回退版本

一、先做准备：找到要回退的目标版本

不管哪种场景，第一步都是确定“要回到哪个版本”，操作只有 1 条命令：

```bash
# 查看简洁的提交历史（每行是一个版本）
git log --oneline
```

执行后会看到类似这样的内容（和你之前的记录对应）：

```
1a29d33 (HEAD -> main, origin/main) Merge branch 'main'...  # 最新的合并版本
acac61f 已经修改  # 你改1.txt的版本（要回退的“错误版本”）
e751f9a Create README.md  # 改1.txt之前的版本（你的“目标版本”）
ac759c2 ...  # 更早的版本
```

- 记住目标版本的 **前 7 位字符**（比如 `e751f9a`），后面要用。

二、场景 1：你改了文件，但还没推到远程（只回退本地）

比如：你在本地改了 1.txt、提交了（`git commit`），但还没执行 `git push`，现在想把 1.txt 恢复到修改前。

1. 执行回退命令（用 `--hard` 彻底恢复，简单直接）：

```bash
git reset --hard e751f9a  # 把 e751f9a 换成你的目标版本前7位
```

2. 验证：打开 1.txt，会发现内容已经回到修改前了。

三、场景 2：你改了文件，还推到远程了（要同步回退远程）

比如：你已经执行了 `git push`，把改 1.txt 的版本推到 GitHub 了，现在想让远程的 1.txt 也恢复到修改前。

1. 先把本地回退到目标版本（和场景 1 一样）：

```bash
git reset --hard e751f9a  # 本地先恢复
```

2. 强制把本地的回退同步到远程（因为远程有“错误版本”，需要覆盖）：

```bash
git push -f origin main  # main是你的分支名，一般都是main
```

3. 验证：去 GitHub 上看 1.txt，内容会和本地一致，回到修改前了。

四、补充：如果回退错了，怎么恢复?

比如：你本来想回退到 A 版本，不小心回退到 B 版本了，想改回来。

1. 查看所有操作记录（包括你刚才的回退操作）：

```bash
git reflog
```

会看到类似这样的记录，找到你“回退前的正确版本”（比如 `acac61f`）：

```
e751f9a HEAD@{0}: reset: moving to e751f9a  # 你刚才的回退操作
acac61f HEAD@{1}: commit: 已经修改  # 你改1.txt的版本（想恢复到这里）
```

2. 恢复到正确版本：

```bash
git reset --hard acac61f  # 把 acac61f 换成你要恢复的版本
```

五、一句话总结

- 只改本地：`git log --oneline` 找目标 ID → `git reset --hard 目标ID`
- 改本地 + 远程：`git reset --hard 目标ID` → `git push -f origin main`
- 回退错了：`git reflog` 找正确 ID → `git reset --hard 正确ID`