

- [第3讲 Git 命令](#)
- [常用命令](#)
- [设置用户签名](#)
- [初始化本地库](#)
- [查看本地库状态](#)
- [添加暂存区](#)
- [提交本地库](#)
- [修改文件](#)
- [历史版本](#)
- [版本穿梭](#)

第3讲 Git 命令

常用命令

命令名称	作用
<code>git config --global user.name</code>	用户名 设置用户签名
<code>git config --global user.email</code>	邮箱 设置用户签名
<code>git init</code>	初始化本地库
<code>git status</code>	查看本地库状态
<code>git add 文件名</code>	添加到暂存区
<code>git commit -m "日志信息" 文件名</code>	提交到本地库
<code>git reflog</code>	查看历史记录
<code>git reset --hard 版本号</code>	版本穿梭

设置用户签名

- 第一次用 `Git` 要执行的命令，配置用户，邮箱，如果不设置提交代码时会报错。
- 基本语法：

```
git config --global user.name 用户名
git config --global user.email 邮箱
```

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Desktop
$ git config --global user.name yaoxin

hp@LAPTOP-ARLL3DSO MINGW64 /e/Desktop
$ git config --global user.email 454115408@qq.com

hp@LAPTOP-ARLL3DSO MINGW64 /e/Desktop
$ cat ~/.gitconfig
[user]
    name = yaoxin
    email = 454115408@qq.com
[core]
    excludesfile = C:/Users/hp/git.ignore
```

说明：签名的作用是区分不同操作者身份。用户的签名信息在每一个版本的提交信息中能够看到，以此确认本次提交是谁做的。Git 首次安装必须设置一下用户签名，否则无法提交代码。

注意：这里设置用户签名和将来登录 GitHub（或其他代码托管中心）的账号没有任何关系。（只代表本地 windows 用户。）

初始化本地库

- 基本语法

```
git init
```

- 创建 E:\Git-Space\git-test 本地库

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test
$ git init
Initialized empty Git repository in E:/Git-Space/git-test/.git/

hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ ll
total 0

hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ ll -a
total 4
drwxr-xr-x 1 hp 197610 0 8月 8 17:15 ./
drwxr-xr-x 1 hp 197610 0 8月 8 17:15 ../
drwxr-xr-x 1 hp 197610 0 8月 8 17:15 .git/

hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
```

查看本地库状态

- 基本语法

```
git status
```

1. 首次查看

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

2. 新增文件 test.txt

3. 再次查看（检测到未追踪的文件）

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.txt

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

hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
```

添加暂存区

- 基本语法

```
git add 文件名
```

1. 将工作区的文件添加到暂存区

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git add test.txt
```

2. 查看状态（检测到暂存区有新文件）

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test.txt

hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
```

3. 删除暂存区文件

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git rm --cached test.txt
rm 'test.txt'

hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.txt

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

提交本地库

- 基本语法

```
git commit -m "日志信息" 文件名
```

1. 将暂存区的文件提交到本地库

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git commit -m "第一次提交" test.txt
[master (root-commit) c5bbd23] 第一次提交
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
```

2. 查看状态 (没有文件需要提交)

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master
nothing to commit, working tree clean
```

3. 查看日志

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reflog
c5bbd23 (HEAD -> master) HEAD@{0}: commit (initial): 第一次提交
```

4. 查看详细日志

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git log
commit c5bbd231be30a3dd7bb9c5e0f33f5a5494230375 (HEAD -> master)
Author: yaoxin <454115408@qq.com>
Date: Mon Aug 8 17:28:26 2022 +0800
```

第一次提交

修改文件

1. 查看文件

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ cat test.txt
```

2. 修改文件

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ vim test.txt
```

3. 再次查看文件

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ cat test.txt
xyxyxyxyx
xyxyxyxyxyxyx
```

4. 查看文件状态

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

5. 将修改的文件再次添加暂存区

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git add test.txt
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory
```

6. 查看文件状态

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   test.txt
```

7. 提交文件

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git commit -m "第二次提交" test.txt
```

```
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory
[master 985076e] 第二次提交
1 file changed, 3 insertions(+)
```

8. 查看文件状态

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git status
On branch master
nothing to commit, working tree clean
```

9. 查看日志

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reflog
985076e (HEAD -> master) HEAD@{0}: commit: 第二次提交
c5bbd23 HEAD@{1}: commit (initial): 第一次提交
```

历史版本

- `git reflog` 查看版本信息

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reflog
5228695 (HEAD -> master) HEAD@{0}: commit: 第三次提交
985076e HEAD@{1}: commit: 第二次提交
c5bbd23 HEAD@{2}: commit (initial): 第一次提交
```

- `git log` 查看版本详细信息

```
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git log
commit 5228695d92c796c22754fa17887135b6d7cec3b4 (HEAD -> master)
Author: yaoxin <454115408@qq.com>
Date: Tue Aug 9 08:24:03 2022 +0800
```

第三次提交

```
commit 985076e001184b393025ec518cfddc50a4922505
Author: yaoxin <454115408@qq.com>
Date: Tue Aug 9 08:22:14 2022 +0800
```

第二次提交

```
commit c5bbd231be30a3dd7bb9c5e0f33f5a5494230375
Author: yaoxin <454115408@qq.com>
Date: Mon Aug 8 17:28:26 2022 +0800
```

第一次提交

版本穿梭

- 基本语法

```
git reset --hard 版本号
```

```
// 首先查看当前的历史记录, 可以看到当前版本是在 5228695
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reflog
5228695 (HEAD -> master) HEAD@{0}: commit: 第三次提交
985076e HEAD@{1}: commit: 第二次提交
c5bbd23 HEAD@{2}: commit (initial): 第一次提交

// 切换到 985076e 版本, 也就是第二次提交的版本
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reset --hard 985076e
HEAD is now at 985076e 第二次提交

// 切换完毕之后再查看历史记录, 当前成功切换到了 985076e 版本
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reflog
985076e (HEAD -> master) HEAD@{0}: reset: moving to 985076e
5228695 HEAD@{1}: commit: 第三次提交
985076e (HEAD -> master) HEAD@{2}: commit: 第二次提交
c5bbd23 HEAD@{3}: commit (initial): 第一次提交

// 查看文本, 已经回到第二次提交的内容
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ cat test.txt
yxyxyxyxyx
xyxyxyxyxyxyxyx

// 查看日志
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reflog
985076e (HEAD -> master) HEAD@{0}: reset: moving to 985076e
5228695 HEAD@{1}: commit: 第三次提交
985076e (HEAD -> master) HEAD@{2}: commit: 第二次提交
c5bbd23 HEAD@{3}: commit (initial): 第一次提交

// 穿梭到第一次提交版本 c5bbd23
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
```



```
$ git reset --hard c5bbd23
HEAD is now at c5bbd23 第一次提交

// 查看日志, 观察是否切换到第一次提交
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reflog
c5bbd23 (HEAD -> master) HEAD@{0}: reset: moving to c5bbd23
985076e HEAD@{1}: reset: moving to 985076e
5228695 HEAD@{2}: commit: 第三次提交
985076e HEAD@{3}: commit: 第二次提交
c5bbd23 (HEAD -> master) HEAD@{4}: commit (initial): 第一次提交

// 穿梭到第三次提交版本 5228695
hp@LAPTOP-ARLL3DSO MINGW64 /e/Git-Space/git-test (master)
$ git reset --hard 5228695
HEAD is now at 5228695 第三次提交
```

- Git 切换版本, 底层其实是移动的 HEAD 指针。