目录

[一：什么是Git 1](#_Toc27004935)

[二：常用术语 1](#_Toc27004936)

[三：使用Git方式导入项目 2](#_Toc27004937)

[四：创建项目（github） 3](#_Toc27004938)

[五：修改代码 4](#_Toc27004939)

[六：git安装 5](#_Toc27004940)

[七：基本命令 5](#_Toc27004941)

[八：理解工作区与暂存区的区别 8](#_Toc27004942)

[九：Git撤销修改和删除文件操作 8](#_Toc27004943)

[十：远程仓库 9](#_Toc27004944)

[十一：创建与合并分支 10](#_Toc27004945)

[十二：多人协作 11](#_Toc27004946)

# 一：什么是Git

简单说就是用于管理多人协同开发项目的技术。

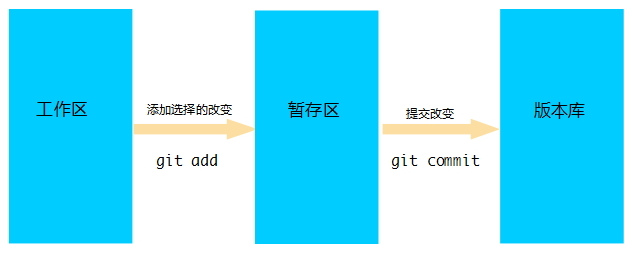
# 二：常用术语

**1)、仓库：**受版本控制的所有文件修订历史的共享数据库

**2)、工作空间：**本地硬盘或Unix 用户帐户上编辑的文件副本

**3)、工作树/区：**工作区中包含了仓库的工作文件。您可以修改的内容和提交更改作为新的提交到仓库

**4)、暂存区：**暂存区是工作区用来提交更改（commit）前可以暂存工作区的变化。



**5)、签入：**将新版本复制回仓库

**6 )、签出：**从仓库中将文件的最新修订版本复制到工作空间

**7)、提交：**对各自文件的工作副本做了更改，并将这些更改提交到仓库

**8)、冲突:** 多人对同一文件的工作副本进行更改，并将这些更改提交到仓库

**9)、合并：**将某分支上的更改联接到此主干或同为主干的另一个分支

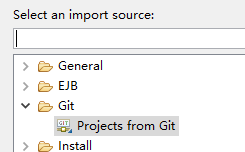
**10)、分支：**从主线上分离开的副本，默认分支叫master

**11)、修订：**表示代码的一个版本状态。

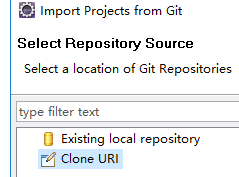
**12)、标记：**标记指的是某个分支某个特定时间点的状态。

# 三：使用Git方式导入项目

1.import->Git->Projects from Git



2.选择Clone URI

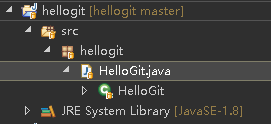


3.输入git参数

<https://github.com/smfzxl/zxl.git>

4.选择本地存放位置

把项目放在D:\project\hellogit 这个地方

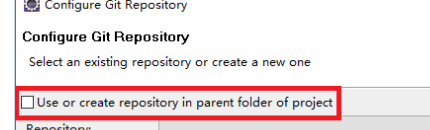


# 四：创建项目（github）

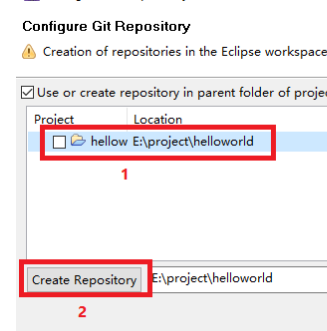
1.在Eclipse中创建一个java项目helloword

2.右键项目->Team->Share Project

3. 配置仓库点,击一下 Use or create repository in parent folder of project



4. 创建仓库



5.提交项目

1.右键项目->Team->Commit

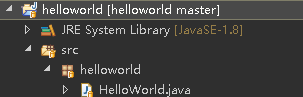
2.设置提交信息：test   
点击红框框，选中所有文件

6.设置git参数

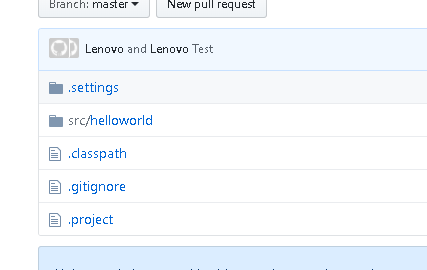
7.指定分支，使用默认的分支:master

8.提交成功

9.图标变化

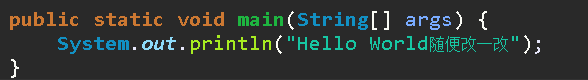


10.观察gitup上的项目



# 五：修改代码

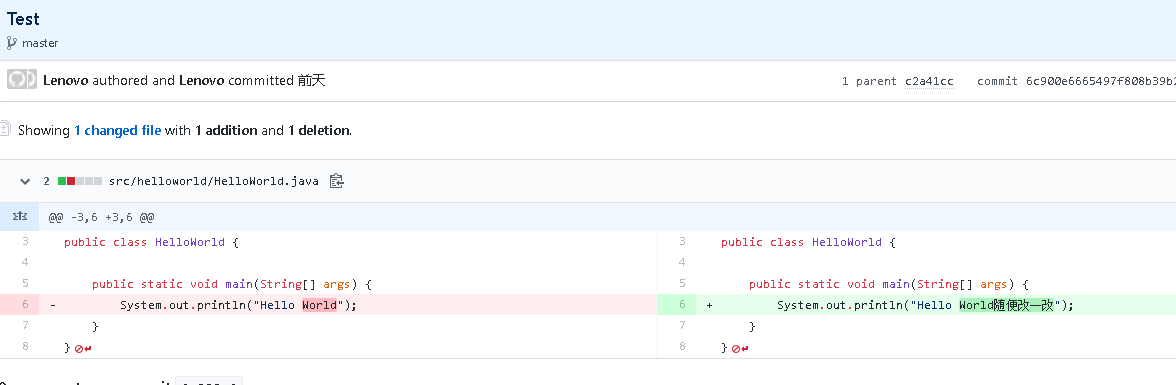
1.修改helloword



2.提交修改

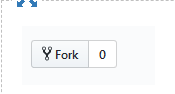
右键HelloWorld.java->Team->Commit

3.查看提交成功代码

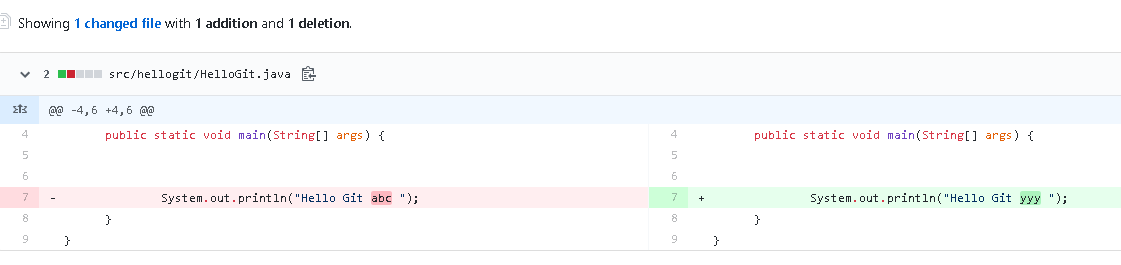


六：修改别人代码

1. 用自己账号登录gitup
2. 访问要修改的项目<https://github.com/how2j/hellogit>
3. 点击右上角的Fork按钮



1. 对项目进行修改
2. 回到github的hellogit项目下，然后点击New Pull Request，即发起一次合并请求。
3. 比较改变的地方



1. 填写修改原因点击Create pull request

# 六：git安装

下载安装包进行安装，

安装完成git –version检测版本是否安装成功



# 七：基本命令

1.创建目录



2.创建仓库（版本库）



3.添加test.txt到暂存区里面去，没有任何提示则添加成功



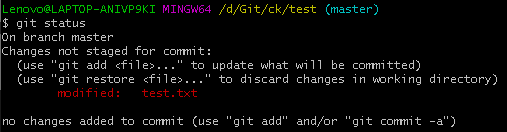
4.提交文件到仓库，其中-m 是本次提交的注释内容，这个必须填写



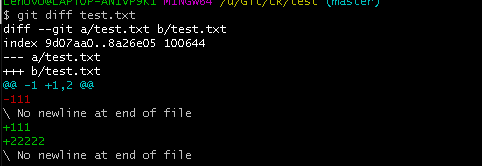
5．查看是否还有文件未提交



6.修改文件test里面的内容，继续使用git status查看，如下，说明文件已经被修改但是未被提交的修改



1. 查看修改内容，可以看到test.txt从一行111改成 二行 添加了一行22222内容



1. 将修改后的文件添加到暂存区



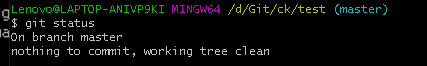
1. 查看状态，添加成功



11.提交到仓库



12.查看状态有没有未提交



13. 当前目录下新建一个index.js文件



14.删除js文件



15.删除文件夹



16.移动文件夹

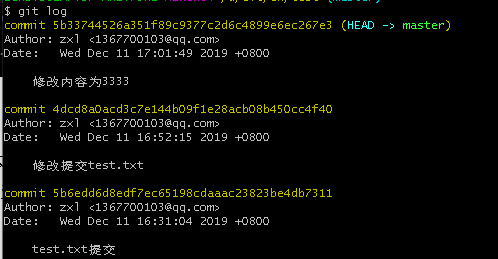


17.覆盖文件内容

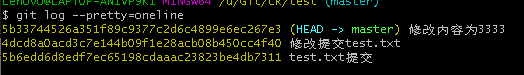


版本回退

1. 查看历史提交记录



1. 简化查看



1. 回退到上一个版本~后面是1就回退到前一个版本，2就是前两个版本，3就是前三个，以此类推



4.回退到上一个版本，那么如果要回退到上上个版本只需把HEAD^ 改成 HEAD^^ 以此

类推



5.查看文件内容



6.查看版本号



7.根据版本号回退



<https://www.cnblogs.com/best/p/7474442.html>

<https://www.cnblogs.com/tugenhua0707/p/4050072.html>

# 八：理解工作区与暂存区的区别

工作区：就是你在电脑上看到的目录test里的文件(.git隐藏目录版本库除外)，或者以后需要再新建的目录文件等等都属于工作区范畴。

**版本库(Repository)：**工作区有一个隐藏目录.git,这个不属于工作区，这是版本库。其中版本库里面存了很多东西，其中最重要的就是stage(暂存区)，还有Git为我们自动创建了第一个分支master,以及指向master的一个指针HEAD。

使用Git提交文件到版本库有两步：

  第一步：是使用 git add 把文件添加进去，实际上就是把文件添加到暂存区。

  第二步：使用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上。

# 九：Git撤销修改和删除文件操作

1. 撤销修改



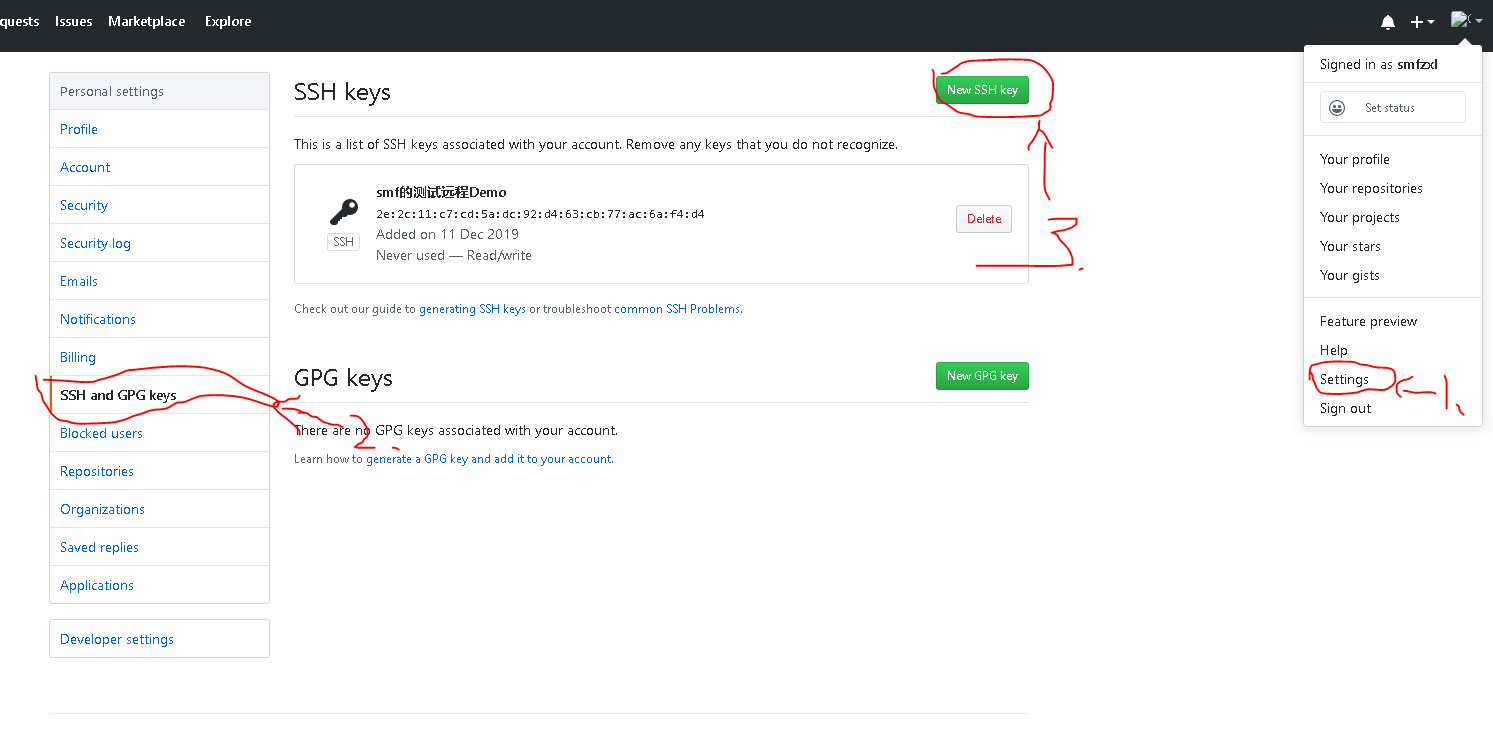
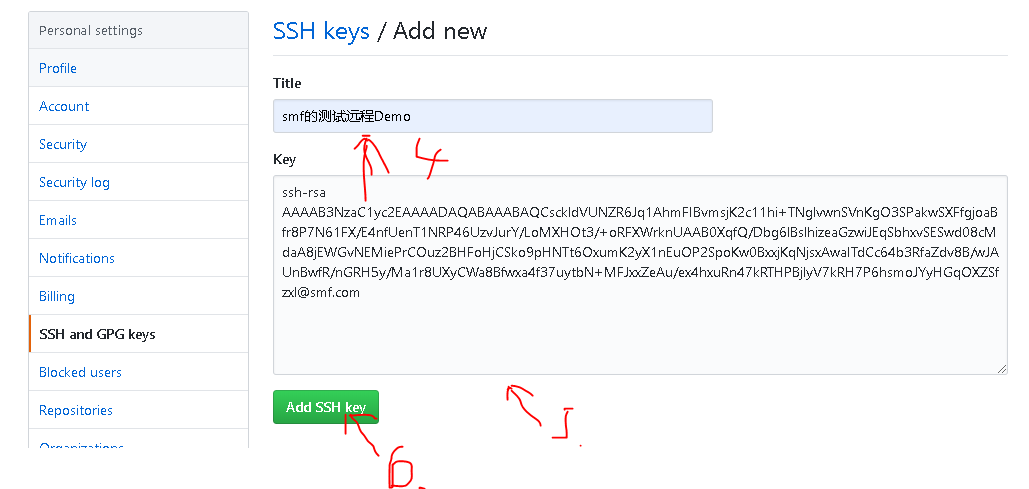
**注意：**命令git checkout -- readme.txt 中的 -- 很重要，如果没有 -- 的话，那么命令变成创建分支了。

1. 删除文件

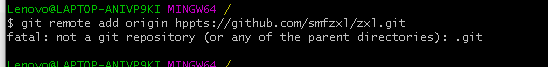


# 十：远程仓库

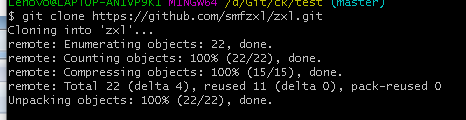
在了解之前，先注册github账号，由于你的本地Git仓库和github仓库之间的传输是通过SSH加密的，所以需要一点设置：

1. 创建SSH Key。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有id\_rsa和id\_rsa.pub这两个文件，如果有的话，直接跳过此如下命令，如果没有的话，打开命令行，输入如下命令：ssh-keygen  -t rsa –C [zxl@smf.com](mailto:zxl@smf.com)
2. 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。
3. 登录github,打开” settings”中的SSH Keys页面，然后点击“Add SSH Key”,填上任意title，在Key文本框里黏贴id\_rsa.pub文件的内容。
4. 
5. 

6. 目前，在GitHub上的这个testgit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。



7.从远程库克隆



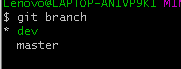


# 十一：创建与合并分支

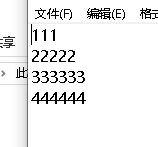
1．创建并切换分支dev



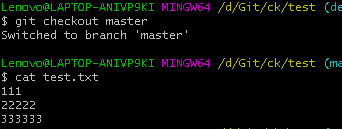
2.查看当前分支



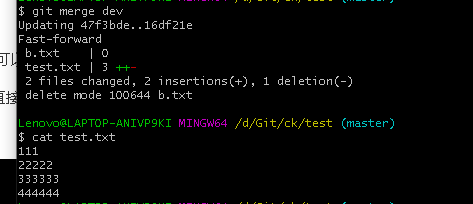
3.在dev分支上增加4444



4.切换分支master444不见，因为是在dev上增加的



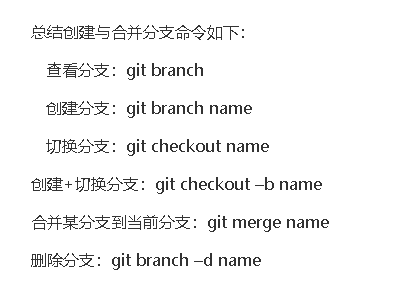
5.合并分支把dev分支上的内容合并到分支master上了，可以在master分支上，使用如下命令 git merge dev



6.删除分支



7.总结



# 十二：多人协作

 当你从远程库克隆时候，实际上Git自动把本地的master分支和远程的master分支对应起来了，并且远程库的默认名称是origin。

1. 要查看远程库的信息 使用 git remote
2. 要查看远程库的详细信息 使用 git remote –v

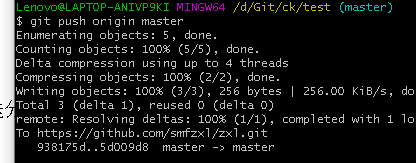
1.查看远程库信息

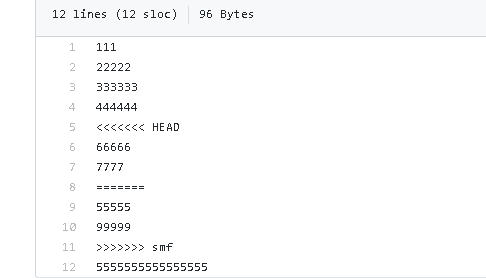


2.详细信息



3.推送分支

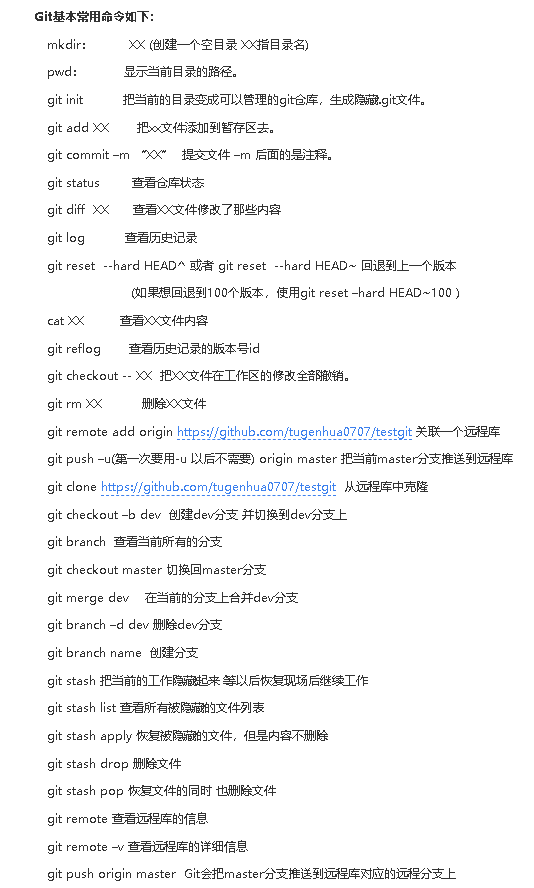




因此：多人协作工作模式一般是这样的：

1. 首先，可以试图用git push origin branch-name推送自己的修改.
2. 如果推送失败，则因为远程分支比你的本地更新早，需要先用git pull试图合并。
3. 如果合并有冲突，则需要解决冲突，并在本地提交。再用git push origin branch-name推送。

**Git基本常用命令如下：**



git fetch(从该URL中获取所有分支到本地存储库)