

# 练习3：模拟团队开发：将项目推送到 GitHub，并尝试进行团队协作开发

2025年9月4日 11:25

1. 把本地项目推送到远程仓库（这里我们用 GitHub 举例，你也可以用 Gitee 或自己云服务的 Git 服务器）。
2. 多人协作开发，体验拉取、推送、合并的完整流程。

## 练习 3：模拟团队开发

### 步骤 1：准备一个远程仓库

1. 在 GitHub 上新建一个仓库（例如 team-demo）。
  - 登录 GitHub → 点击右上角 New repository。
  - 填写仓库名 → 选择 Public 或 Private。
  - 不要勾选初始化 README（避免和本地冲突）。
2. 复制仓库地址，例如：

<https://github.com/your-username/team-demo.git>

### 步骤 2：将本地项目推送到远程

在本地项目目录下执行：

```
# 初始化仓库（如果还没初始化）
git init
# 连接远程仓库
git remote add origin https://github.com/your-username/team-demo.git
# 把本地代码推送到远程 main 分支
git branch -M main
git push -u origin main
```

执行后，你的项目代码就上传到 GitHub 了。

### 步骤 3：队友克隆仓库

你的队友执行：

```
git clone https://github.com/your-username/team-demo.git
```

这会把项目下载到本地。

### 步骤 4：分支协作开发

## 4.1 你创建一个功能分支

```
git checkout -b feature-login
```

修改文件后提交：

```
git add .
git commit -m "添加登录功能"
git push origin feature-login
```

## 4.2 队友拉取并开发新功能

队友可以基于 main 创建另一个分支：

```
git checkout -b feature-register
修改文件后提交：
```

```
git add .
git commit -m "添加注册功能"
git push origin feature-register
```

## 步骤 5：合并分支

1. 在 GitHub 上发起 Pull Request (PR)

- 你和队友都可以在网页上对 feature-login / feature-register 发起 PR → 合并到 main。
- 也可以本地合并：

```
git checkout main
git pull origin main # 更新最新 main
git merge feature-login
git push origin main
```

2. 队友也需要同步：

```
git pull origin main
```

## 步骤 6：解决冲突

如果两个人修改了同一个文件的同一行，合并时会出现冲突。

冲突解决步骤：

1. 打开冲突文件，手动选择保留的内容。
2. 修改完毕后执行：

```
git add 文件名
git commit
git push origin main
```

## 步骤 7：完整协作流程总结

1. **仓库初始化**: 创建远程仓库 → 推送本地项目。
2. **团队协作**: 每人 clone 仓库 → 在独立分支开发功能。
3. **提交代码**: 本地 git commit → git push origin 分支名。
4. **合并代码**: 通过 Pull Request 或本地 merge 合并到 main。
5. **冲突解决**: 手动解决冲突 → 提交 → 推送。
6. **保持同步**: 所有成员定期执行:

```
git pull origin main
```

⚡ 练习目标达成后，你和队友就能真正体验团队开发的流程了。