

修改文件后上传代码至远程仓库

1. 开始工作前，先拉取远程最新代码
2. 在本地修改文件
3. 查看修改状态bash
4. 暂存修改bash
5. 提交到本地仓库bash
6. 推送到远程仓库bash
7. 为什么要先拉取?
8. 两种场景的区别

1. 开始工作前，先拉取远程最新代码

确保本地代码是最新的，避免后续推送冲突：bash

```
1  git pull origin 分支名 # 例如 git pull origin main
```

2. 在本地修改文件

编辑你的文件（如 `1.txt`）并保存。

3. 查看修改状态bash

```
1  git status
```

4. 暂存修改bash

▼

Bash |

```
1  git add 1.txt # 或 git add . 暂存所有修改
```

5. 提交到本地仓库bash

▼

Bash |

```
1  git commit -m "描述修改内容"
```

6. 推送到远程仓库bash

▼

Bash |

```
1  git push origin 分支名 # 例如 git push origin main
```

7. 为什么要先拉取？

- 防止远程仓库有其他人的新提交，导致你的推送被拒绝（就像之前你遇到的 `rejected` 错误）
- 确保你的修改是基于最新版本进行的，减少合并冲突

8. 两种场景的区别

场景	流程	适用情况
纯本地仓库	修改 → add → commit	个人项目、无需远程备份
含远程仓库	pull → 修改 → add → commit → push	多人协作、需要远程备份