学习笔记

一、学习资料来源

链接：

1. <https://blog.csdn.net/weixin_43908355/article/details/140097124?sharetype=blogdetail&shareId=140097124&sharerefer=APP&sharesource=m0_74027483&sharefrom=link>
2. https://blog.csdn.net/m0\_68987050/article/details/146215006?sharetype=blogdetail&shareId=146215006&sharerefer=APP&sharesource=m0\_74027483&sharefrom=link

实践流程

1. 本地安装配置Git：在Git官网，下载Git安装包。按照安装向导的提示进行安装。安装完成后，打开命令行工具，输入git --version检查是否安装成功，若显示版本号则表示安装配置完成。

2. 创建本地仓库：创建仓库文件夹，打开命令行工具，使用cd命令进入该文件夹。然后执行git init命令，此时会在该文件夹下生成一个隐藏的.git文件夹，标志着本地仓库创建成功。

3. 注册远程仓库账号：选择一个远程仓库平台，如GitHub。在其官方网站上进行账号注册，填写相关信息并完成验证。

4. 创建公开远程仓库：登录已注册的远程仓库账号，在账号主页上找到创建仓库的选项，填写仓库名称、描述等信息，选择仓库类型为公开，点击创建按钮。

5. 选择公开本地代码推送到远程仓库：在本地仓库中添加一些代码文件，使用git add命令将文件添加到暂存区，再使用git commit命令提交到本地仓库，提交时添加适当的提交信息。然后将本地仓库与远程仓库进行关联，使用git remote add origin命令（将远程仓库地址替换为实际的远程仓库地址）。最后使用git push -u origin master命令将本地代码推送到远程仓库。

6. 多次提交代码并观察变更：在本地对代码进行修改，重复执行git add、git commit和git push操作，每次提交时注意添加不同的提交信息。在远程仓库中查看每次提交的记录，对比代码的变更情况。

二、遇到的困难及解决方法

1. 远程仓库关联问题：在将本地仓库与远程仓库关联时，可能会出现连接错误等情况。解决方法是检查远程仓库地址是否正确，确保网络连接正常。若地址无误但仍无法连接，可尝试重新生成SSH密钥，并将其添加到远程仓库账号中。

2. 代码提交问题：在提交代码时，可能会出现提交失败的情况。解决方法是注意提交信息的规范，避免使用特殊字符等可能导致错误的内容。

三、心得

通过本次实践，我对Git的安装配置及远程仓库的使用有了更深入的了解。在实践过程中，遇到问题并解决问题的过程让我积累了宝贵的经验。学会了如何在本地和远程仓库之间进行代码的同步和管理，这对于开发非常重要。同时，也认识到在操作过程中要注重细节，如安装配置的准确性、代码提交信息的规范性等。在未来的学习和工作中，我会继续使用Git进行代码管理，并不断学习其更多的功能和应用场景。