间）。到 2002 年，整个项目组开始启用一个专有的分布式版本控制系统 BitKeeper 来管理和维护代码。

到了 2005 年，开发 BitKeeper 的商业公司同 Linux 内核开源社区的合作关系结束，他们收回了 Linux 内核社区免费使用 BitKeeper 的权力。 这就迫使 Linux 开源社区（特别是 Linux 的缔造者 Linus Torvalds）基于使用 BitKeeper 时的经验教训，开发出自己的版本系统。

他们对新的系统制订了若干目标：

速度

简单的设计

对非线性开发模式的强力支持（允许成千上万个并行开发的分支）

完全分布式

有能力高效管理类似 Linux 内核一样的超大规模项目（速度和数据量）

**1.2 Git与SVN对比**

SVN是集中式版本控制系统，版本库是集中放在中央服务器的，而开发人员工作的时候，用的都是自己的电脑，所以首先要从中央服务器下载最新的版本，然后开发，开发完后，需要把自己开发的代码提交到中央服务器。

集中式版本控制工具缺点：

服务器单点故障

容错性差

Git是分布式版本控制系统（Distributed Version Control System，简称 DVCS） ，分为两种类型的仓库：

本地仓库和远程仓库。

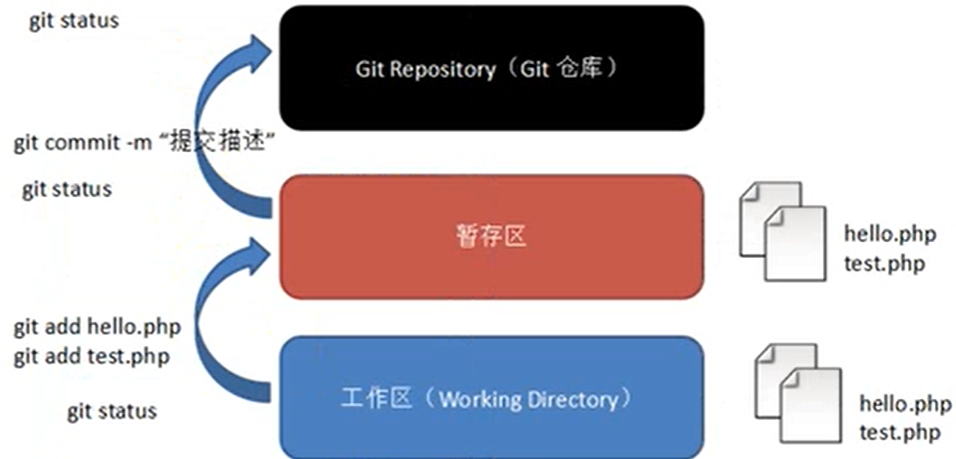
本地仓库：是在开发人员自己电脑上的Git仓库

远程仓库：是在远程服务器上的Git仓库

Clone：克隆，就是将远程仓库复制到本地

Push：推送，就是将本地仓库代码上传到远程仓库

Pull：拉取，就是将远程仓库代码下载到本地仓库



**1.3 Git工作流程**

工作流程如下：

1．从远程仓库中克隆代码到本地仓库

2．从本地仓库中checkout代码然后进行代码修改

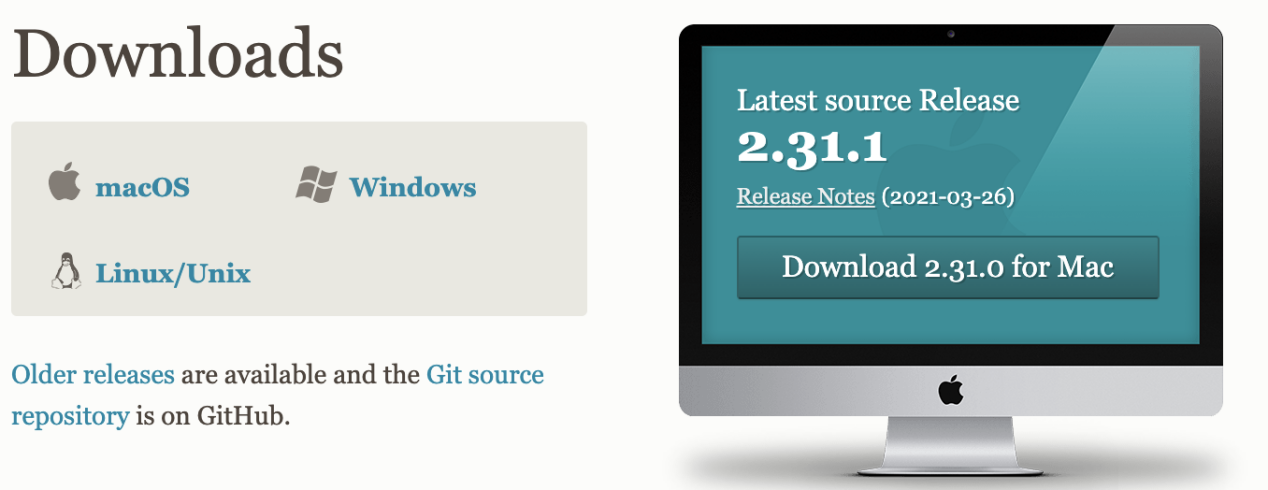
3．在提交前先将代码提交到暂存区

4．提交到本地仓库。本地仓库中保存修改的各个历史版本

5．修改完成后，需要和团队成员共享代码时，将代码push到远程仓库

**1.4 Git下载与安装**

下载地址： <https://git-scm.com/download>



双击下载的安装文件来安装Git。  
一直下一步直到安装完成即可

安装完成后在电脑桌面（也可以是其他目录）点击右键，如果能够看到如下两个菜单则说明Git安装成功。



**二、Git代码托管服务**

**2.1 常用的Git代码托管服务**

前面我们已经知道了Git中存在两种类型的仓库，即本地仓库和远程仓库。那么我们如何搭建Git远程仓库呢？我们可以借助互联网上提供的一些代码托管服务来实现，其中比较常用的有GitHub、码云、GitLab等。

gitHub（ 地址：https://github.com/ ）是一个面向开源及私有软件项目的托管平台，因为只支持Git 作为唯一的版本库格式进行托管，故名gitHub

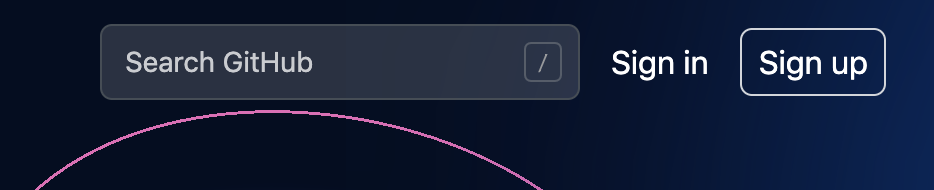
码云（地址： https://gitee.com/ ）是国内的一个代码托管平台，由于服务器在国内，所以相比于GitHub，码云速度会更快

GitLab （地址： https://about.gitlab.com/ ）是一个用于仓库管理系统的开源项目，使用Git作为代码管理工具，并在此基础上搭建起来的web服务

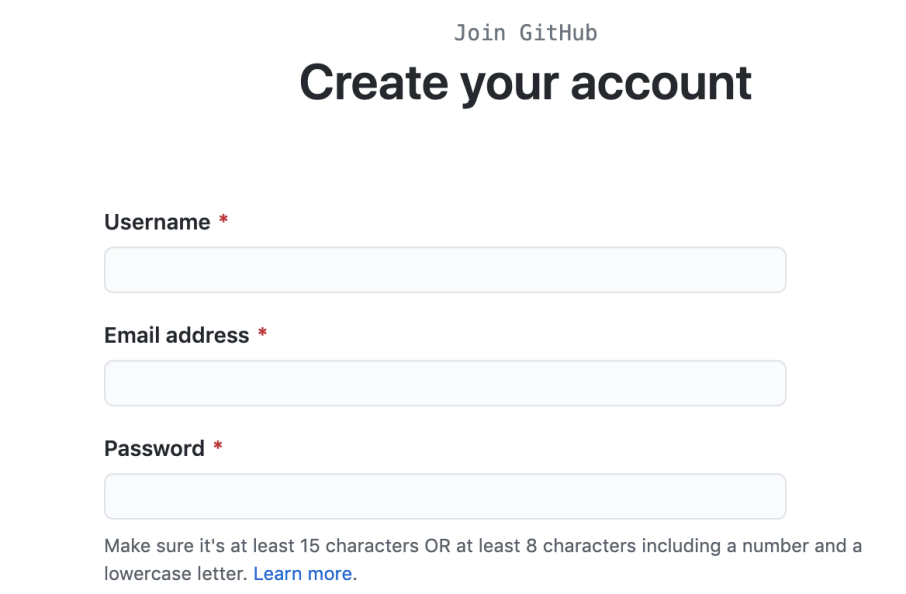
**2.2使用github作为演示**

进入github首页（https://github.com/ ）

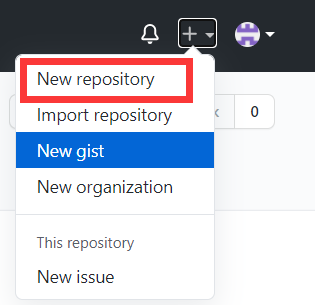
有账号直接登录，没有注册一个



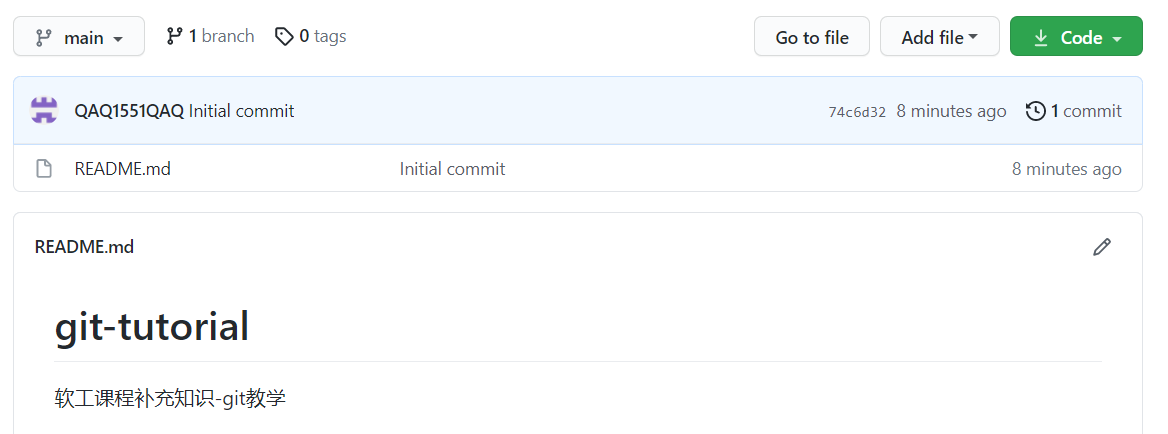
点击sign up，注册



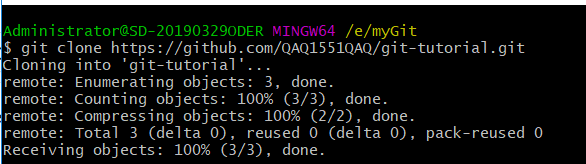
登录之后，创建仓库：



创建后的示例：



拉取项目代码到本地：



如下图所示：



添加user.email和user.name



到这一步，就可以通过git push命令将本地修改后的代码更新到github上了。

常用命令：

git init 新建一个空的仓库  
**git status 查看状态  
git add . 添加文件  
git commit -m '注释' 提交添加的文件并备注说明**git remote add origin git@github.com:jinzhaogit/git.git 连接远程仓库  
**git push -u origin master 将本地仓库文件推送到远程仓库**git log 查看变更日志  
git reset --hard 版本号前六位 回归到指定版本  
**git branch 查看分支**git branch newname 创建一个叫newname的分支  
git checkout newname 切换到叫newname的分支上  
git merge newname 把newname分支合并到当前分支上  
**git pull origin master 将master分支上的内容拉到本地上**