

Git基础入门（适用0基础新手版）

本篇笔记适用于刚开始接触git以及不知道git是什么的同学，由于笔者也在学习过程中，可能有些理解有误的地方还请多多包涵和指正！（更新中）

Git基础入门（适用0基础新手版）

- 一、Git是什么（前言）
 - 版本控制
- 二、常用的Linux命令
- 三、Git必要的配置
- 四、Git的工作原理
- 五、Git项目搭建（基本命令操作）
 - 创建仓库
 - Git文件操作
 - 注册gitee/github账号

一、Git是什么（前言）

许多同学在首次听到git的时候都会将其与github弄混，或者以为它们是同一个东西。

Git是一个**分布式版本控制系统**，而**GitHub**是一个基于Git的**代码托管平台**，也就是一个远程的git仓库。你可以简单理解成GitHub是使用Git这个版本控制工具的服务平台。

那么版本控制是什么？

版本控制

- 版本控制是一种在开发过程中用于管理我们对文件、目录、工程等内容的修改历史，方便查找和备份更改的历史记录的技术。简单来说就是用于管理多人协同开发项目的技术。（这也是我们为什么一定要学会使用git的原因！）
- 举个例子，你在做项目答辩的PPT时，肯定需要经过反复修改，然后会得到不同的新的版本（初稿、改版1、改版2、终稿、最终稿）一般来说你也许会将每个版本都备份保存一份，这个过程其实就是人工进行版本管理。而在面对多人协同开发项目的过程中，这样人工的控制肯定是行不通的，如果不采用版本控制，可能会导致最后多人改出来的版本不相同，极大的浪费人力成本。
- 版本控制包括三种

1. 本地版本控制

记录每次文件的更新，类似于我上面的举例，适用于个人

2. 集中版本控制

可以理解成中央集权，将所有的版本数据都保存在一台服务器上，协同开发者在服务器上同步更新或者上传自己的修改。获取时从服务器上获取最新的版本。**安全性好，但有风险**（服务器崩坏）

3. 分布式版本控制

所有版本信息仓库全部同步到本地的每个用户，实时更新，这样在服务器崩坏时，也可以从用户的本地仓库进行拷贝。**有一定安全隐患**（信息泄露）

- Git就是目前主流，也是最先进的分布式版本控制系统。

二、常用的Linux命令

在哪个文件夹右键打开git bash（后文会介绍！如果已经下载了git可以先跟着操作一下，没有的话后面也有下载的相关说明，下载完后可以尝试一下），就会默认在当前文件夹目录下

因为git bash使用的是Unix或Linux风格的命令行，因此我们会需要去学习一些基础的Linux命令

1. **cd** 进入目录

cd + 目标目录

2. **cd ..** 回退到上一级目录

cd与..之间要加空格！

3. **pwd** 显示当前所在的目录路径

4. **clear** 清屏

cls为windows的清屏命令

5. **ls(l)** 显示当前目录中的所有文件，ll列出的内容更加详细

6. **touch** 新建一个文件

7. **rm** 删除一个文件

8. **mkdir** 新建一个目录（文件夹）

9. **rm -r** 删除目录（文件夹）

rm与-r之间要加空格！

！切记！不要尝试在Linux中使用rm -rf /命令！因为在Linux中，一切皆文件，“/”代表根目录，也就是文件系统层级结构的最顶层目录，一旦输入该命令意味着删除电脑中所有文件！

10. **mv** 移动文件

mv+空格+要移动的文件+目标目录

11. **history** 查看历史命令

12. **help** 帮助

13. **exit** 退出

三、Git必要的配置

打开[Git官网](#)下载对应操作系统的版本，安装时基本上采用默认设置即可

在开始界面打开git，会发现有三个程序**Git Bash**、**Git CMD**、**Git GUI**，同时，对任意文件夹进行鼠标**右键**也可以看到。

- **Git Bash**: Unix或Linux风格的命令行界面
- **Git CMD**: Windows风格的命令行界面
- **Git GUI**: 图形界面的Git，相当于提供可视化方式来管理

我们在下载完git后**必须要**设置**用户名**和**邮箱**！（这是每次git提交都会使用的信息）下面是详细的步骤：

1. 找到C盘，打开用户文件夹，按照下列路径找到gitconfig文件

C:\Users\你的用户名称\.gitconfig

记得与git安装目录下 Git\etc\gitconfig文件相区分！

这个为系统的配置文件(system)，而C盘中的则是用户的配置文件(global)

2. 打开该文件，这里为git的用户配置
3. 打开git bash，输入下面的代码

git config --global user.name "此处输入你的用户名称"

git config --global user.email "此处输入你的邮箱"

4. 在系统中的配置文件里查看是否更改成功（本质即为读写文件）

查看系统配置的命令：**git config --system --list**

查看全局配置（用户自己配置的）的命令：**git config --global --list**

查看所有配置：**git config -l**（此处为字母L）

四、Git的工作原理

Git本地有三个工作区域：工作目录(Working Directory)、暂存区(Stage/Index)、资源库(Repository)。加上远程的git仓库(Remote Directory)就可以分为四个工作区域。文件在这四个区域之间的转换关系如下：

- Working Directory：工作区，也就是平时存放代码的地方
- Stage/Index：暂存区，临时存放改动，事实上只是一个文件
- Repository：本地仓库，安全存放数据的位置，里面有提交的所有版本数据
- Remote Directory：远程仓库，托管代码的服务器，用于数据交换

国外：GitHub

国内：Gitee

git的工作流程一般是这样的：

1. 在工作目录中添加、修改文件
2. 将需要进行版本管理的文件放入暂存区
3. 将暂存区域的文件提交到git仓库

因此，git管理的文件有三种状态：已修改(modified)、已暂存(staged)、已提交(commited)

五、Git项目搭建（基本命令操作）

创建仓库

创建本地仓库的方法有两种，一种是创建全新的仓库，另一种是克隆远程仓库

1. 本地仓库搭建

新建项目文件夹，打开git bash，输入**git init**命令初始化git项目文件，初始化后的项目文件会产生一个叫做.git的目录，即成功建立一个git代码库，里面包含了关于版本等的所有信息

.git目录为**隐藏文件**！必须在文件资源管理器中打开隐藏的项目的按钮才可以看到

2. 克隆远程仓库

在gitee/github中打开你想要克隆的项目文件，点击克隆后会出现一个链接，点击一下复制复制该链接，输入**git clone**+刚复制的链接，下载完毕后，目录中会出现克隆的项目文件

Git文件操作

文件的四种状态

1. **untracked**:未跟踪。即此文件在文件夹中，但未加入到git库，不参与版本控制。通过**git add**使状态变为**staged**
2. **unmodify**:已入库，未修改。即版本库中的文件最新版本内容与文件夹中一致。修改后会变为**modified**，也可通过**git rm**移出版本库，成为**untracked**文件
3. **modified**:已修改。可通过**git add**进入暂存状态，或使用**git checkout**丢弃修改，回到**unmodify**状态
4. **staged**:暂存。执行**git commit**将修改同步到库中，此时库中文件和本地文件内容一致，文件为**unmodify**状态，执行**git reset**取消暂存，文件状态为**modified**

文件首先在我们的工作区创建，我们使用git add命令添加到暂存区，再使用git commit命令提交到本地仓库，最后使用git push将文件提交到远程仓库上。反过来，从远程仓库将代码传到本地，我们使用git pull命令，将本地仓库提交的文件回滚到暂存区使用git reset，暂存区还未提交的文件传回本地我们则使用git checkout

1. **git status**：查看当前文件夹下所有文件状态

git status+文件名称-->查看指定文件状态

2. **git add**：将文件添加到暂存区

git add+文件名称-->对指定文件进行操作

git add+空格+**(git add .)**-->将文件夹中所有文件添加到暂存区

3. **git commit -m**：将暂存区中的内容提交到本地仓库

-m “注释内容” 为提交时显示的注释内容

4. **git push**：将本地仓库的内容提交到远程仓库

注册gitee/github账号

注册账号后记得配置SSH公钥！配置教程详见网站，关于更多git相关的命令和内容，推荐大家使用这个网站进行搜索 [Git大全](#) 如果是想多人共同合作一个项目的话还需学习分支部分的内容 这个教程仅作为引入，不是很详细，主要是帮助大家理解git是什么，具体的使用技巧还需在平时的使用过程中多多练习！