backend-plus-02

Task.01 Git——不仅仅是commit

当你执行git init或者git clone时，你是否注意到工作区出现的.git文件？你知道它是什么吗？里面都放了什么？

1.git的分支是什么？如何创建/切换/合并分支？merge和rebase的区别是什么？

- **分支**：就像一条条并行的“小路”，你可以在一个分支上改代码，不影响主路（main）。
- **创建分支**：`git branch feature-login`
- **切换分支**：`git checkout feature-login` 或 `git switch feature-login`
- 合并分支：

  merge：把两个分支的内容“拼”在一起，会生成一个“合并提交”。

  rebase：把你的修改“搬”到另一个分支后面，看起来像是一直在一条线上改，更干净。

merge是“合并”，rebase 是“重放”。

2.HEAD是什么，它有哪些状态，相对引用如何使用？

- **HEAD**：就是你当前所在的“位置”，比如你现在在哪个分支。
- 它的状态有三种：

  指向某个提交（commit）

  指向某个分支名（比如 main）

  指向一个标签（tag）
- 相对引用：

  HEAD~1：上一个提交

  HEAD~2：上上个提交

  HEAD^：同 HEAD~1

用来快速定位历史版本。

3.什么是远程分支？你知道main与origin/main的关系吗？

- 远程分支：别人仓库里的分支，比如 GitHub 上的 main。
- origin/main：是本地对远程main的“影子”。

  main：本地的主分支

  origin/main：从远程拉下来的主分支

每次 pull 就是把 origin/main 同步到你的 main。

4.什么是git冲突？如何解决？

- **冲突**：两个人改了同一段代码，Git 不知道该保留谁的。
- 解决方法：
  1. 手动打开冲突文件，看到 <<<<<<<, =======, >>>>>>> 标记
  2. 选择保留哪部分代码
  3. 删除标记，保存文件

4. git add 文件名→ git commit

5.fetch和pull有什么区别?

- git fetch：只下载别人改了什么，不自动合并，安全。
- git pull：等于 fetch + merge，直接把别人的东西合并进来。

推荐先 fetch 看看变化，再决定要不要 merge。

6.什么是锁定的main分支？什么是PR？为什么要这么做？有什么好处?

- **锁定 main 分支**：不允许直接 push 到 main，防止误操作。
- **PR（Pull Request）**：提一个申请，让别人审核你的代码再合并。
- 为什么要这样做?

  防止乱改代码

  让团队一起检查质量

  更安全、更规范

好处：代码更稳定，协作更清晰。

7.简单了解一下git flow工作流

git flow是一种标准开发流程：

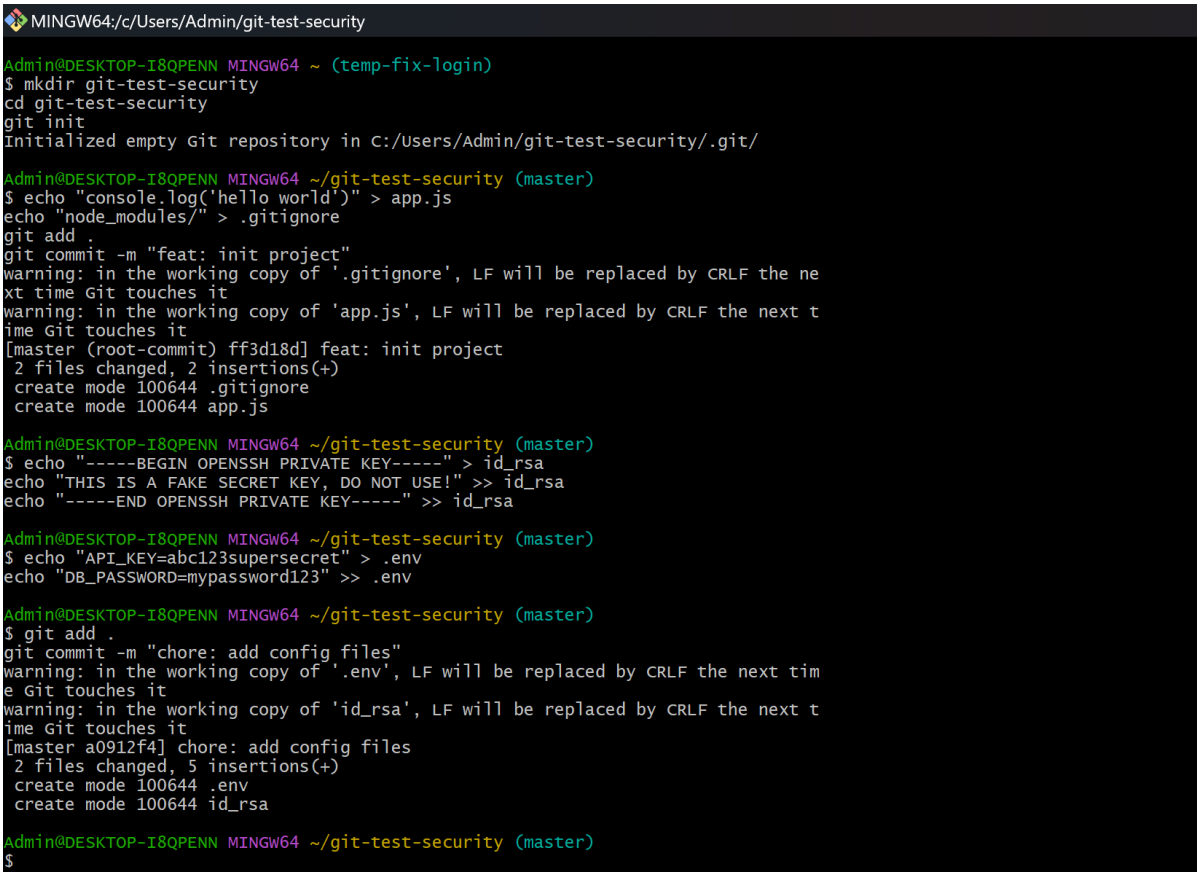
1. 从 develop分支开发新功能
2. 功能做好后，合并到 develop
3. 准备发布时，从 develop创建 release 分支
4. 发布后，合并到 main 和 develop
5. 修复 bug 用 hotfix 分支

结构清晰，适合团队项目。

总结：Git 不只是“add、commit、push”，它是团队协作的“交通规则”和“版本记录本”。掌握这些，就能高效又安全地写代码啦。

模拟场景01

1.先制造一个“误提交敏感文件”的错误场景



尝试修复——添加 .gitignore——更新 .gitignore——提交这个更改——模拟 push 到远程仓库（GitHub）被拒

```
Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ echo "id_rsa" >> .gitignore
echo ".env" >> .gitignore
echo "*.key" >> .gitignore

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git add .gitignore
git commit -m "fix: ignore sensitive files"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it
[master 42bf3c9] fix: ignore sensitive files
 1 file changed, 3 insertions(+)

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git log --oneline --name-only
42bf3c9 (HEAD -> master) fix: ignore sensitive files
.gitignore
a0912f4 chore: add config files
.env
id_rsa
ff3d18d feat: init project
.gitignore
app.js

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git remote add origin https://github.com/zhaohaozhen/zhz-eternity-01-repo

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git push origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/zhaohaozhen/zhz-eternity-
01-repo'

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ echo '#!/bin/bash
# 模拟远程仓库的安全检查
if git rev-list --all --grep="id_rsa" > /dev/null; then
    echo "检测到 id_rsa 文件，禁止推送！"
    exit 1
fi
if git rev-list --all --grep=".env" > /dev/null; then
    echo "检测到 .env 文件，禁止推送！"
    exit 1
fi
echo "安全检查通过，可以推送。"
' > check_security.sh
chmod +x check_security.sh

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ ./check_security.sh
检测到 id_rsa 文件，禁止推送！

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$
```

为什么第二次push还报错?

因为：

- 第一次 commit 时，提交了密钥文件 → 它已经进入 Git 的历史
- 虽然现在修改了 .gitignore并重新 commit，但 Git 不会自动删除已提交的文件
- 所以当 push时，系统仍然会检查所有提交，发现有敏感文件 → 报错！

```
Admin@DESKTOP-I8QPENN MINGW64 ~ (temp-fix-login)
$ pip install git-filter-repo
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: git-filter-repo in d:\steam2\lib\site-packages (2
.47.0)

Admin@DESKTOP-I8QPENN MINGW64 ~ (temp-fix-login)
$ git filter-repo --path "id_rsa" --replace ""
git filter-repo --path ".env" --replace ""
git-filter-repo: error: ambiguous option: --replace could match --replace-text,
--replace-message, --replace-refs
git-filter-repo: error: ambiguous option: --replace could match --replace-text,
--replace-message, --replace-refs
```

正确解决方法：从 Git 历史中删除该文件

步骤一：进入该文件 id_rsa

步骤二：使用 git filter-repo删除文件

1.先安装，pip install git-filter-repo；

```
Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ pip install --upgrade git-filter-repo
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: git-filter-repo in d:\steam2\lib\site-packages (2
.47.0)
```

2.再删除特定文件id_rsa：git filter-repo --path . --exclude "id_rsa"；

```
Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git filter-repo --path "id_rsa" --replace-text "" --force
NOTICE: Removing 'origin' remote; see 'Why is my origin removed?'
        in the manual if you want to push back there.
        (was https://github.com/zhaohaozhen/zhz-eternity-01-repo)
Parsed 3 commits
New history written in 0.38 seconds; now repacking/cleaning...
Repacking your repo and cleaning out old unneeded objects
HEAD is now at b71f5c7 chore: add config files
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Completely finished after 1.11 seconds.

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git filter-repo --path ".env" --replace-text "" --force
Parsed 1 commits
New history written in 0.21 seconds; now repacking/cleaning...
Repacking your repo and cleaning out old unneeded objects
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Completely finished after 0.86 seconds.

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
```

3.更新 .gitignore，最后推送新历史到远程：git push origin master --force-with-lease

```
Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git remote add origin https://github.com/zhaohaozhen/zhz-eternity-01-repo.git

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ echo "id_rsa" >> .gitignore
echo ".env" >> .gitignore
git add .gitignore
git commit -m "Add ignore for sensitive files"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
[master (root-commit) f01a6d6] Add ignore for sensitive files
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git push origin master --force-with-lease
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 246 bytes | 246.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/zhaohaozhen/zhz-eternity-01-repo/pull/new/master
remote:
To https://github.com/zhaohaozhen/zhz-eternity-01-repo.git
 * [new branch]      master -> master

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git log --oneline --name-only | grep "id_rsa"

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git log --oneline --name-only | grep ".env"

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
```

步骤四：再次 push：重新添加远程仓库（origin）——确保 .gitignore忽略敏感文件——强制推送新历史到远程

```
Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ echo "id_rsa" >> .gitignore
echo ".env" >> .gitignore
echo "id_rsa.pub" >> .gitignore
echo "*.pem" >> .gitignore

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git add .gitignore
git commit -m "Add ignore for private keys and env files"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
[master 8de8956] Add ignore for private keys and env files
 1 file changed, 4 insertions(+)

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
$ git push origin master --force-with-lease
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/zhaohaozhen/zhz-eternity-01-repo.git
   f01a6d6..8de8956  master -> master

Admin@DESKTOP-I8QPENN MINGW64 ~/git-test-security (master)
```

心得经验：不要直接删掉 .gitignore中的文件就 push， 不要以为改了 .gitignore就能“擦除”历史。Git 是版本控制系统，一旦提交，就不能“假装没发生”。必须主动清理历史。

模拟场景02

创建测试项目——创建 main 和 develop 分支——模拟微光娘在 develop 上开发登录功能，包括“未提交的更改”场景

```
Admin@DESKTOP-I8QPENN MINGW64 ~ (temp-fix-login)
$ mkdir micro-light-2025
cd micro-light-2025

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (temp-fix-login)
$ git init
Initialized empty Git repository in C:/Users/Admin/micro-light-2025/.git/
```

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (master)
$ git checkout -b main
Switched to a new branch 'main'

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (main)
$ echo "Welcome to MicroLight 2025!" > README.md
git add README.md
git commit -m "Initial commit"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it
[main (root-commit) 8a3eccf] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (main)
$ git checkout -b develop
Switched to a new branch 'develop'
```

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (main)
$ echo "Welcome to MicroLight 2025!" > README.md
git add README.md
git commit -m "Initial commit"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it
[main (root-commit) 8a3eccf] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ echo "function login(username, password) { return true; }" > login.js

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        login.js

nothing added to commit but untracked files present (use "git add" to track)
```

目标

1. 把她在 develop 上的修改合法化
2. 不破坏团队协作流程
3. 最终让她的代码合并进 develop，并删除她乱改的分支

为什么不能直接删掉 develop?

因为她已经把代码推到了 develop，如果直接删，会导致：

其他人无法拉取最新代码

可能造成数据丢失或混乱

所以我们要保留她的修改，只是换一种合规的方式让它进入主干。

解决方法

创建新的 feature 分支（用于修复）

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (temp-fix-login)
$ # 从 develop 创建一个新的 feature 分支（保留她的改动）
git checkout develop
git checkout -b feature/login-fix
Switched to branch 'develop'
Switched to a new branch 'feature/login-fix'
```

重置 develop 分支（回到干净状态）

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (feature/login-fix)
$ # 回退 develop 到上次合并前的状态（即没有 login.js 的状态）
git checkout develop
git reset --hard HEAD~1
Switched to branch 'develop'
fatal: ambiguous argument 'HEAD~1': unknown revision or path not in the working
tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
```

找到那个提交的哈希

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ git reflog
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{0}: checkout: moving from feature/login-fix to develop
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{1}: checkout: moving from develop to feature/login-fix
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{2}: checkout: moving from temp-fix-login to develop
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{3}: checkout: moving from develop to temp-fix-login
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{4}: checkout: moving from develop to develop
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{5}: checkout: moving from develop to develop
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{6}: checkout: moving from develop to develop
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{7}: checkout: moving from feature/login to develop
2b034e7 (feature/login) HEAD@{8}: checkout: moving from main to feature/login
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{9}: checkout: moving from develop to main
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{10}: checkout: moving from feature/login to develop
2b034e7 (feature/login) HEAD@{11}: commit: feat: implement login function
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{12}: checkout: moving from develop to feature/login
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{13}: checkout: moving from main to develop
8a3eccf (HEAD -> develop, origin/temp-fix-login, temp-fix-login, main, feature/l
ogin-fix) HEAD@{14}: commit (initial): Initial commit

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ # 切换到 develop
git checkout develop

# 创建新分支保存她的改动
git checkout -b feature/login-fix
Already on 'develop'
fatal: a branch named 'feature/login-fix' already exists
```

将 login.js 的改动移到 feature/login-fix分支

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ # 检查 feature/login 是否存在
git branch -a | grep feature/login

# 如果存在，从中取出 login.js
git checkout feature/login -- login.js
  feature/login
  feature/login-fix

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ git add login.js
git commit -m "Add login function [fix: direct commit to develop]"
[develop f576785] Add login function [fix: direct commit to develop]
 1 file changed, 1 insertion(+)
 create mode 100644 login.js
```

步骤四：清理本地 develop 分支

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ # 查看最近几次提交
git log --oneline -n 3

# 找到你要回退到的那个“干净”的提交（比如初始提交或上一次合并前）
# 假设要回退到 HEAD~1
git reset --hard HEAD~1
f576785 (HEAD -> develop) Add login function [fix: direct commit to develop]
8a3eccf (origin/temp-fix-login, origin/feature/login-fix, temp-fix-login, main, feature/login-fix) Ini
tial commit
HEAD is now at 8a3eccf Initial commit
```

步骤五：推送 feature 分支并申请 PR

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ # 切换到 feature 分支
git checkout feature/login-fix

# 确保 login.js 存在（如果没有就再取一次）
git checkout feature/login -- login.js

# 添加并提交
git add login.js
git commit -m "feat: add login function via PR"
Switched to branch 'feature/login-fix'
[feature/login-fix a61e27b] feat: add login function via PR
 1 file changed, 1 insertion(+)
 create mode 100644 login.js

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (feature/login-fix)
$ git push origin feature/login-fix
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 354 bytes | 354.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:zhaohaozhen/zhz-eternity-01-repo.git
   8a3eccf..a61e27b  feature/login-fix -> feature/login-fix
```

步骤六：审核后合并到 develop

```
Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (feature/login-fix)
$ git checkout develop
git pull origin develop
Switched to branch 'develop'
fatal: couldn't find remote ref develop

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ git fetch origin
git branch -r
  origin/HEAD -> origin/main
  origin/feature/login-fix
  origin/main
  origin/master
  origin/temp-fix-login

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ # 切换到 develop
git checkout develop

# 推送 develop 分支到远程
git push origin develop
Already on 'develop'
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/zhaohaozhen/zhz-eternity-01-repo/pull/new/develop
remote:
To github.com:zhaohaozhen/zhz-eternity-01-repo.git
 * [new branch]      develop -> develop

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ git pull origin develop
From github.com:zhaohaozhen/zhz-eternity-01-repo
 * branch            develop    -> FETCH_HEAD
Already up to date.

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ git merge origin/feature/login-fix
Updating 8a3eccf..a61e27b
Fast-forward
 login.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 login.js

Admin@DESKTOP-I8QPENN MINGW64 ~/micro-light-2025 (develop)
$ # 最关键的一条：
git merge origin/feature/login-fix
Already up to date.
```