

一、学习一些指令

ls: 当前工作列表下边的东西

cd 输入路径按 tab 键可以自动补全

cd .. 上级文件夹, 比如 cd ../document

cd /主目录

cat + 文件路径可以显示内容。比如 cat abc.txt

按上方向键↑可以把上一个命令调出来

绝对路径: 就是从根目录开始一层一层的写出

例如: cat c/users/document/abc.txt

相对路径: 不用/开头

例如: cat desktop/abc.txt

-a 或者—all

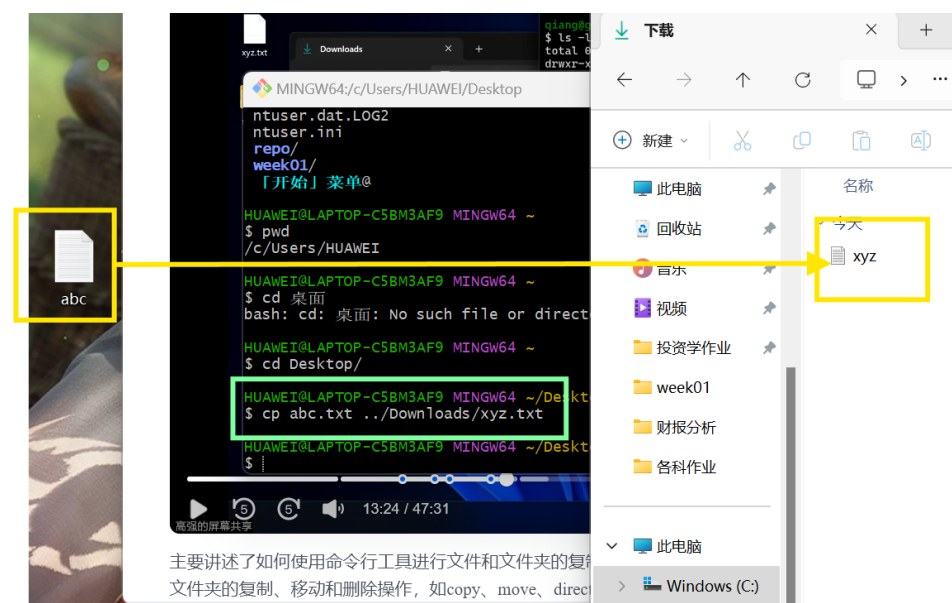
Cp:

复制单个的文件

例如: cp abc.txt ../downloads/(写出路径)

cp abc.txt ../downloads/xyz.txt (这个文件被复制到 downloads 下并且名字改变)

复制整个文件夹 cp -r store ../downloads/ (复制 store 这个文件夹, 同理可以改名字)



Mv:移动、重命名, 例子如下。

1. 移动文件

如果你想将一个文件从一个目录移动到另一个目录，可以使用 `mv` 命令。例如，将 `file.txt` 文件从当前目录移动到 `Documents` 目录：

```
bash ^  
  
mv file.txt Documents/
```

这里 `file.txt` 是源文件，`Documents/` 是目标目录。

2. 移动目录

移动目录的操作与移动文件类似。假设要将 `projects` 目录移动到 `workspace` 目录下：

```
bash ^  
  
mv projects/ workspace/
```

`projects/` 是源目录，`workspace/` 是目标目录。

3. 重命名文件

当源和目标都在同一目录下，且目标名称不同时，`mv` 命令可用于重命名文件。例如，将 `oldname.txt` 重命名为 `newname.txt`：

```
bash ^  
  
mv oldname.txt newname.txt
```

4. 重命名目录

同样，也可以使用 `mv` 命令重命名目录。比如将 `old_folder` 重命名为 `new_folder`：

```
bash ^  
  
mv old_folder/ new_folder/
```

Mkdir:创建文件夹

Rm:删除。（这个不能撤销，可以用-r）

Df (disk free) -h:看磁盘空间和剩余空间

Du(disk use)

使用 `-h` (human-readable) 选项，会将磁盘使用量以人类可读的格式显示，如 KB、MB、GB 等。

使用 `-s` (summarize) 选项，只会显示指定文件或目录的总大小，而不会显示其内部子目录和文件的详细信息。例如，要查看当前目录的总大小：`du -s`。若要查看 `Documents` 目录的总大小：`du -s Documents`

二、创建私密的代码仓库：

创建了代码仓库“mywork”，复制 ssh 代码

```
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo
$ pwd
/c/Users/HUAWEI/repo

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo
$ git clone git@gitcode.com:dayuwanzi/mywork.git
Cloning into 'mywork'...
warning: You appear to have cloned an empty repository.

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo
$ ls -l
total 4
drwxr-xr-x 1 HUAWEI 197609 0 Mar 16 21:45 mywork/
-rw-r--r-- 1 HUAWEI 197609 0 Mar 10 16:11 script1.py
drwxr-xr-x 1 HUAWEI 197609 0 Mar 12 22:54 week01/

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo
$ cd my work
bash: cd: too many arguments

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo
$ cd mywork/

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ ls -l
total 0

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ git log
fatal: your current branch 'main' does not have any commits yet
```

Git log 是看几次提交（图中为 0 次）

❌

`git add` 命令用于将工作目录中修改的文件添加到暂存区（也称为索引）。在 Git 里，工作目录是你实际编辑文件的地方，而暂存区是一个中间区域，你可以选择将哪些修改纳入到下一次提交中。

语法及示例

```
bash ^
git add <文件或目录>
```

- **添加单个文件：**假设你修改了项目中的 `example.txt` 文件，使用以下命令将其添加到暂存区：

```
bash ^
git add example.txt
```

- **添加多个文件：**若要同时添加 `file1.txt` 和 `file2.txt` 两个文件，可使用空格分隔文件名：

```
bash ^
git add file1.txt file2.txt
```

- **添加整个目录：**如果要 `src` 目录下的所有修改都添加到暂存区，可以这样操作：

```
bash ^
git add src/
```

- **添加所有修改：**使用 `.` 可以将当前目录下的所有修改（包括新增、修改和删除的文件）都添加到暂存区：

```
bash ^
git add .
```

3. git status

含义

`git status` 命令用于显示工作目录和暂存区的状态。它会告诉你哪些文件被修改了但还未添加到暂存区，哪些文件已经添加到暂存区准备提交，以及是否有新增或删除的文件等信息。

语法及示例

bash ^

```
git status
```

- **查看状态:** 在项目目录下执行该命令, 会输出类似以下的信息:

plaintext \wedge

```
On branch master
Your branch is up to date with 'origin/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:   example.txt

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

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

上述输出表明 `example.txt` 文件被修改了但还未添加到暂存区, `new_file.txt` 是一个新增的未被 Git 跟踪的文件。

2. git commit

含义

`git commit` 命令用于将暂存区中的内容永久保存到本地仓库中，形成一个新的提交记录。每次提交都会包含一个描述性的提交信息，用于说明这次提交做了哪些更改。

语法及示例

```
bash ^
git commit -m "提交信息"
```

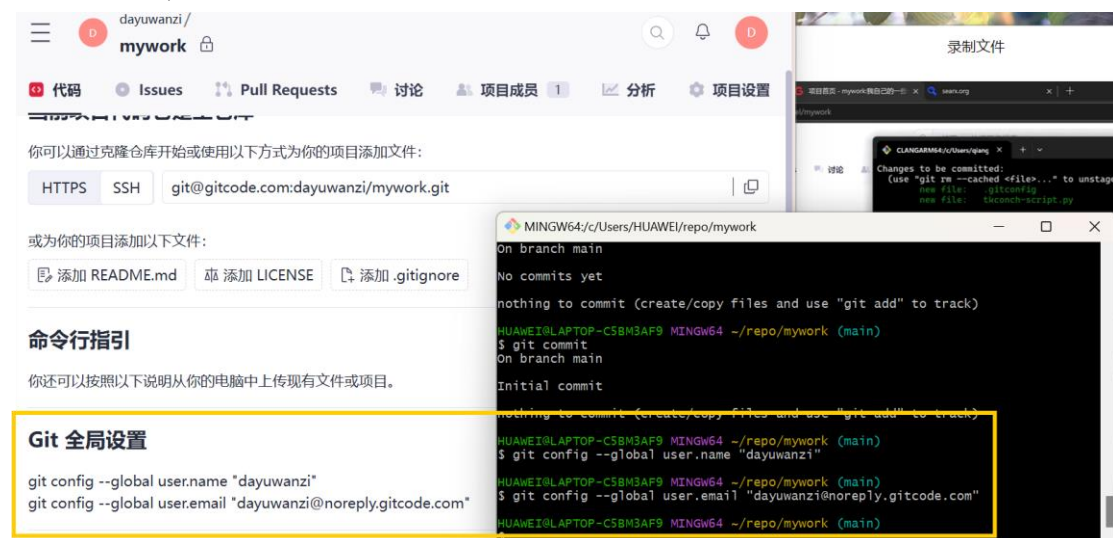
- **基本提交**：当你已经使用 `git add` 将修改添加到暂存区后，使用以下命令进行提交，并添加简要的提交信息：

```
bash ^
git commit -m "修复了 example.txt 文件中的拼写错误"
```

- **直接提交（跳过暂存区）**：如果你想要直接将工作目录中的修改提交，而不经暂存区，可以使用 `-a` 选项。不过，此选项只会提交已经被 Git 跟踪的文件，新增的文件仍需要先使用 `git add` 添加：

```
bash ^
git commit -am "更新了所有已跟踪文件的内容"
```

Git 全局设置，设置一次就好。



Git log 可以看提交历史。

自己尝试，在 mywork 中添加了 abcd.txt，放在暂存区并提交，使用一系列 git 代码。

```
HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ git add abcd.txt

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ git status
On branch main

No commits yet

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

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ git commit -m'^C

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ git commit -m"added some files for test"
[main (root-commit) da25e24] added some files for test
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 abcd.txt

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ git log
commit da25e246f319dda98372efd9542456d5b7e9329d (HEAD -> main)
Author: dayuwanzi <dayuwanzi@noreply.gitcode.com>
Date:   Sun Mar 16 22:43:40 2025 +0800

    added some files for test

HUAWEI@LAPTOP-C5BM3AF9 MINGW64 ~/repo/mywork (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Start Git Hooks Checking
To gitcode.com:dayuwanzi/mywork.git
 * [new branch]      main -> main
```

[PASSED]