

# GitHub 版本控制技术

授课人：姜自婷（15887066015@163.com）

授课班级：16 级软件技术 1 班

## 课程介绍

1、教材：无

推荐教材：《GitHub 入门与实践》大塚弘记著

网络学习：

官网 <https://github.com>

廖雪峰 Git 教程

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

视频课程 <http://www.imooc.com/learn/390>

2、课程成绩：平时成绩（50%）+期末考核（50%）



## 目录

课程介绍.....	1
学习这门课程的目的.....	2
GitHub 是什么.....	3
使用 GitHub 的前期准备.....	4
GitHub 注册.....	4
Git 客户端 msysgit 下载安装 .....	5
登录 Github 网站同时登陆到 Github for windows.....	6

## 学习这门课程的目的

### 解决这些问题

- 1、代码审查不到位，审查效率低下；

- 2、只有编程者本人能看懂的代码、可靠性不高的代码直接被部署至正式环境中；
- 3、因键入错误、理解错误而造成的低级代码错误导致 BUG 频繁出现；
- 4、**没有机会和其他人相互交流代码，共享知识，相互学习、指正、改善；**
- 5、没有一个简单高效、能在一天之内添加多个功能的开发流程。

## GitHub 是什么

1、从 Git 说起：①Git 的中文意思是“傻子”。②Git 也可以是 global information tracker（全局信息跟踪器）的意思。③**本质：Git 是版本控制工具**，是由 Linux 之父 Linus Torvalds 为了更好地管理 linux 内核开发而创立的**分散型版本管理系统**；



2、什么是版本管理：**管理更新的历史记录**。它为我们提供了一些软件开发过程中必不可少的功能，例如记录一款软件添加或更改软件的过程，回滚到特定阶段，恢复误删除的文件等。所以简单地说，**Git 是一个管理你的[代码的历史记录]的工具**。

3、分散型与集中型：

**集中型**：集中型将所有数据集中存放在服务器中，便于管理。但是一旦服务器宕机，就会无法获取最新的源代码。

**分散型**：GitHub 将仓库 fork 给每一个用户。Fork 就是将 GitHub 的某个特定仓库复制到自己的账户下。分散型拥有多个仓库。

4、**GitHub**：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。是一个让开发者与朋友、同事、同学及陌生人共享代码的完美场所。

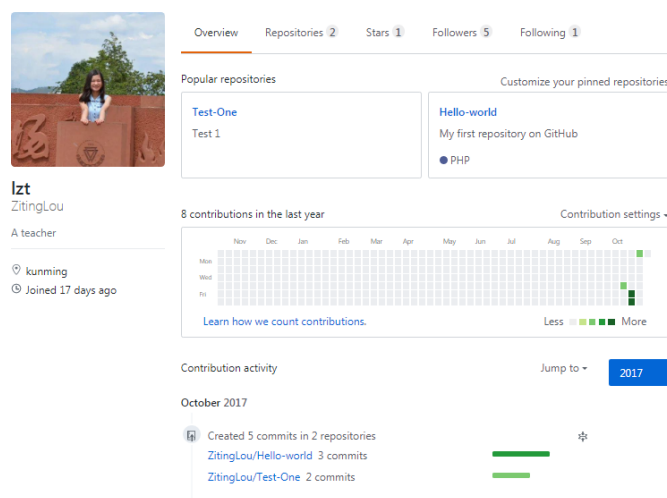
GitHub 于 2008 年 4 月 10 日正式上线，除了 Git 代码仓库托管及基本的 Web 管理界面以

外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。目前，其注册用户已经超过百万，托管版本数量也是非常之多，其中不乏知名开源项目 Ruby on Rails、jQuery 等。

5、Github 主要有下面两个功能：

- 开源社交平台
- 项目管理平台

开源社交平台：Github 从一开始就把开发者的身份、他的关注、他的代码贡献、讨论和活动时间线作为核心功能来做。每个开发者都有一个属于自己的 profile 页面。



项目管理平台：Github 在各种开发者大会上宣扬自己的开源社区，也是项目管理平台，它本身便适合开发 Github 这样的大型项目。

## 使用 GitHub 的前期准备

- 1、注册 Github 账号
- 2、下载 Github for windows
- 3、登录 Github 网站同时登陆到 Github for windows

## GitHub 注册

- 1、打开 GitHub 官网 <https://github.com>。
- 2、点击绿色按钮 , 填写用户名 Username, 邮件地址 Email Address

和密码 Password。

- 3、点击绿色按钮 **Create an account**，创建账号。

**Create your personal account**

There were problems creating your account.

**Username**

Zhu-bajie ✓

**Email Address**

pigheaven@gmail.com ✓

**Password**

\*\*\*\*\* ✓

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

**Create an account**

- 4、在下一个页面点击绿色按钮 **Finish sign up** 完成注册。

You've taken your first step into a larger world, @Zhu-bajie.

✓ Completed Set up a personal account	📁 Step 2: Choose your plan	🔄
--	-------------------------------	---

**Choose your personal plan**

☒ Unlimited public repositories for free.

☐ Unlimited private repositories for \$7/month. ([view in CNY](#))

Don't worry, you can cancel or upgrade at any time.

☐ **Help me set up an organization next**  
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.  
[Learn more about organizations.](#)

**Both**

- ✓ Cc
- ✓ Iss
- ✓ Op
- ✓ Ur
- ✓ Jo

**Finish sign up**

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。

## Git 客户端 msysgit 下载安装

- 1、下载 Git for Windows(下载对应的 exe 文件，注意你的操作系统是 32 位还是 64 位)

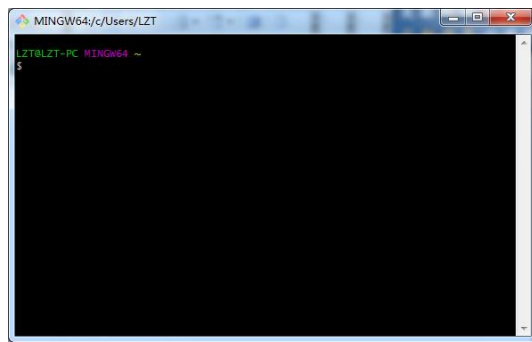
<https://git-scm.com>（官网下载装，有可能会失败）

<https://github.com/waylau/git-for-win>（离线包下载）

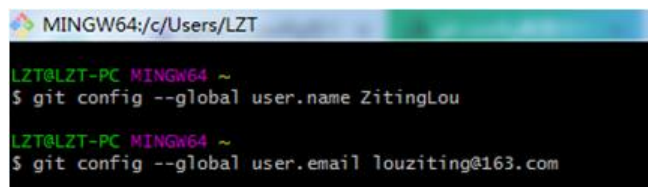
2、双击安装，中间不用做任何改动，一直下一步就行。如果你想修改安装位置，请放在纯英文路径下。

3、安装成功，就可以使用 **git 命令行工具**进行初始化设置了。

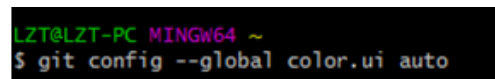
Git Bash 使用的是 MinGW（一个可自由使用和自由发布的 Windows 特定头文件和使用 GNU 工具集导入库的集合），其界面如下图所示：



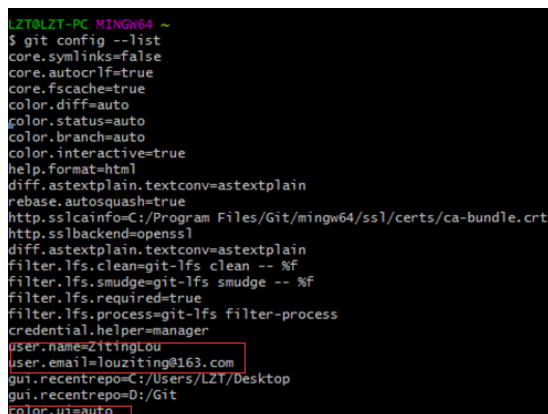
(1) 设置使用 Git 时的姓名和邮箱地址。



(2) 提高命令输出的可读性，将 color.ui 设置为 auto。



(3) 使用 git config --list 检查你的设置。



## 登录 Github 网站同时登陆到 Github for windows

当你通过 Git 连接到一个 GitHub 仓库时，需要验证 GitHub。

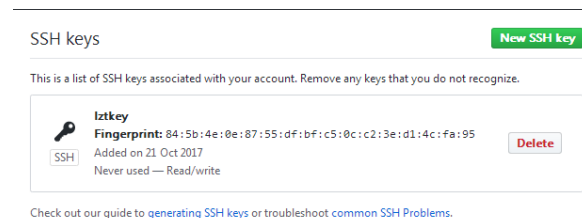
(1) 设置 SSH Key: 运行下面的命令创建 SSH Key，红色框中是输入密码后的 fingerprint

值，id\_rsa 文件是私有密钥，id\_rsa.pub 是公开密钥。

```
LZT@LZT-PC MINGW64 ~
$ ssh-keygen -t rsa -C louziting@163.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c:/Users/LZT/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/LZT/.ssh/id_rsa.
Your public key has been saved in /c:/Users/LZT/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:UHcsz1a3MUSH7UUpiny9F60YwR1VNNV7KAq514RUmMo louziting@163.com
The key's randomart image is:
+---[RSA 2048]---+
|  .  o+=@ |
|  .  .+=00 |
|  ...+..+00+ |
|  .Eo=+00+0 |
|  S+. +0.0 |
|  . 0 0... |
|  .  .  |
+---[SHA256]-----+
```

(2) 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。

```
LZT@LZT-PC MINGW64 ~
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDJnwqkSamEdtduxG/7Ccop8q86dM5Xtap0CEyrfKTV
JKG+jnSIu00JhZaRSMjHNB10yF49zs1w0rD19TREq2QjK32hya6B80+yhVEdwYLabeTv158FoviUz+Ze+
BdKSR7yUyCXOPVo+rZJyt3hX9m/1K03jFott9nvofHecf9yKZUHkLcXvM2qpMpxYtPz1ufZW4m/kkt+4
SVvi1cRzGDITzyFphHqxtEs4zjj3mx8LY1yTv2GuAGes4b1A0MwKYsPzy331Z4uY1zyKTK6ZSIwfKc1Z
NjLBawk7ehrd6j0QDoixEu0qhuRw/tH/bx6snj3Zr+kI9ds1G1UN9I+E6gc1 louziting@163.com
```



(2) 添加成功之后，创建账户时所用的邮箱会接收到一封提示“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。让我们来实际试一试。

```
LZT@LZT-PC MINGW64 ~
$ ssh -T git@github.com
The authenticity of host 'github.com (192.30.255.112)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWg17E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.255.112' (RSA) to the list of known hosts.
Enter passphrase for key '/c:/Users/LZT/.ssh/id_rsa':
Hi ZitingLou! You've successfully authenticated, but GitHub does not provide shell access.
```