

练习1：创建一个本地仓库，尝试进行多次提交并查看日志

2025年9月4日 10:58

1. 创建一个本地 Git 仓库

1.1 打开终端/命令行

- Windows 用户可以使用 Git Bash 或 CMD。
- macOS/Linux 用户可以直接打开终端。

1.2 创建一个新目录并进入

首先，在你希望创建 Git 仓库的文件夹内创建一个文件夹目录：

1.3 初始化 Git 仓库

使用 `git init` 命令来初始化一个新的 Git 仓库：

```
git init
```

此时，Git 会在你的 `my_project` 目录下创建一个 `.git` 文件夹，这个文件夹是 Git 用来跟踪版本信息的地方。

2. 创建并修改文件

2.1 创建一个新的文件

创建一个新文件，比如 `hello.txt`：

```
echo "Hello, Git!" > hello.txt
```

2.2 查看文件状态

运行 `git status` 命令查看当前文件状态：

```
git status
```

Git 会显示 `hello.txt` 文件处于 **未暂存** 状态 (Untracked)，即它还没有被 Git 跟踪。

3. 添加文件到暂存区

3.1 添加文件

使用 `git add` 命令将 `hello.txt` 添加到 Git 暂存区 (stage)：

```
git add hello.txt
```

3.2 查看文件状态

再次运行 `git status`，现在文件会显示为 **已暂存** (Staged)：

```
git status
```

Git 会提示 `hello.txt` 已被添加到暂存区，准备提交。

4. 提交更改

4.1 提交文件

使用 `git commit` 命令提交文件：

```
git commit -m "Add hello.txt with initial content"
```

`-m` 后面的 "Add hello.txt with initial content" 是本次提交的描述信息 (commit message)，描述本次提交的目的或更改内容。

5. 修改文件并提交第二次

5.1 修改文件

编辑 `hello.txt` 文件，加入新的内容

5.2 添加修改到暂存区

再次使用 `git add` 将修改后的文件添加到暂存区：

```
git add hello.txt
```

5.3 提交第二次更改

提交这个修改：

```
git commit -m "Update hello.txt with new content"
```

6. 查看 Git 提交日志

6.1 查看日志

使用 `git log` 查看提交日志：

```
git log
```

Git 会列出所有提交的历史记录，每次提交都有一个 commit hash，作者，日期，以及提交信息 (commit message)。

你应该会看到类似下面的输出：

```
commit a1b2c3d4e5f678901234567890abcdef12345678
Author: Your Name <youremail@example.com>
Date:   Mon Oct 2 12:34:56 2023 +0000
Update hello.txt with new content
commit 1234567890abcdef1234567890abcdef12345678
Author: Your Name <youremail@example.com>
Date:   Mon Oct 2 12:12:34 2023 +0000
Add hello.txt with initial content
```

6.2 查看提交历史 (简化版)

如果你不需要太多的细节，使用 `git log --oneline` 命令，只显示每次提交的简短版本：

```
git log --oneline
```

你会看到每次提交的 commit hash (缩略版) 和 提交信息：

```
alb2c3d Update hello.txt with new content
1234567 Add hello.txt with initial content
```

7. 回顾和撤销操作（可选）

7.1 查看更改（查看差异）

在修改文件后，你可以查看文件内容的差异：

```
git diff hello.txt
```

这会显示 `hello.txt` 文件的修改内容。

7.2 撤销修改

如果你想撤销某个未提交的更改（比如取消对文件的修改），可以使用：

```
git checkout -- hello.txt
```

7.3 回退到上一个提交

如果你想回到上一个提交，可以使用：

```
git reset --hard HEAD~1
```

这会将项目回到上一个提交的状态，**注意：这会丢失所有未提交的修改。**

8. 练习总结

- 你学会了如何使用 `git init` 初始化仓库，如何通过 `git add` 添加文件到暂存区，并使用 `git commit` 提交修改。
- 使用 `git log` 查看提交历史，能够回顾项目的变动。
- 练习了文件修改、提交和查看修改历史的基本操作。