**一、学习资料来源**

1. Git官方文档：<https://git-scm.com/book/zh>
2. GitHub官方指南：<https://docs.github.com/zh>
3. 廖雪峰Git教程：<https://www.liaoxuefeng.com/wiki/896043488029600>
4. Gitee帮助中心：<https://gitee.com/help>

**二、实践流程**

**1. 本地Git环境搭建**

* **安装Git**：从官网下载对应系统的安装包，按默认配置完成安装；在命令行工具中验证安装是否成功。
* **基础配置**：设置全局用户名和邮箱，用于标识提交者身份
* **创建本地仓库**：新建项目文件夹并初始化为Git仓库；创建基础文件（如README.md），完成首次提交。

**2. 远程仓库操作**

* **注册账号**：选择GitHub或Gitee平台完成注册，验证邮箱。
* **创建远程仓库**：在平台新建公开仓库，保留默认初始化选项。
* **关联本地与远程仓库**：生成SSH密钥并添加到平台账户，建立安全连接；将本地仓库推送到远程仓库的指定分支。
* **三次代码提交实践**：第一次提交：初始化项目结构与基础文档；第二次提交：添加核心功能代码文件；第三次提交：修复文档中的描述错误；每次提交后通过对比工具观察文件变更差异。

**3. 学习笔记整理**

* **创建公开仓库**：单独建立仓库存放本次实践的学习笔记。
* **笔记内容组织**：按学习来源、操作步骤、问题解决、心得四部分撰写；记录配置过程中关键节点的截图或命令行输出。

**三、遇到的困难与解决方法**

**问题1：SSH连接认证失败：**推送代码时提示权限错误。

* **解决**：重新生成SSH密钥对，确保密钥类型与平台要求匹配；核对公钥内容是否完整粘贴到平台账户设置；通过连接测试命令验证通信是否正常。

**问题2：代码冲突导致推送失败：**远程仓库有更新未同步到本地时无法推送。

* **解决**：使用拉取命令合并远程修改到本地；手动解决冲突文件中的代码差异；重新提交并完成推送。

**问题3：中文路径显示异常：**命令行查看文件状态时出现乱码。

* **解决**：修改Git配置参数，关闭路径转义功能。

**四、实践心得**

1. **环境配置的重要性**：SSH密钥、用户信息等基础配置直接影响后续操作流畅度，需反复验证
2. **提交原子化**：细粒度提交便于追溯修改历史，避免混杂多个功能的变更
3. **远程协作思维**：理解本地与远程仓库的同步机制，掌握**push**/**pull**的核心逻辑
4. **文档即代码**：README等说明文件需随项目进度持续更新，保持与代码的一致性
5. **可视化工具价值**：配合Git图形界面工具（如SourceTree）可直观理解分支结构