Linux Terminal
Cheat Sheet

Linux Terminal: The Ultimate Cheat Sheet

If you're a Linux user, the Terminal is probably the most powerful tool you would ever have. But the thing about the Terminal is that you need to learn how to use it if you want to benefit from it.

For the last few months, I've been playing with the Terminal a lot, and I came up with a long list of useful commands that I use regularly. Please let me know if I missed something important so I can add it to future posts.

TL;DR

- **Basic commands**
  - Zoom in ➜ `[CTRL] + [+]`
  - Zoom out ➜ `[CTRL] + [-]`
  - Print working directory ➜ `pwd`
  - Clear the terminal ➜ `[CTRL] + [l]` or `clear`
  - Assign an alias ➜ `alias [alias-name]="[command-to-run]"`
  - Source a file ➜ `source [name-of-the-file-to-read-and-execute]`
- **Change directory command (cd)**
  - Move to a specific directory ➜ `cd [name-of-your-directory]`

    - Move to the parent directory ➜ `cd ..`
    - Move to the home directory ➜ `cd` or `cd ~`
    - Move to the last directory yo were in ➜ `cd -`

- **List command (ls)**
    - List all visible files and directories ➜ `ls`
    - List all files and directories (include hidden files) ➜ `ls -a`
    - Long Listed Format ➜ `ls -l`
    - Human Readable Format ➜ `ls -lh`
    - Combining arguments: Human Readable Format + Hidden files ➜ `ls -lah`
    - Learn more about the ls command ➜ `man ls`

- **Search**
    - Locate the binary for a program ➜ `which [name-of-the-program]`
    - Locate the binary, source and user manual for a program ➜ `whereis [name-of-the-program]`
    - Locate files and directories by name ➜ `find [path-to-search] -iname [name-of-the-file-you-want-to-search]`
        - Learn more about the find command ➜ `man find`
    - Get a brief description for a command ➜ `whatis [command-name]`

- **History**
    - Get previous commands (one by one) ➜ Use the `Up Arrow key` ⬆ to navigate your history
    - Get previous commands (full list) ➜ `history`.
    - Repeat commands from history (bang command) ➜ `history` ➜ `![number-of-the-command-to-repeat]`
    - Repeat last command (bang-bang command) ➜ `!!`

- **Working with files and directories**
    - Create a new file (without open it) ➜ `touch [name-of-your-file]`
    - Create a new file using a text editor ➜ `vim [name-of-your-file]` or `nano [name-of-your-file]`
    - Copy a file ➜ `cp [source-path-of-your-file] [destination-path-for-your-file]`
    - Create a new directory ➜ `mkdir [new-directory-name]`
    - Remove an empty directory ➜ `rmdir [name-of-the-directory-you-want-to-remove]`
    - **Remove command (rm)**
        - Remove a file ➜ `rm [name-of-your-file]`
        - Remove a directory recursively (use with caution) ➜ `rm -rf [name-of-your-`

`directory]`

- **Concatenate command (cat)**
    - View a single file ➜ `cat [name-of-your-file]`
    - View a single file including the line numbers ➜ `cat -n [name-of-your-file]`
    - Copy the content of one file to another file ➜ `cat [filename-whose-contents-is-to-be-copied] > [destination-filename]`
    - Learn more about the cat command ➜ `man cat`
- **Move command (mv)**
    - Move a file ➜ `mv [source-path-of-your-file] [destination-path-for-your-file]`
    - Rename a file ➜ `mv [name-of-your-file] [new name-of-your-file]`

Basic commands

Zoom in

Type `[CTRL] + [+]`

Zoom out

Type `[CTRL] + [-]`

pwd: Print Working Directory command

It prints the working directory path, starting from the root directory.

```
mauro_codes@DESKTOP-HIQ7662:~$ pwd
/home/mauro_codes

mauro_codes@DESKTOP-HIQ7662:~/projects$ pwd
/home/mauro_codes/projects
```

Clear command

Type `clear` or `[CTRL] + [l]` to clear the entire terminal screen and get a clean terminal to keep working.

Alias command

If you usually run a long command regularly and want to save time, you can assign a shorter alias for that command. Type `alias [alias-name]="[command-to-run]"` to assign a new alias:

```
## Running the ls command
mauro_codes@DESKTOP-HIQ7662:~$ ls
projects

## Assign an alias, so we don't need to add the arguments every time we need to list something
mauro_codes@DESKTOP-HIQ7662:~$ alias ls="ls -lah"

## Running ls again (we get the result of `ls -lah`)
mauro_codes@DESKTOP-HIQ7662:~$ ls
total 16K
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 17:41 .
drwxr-xr-x 1 root        root         512 Jan 22 10:38 ..
-rw------- 1 mauro_codes mauro_codes 3.0K Jan 22 23:58 .bash_history
-rw-r--r-- 1 mauro_codes mauro_codes  220 Jan 22 10:38 .bash_logout
-rw-r--r-- 1 mauro_codes mauro_codes 3.7K Jan 22 17:32 .bashrc
-rw-r--r-- 1 mauro_codes mauro_codes  807 Jan 22 10:38 .profile
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 12:55 projects
```

> Note that this alias won't be persisted for future uses. If you want to persist your aliases, add them at the end of your .bashrc file located in your home directory.

Source a file

You can use the `source` command to read and execute the content of a file line by line. Type `source [name-of-the-file-to-read-and-execute]`:

```
## Print the content of the script.txt file (contains two commands)
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ cat script.txt
echo "hello world" ## Print a hello message
cal                ## Print a calendar

## Source the script.txt to run each command inside
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ source script.txt
hello world

    January 2021
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

Change Directory command (cd)

Move to a specific directory

Type `cd [name-of-your-directory]`:

```
## Check current directory
mauro_codes@DESKTOP-HIQ7662:~$ pwd
/home/mauro_codes

## Change directory
mauro_codes@DESKTOP-HIQ7662:~$ cd projects/

## Check new working directory
mauro_codes@DESKTOP-HIQ7662:~/projects$ pwd
/home/mauro_codes/projects
```

Move to the parent directory

Type `cd ..`:

```
## Check current directory
mauro_codes@DESKTOP-HIQ7662:~/projects$ pwd
/home/mauro_codes/projects

## Move to the parent directory
mauro_codes@DESKTOP-HIQ7662:~/projects$ cd ..

## Check new working directory
mauro_codes@DESKTOP-HIQ7662:~$ pwd
/home/mauro_codes
```

Move to the home directory

Type `cd ~` or just `cd` as an alternative

```
## Check current directory
mauro_codes@DESKTOP-HIQ7662:~/projects/awesome-app$ pwd
/home/mauro_codes/projects/awesome-app

## Move to the home directory
mauro_codes@DESKTOP-HIQ7662:~/projects/awesome-app$ cd ~

## Check new working directory
mauro_codes@DESKTOP-HIQ7662:~$ pwd
/home/mauro_codes
```

Move to the last directory you were in

Type `cd -` to navigate to the previous directory you were in

```
## Check the current directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ pwd
/home/mauro_codes/projects/landing-page

## Move to another directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ cd /home/mauro_codes/

## Check the new directory
mauro_codes@DESKTOP-HIQ7662:~$ pwd
/home/mauro_codes

## Go back to the previus directory you were in
mauro_codes@DESKTOP-HIQ7662:~$ cd -
/home/mauro_codes/projects/landing-page
```

List command (ls)

Lists the content of the directory you're currently in.

List all visible files and directories

Type `ls` without any additional argument to get all the files and directories (this command will exclude hidden files like the dotfiles).

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects$ pwd
/home/mauro_codes/projects

## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects$ ls
awesome-app  landing-page  nextjs-tailwindcss-blog-starter  personal-blog
```

List all files and directories

Type `ls -a` to get all the files and directories (including the hidden files)

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects$ pwd
/home/mauro_codes/projects


## List the content for the working directory (including hidden files)
mauro_codes@DESKTOP-HIQ7662:~/projects$ ls -a
.  ..  .config  .configu  awesome-app  landing-page  nextjs-tailwindcss-blog-starter  personal-
```

Long Listed Format

Type `ls -l` to get all the visible files and directories including additional metadata like permissions, owner, size and modified date and time.

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/nextjs-tailwindcss-blog-starter$ pwd
/home/mauro_codes/projects/nextjs-tailwindcss-blog-starter

## List the content for the working directory (using the long listed format)
mauro_codes@DESKTOP-HIQ7662:~/projects/nextjs-tailwindcss-blog-starter$ ls -l
total 140
-rw-r--r-- 1 mauro_codes mauro_codes   4487 Jan 22 12:55 README.md
drwxr-xr-x 1 mauro_codes mauro_codes    512 Jan 22 12:55 components
-rw-r--r-- 1 mauro_codes mauro_codes   1068 Jan 22 12:55 config.ts
drwxr-xr-x 1 mauro_codes mauro_codes    512 Jan 22 12:55 helpers
```

Human Readable Format

Type `ls -lh` to get all the visible files and directories in long-listed format, but with a Human Readable Format (User-friendly file size).

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/nextjs-tailwindcss-blog-starter$ pwd
/home/mauro_codes/projects/nextjs-tailwindcss-blog-starter

## List the content for the working directory (using the long listed format + human readable fo
mauro_codes@DESKTOP-HIQ7662:~/projects/nextjs-tailwindcss-blog-starter$ ls -lh
total 140K
-rw-r--r-- 1 mauro_codes mauro_codes 4.4K Jan 22 12:55 README.md
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 12:55 components
-rw-r--r-- 1 mauro_codes mauro_codes 1.1K Jan 22 12:55 config.ts
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 12:55 helpers
```

Combining arguments

Type `ls -lah` to get all the files and directories (including hidden files) in Human Readable Format.

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/nextjs-tailwindcss-blog-starter$ pwd
/home/mauro_codes/projects/nextjs-tailwindcss-blog-starter

## List the content for the working directory (include hidden files + human readable format)
mauro_codes@DESKTOP-HIQ7662:~/projects/nextjs-tailwindcss-blog-starter$ ls -lah
total 140K
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 13:08 .
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 12:55 ..
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 12:55 .git
-rw-r--r-- 1 mauro_codes mauro_codes  362 Jan 22 12:55 .gitignore
-rw-r--r-- 1 mauro_codes mauro_codes 4.4K Jan 22 12:55 README.md
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 12:55 components
-rw-r--r-- 1 mauro_codes mauro_codes 1.1K Jan 22 12:55 config.ts
drwxr-xr-x 1 mauro_codes mauro_codes  512 Jan 22 12:55 helpers
```

Learn more about the `ls` command

There are dozens of arguments that you can use with the `ls` command. If you want to dig dipper, type `man ls` in your terminal to display the user manual for the `ls` command.

Search

Locate the binary for a program

If you want to locate where the binary (executable) for a specific command or program is located. You can use the `which` command:

```
## Locate binary for the ls command
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ which ls
/usr/bin/ls
## Locate binary for git
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ which git
/usr/bin/git
```

Locate the binary, source, and user manual for a program

You can use the `whereis` command to locate the binary, source, and user manual for a program. You can use the `-b`, `-m`, and `-s` arguments to limit the results to binaries, manual and source, respectively

```
## Locate binary, manual, and source for git
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ whereis git
git: /usr/bin/git /mnt/c/Program Files/Git/cmd/git.exe /usr/share/man/man1/git.1.gz
## Locate only binary and manual for Git, and only the manual for ls command
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ whereis -bm git -m ls
git: /usr/bin/git /mnt/c/Program Files/Git/cmd/git.exe /usr/share/man/man1/git.1.gz
ls: /usr/share/man/man1/ls.1.gz
```

Locate files and directories by name

Type `find [path-to-search] -iname [name-of-the-file-you-want-to-search]` to find any file or directory that contain the given name in their title.

- The path to search is optional. If it is not specified, the `find` command will run on your current working directory (and its descendants)
- The `-iname` argument means that our search will be case insensitive.

```
- If you want to learn more about this command, type `man find` to display the user manual.
## Check current working directory
mauro_codes@DESKTOP-HIQ7662:~/projects$ pwd
/home/mauro_codes/projects

## Find files that contain "posts" on my current working directory and its descendants
mauro_codes@DESKTOP-HIQ7662:~/projects$ find -iname posts
./nextjs-tailwindcss-blog-starter/pages/posts
./nextjs-tailwindcss-blog-starter/posts
## Find files that contain "posts" on a specific directory and its descendants
mauro_codes@DESKTOP-HIQ7662:~/projects$ find ./nextjs-tailwindcss-blog-starter/pages/ -iname po
./nextjs-tailwindcss-blog-starter/pages/posts
```

Get a brief description for a command

If you don't know what a certain command does, Type `whatis [command-name]` like this:

```
## Asking about the cat command
mauro_codes@DESKTOP-HIQ7662:~/projects$ whatis cat
cat (1)              - concatenate files and print on the standard output
## Asking about the find command
mauro_codes@DESKTOP-HIQ7662:~/projects$ whatis find
find (1)             - search for files in a directory hierarchy
```

History

Get previous commands (one by one)

You can access your recent command by pressing the `Up Arrow key` ⬆. This is very useful if you want to repeat your last command. Let's say we move to a specific directory, and then we check our working directory like this:

```
## Move to a specific directory
mauro_codes@DESKTOP-HIQ7662:~$ cd projects/awesome-app/

## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/awesome-app$ pwd
/home/mauro_codes/projects/awesome-app
```

⬆ We'll get the `pwd` command

⬆⬆ We'll get the `cd projects/awesome-app` command

Repeat previous commands (full list)

Type `history` to get a numerated list containing the previous commands you run. Then, type `![number-of-the-command-to-repeat]` to repeat that command

```
## Get the history list
mauro_codes@DESKTOP-HIQ7662:~$ history
    1  ls
    2  clear
    3  pwd
    4  mkdir projects
    5  cd projects

## Run command number 1 (ls)
mauro_codes@DESKTOP-HIQ7662:~$ !1
projects
```

Repeat the last command

Type `!!` (bang-bang command) to repeat the last command. This is especially useful when you forgot to add `sudo` on your last command:

```
## Running update without sudo (Permission denied)
mauro_codes@DESKTOP-HIQ7662:~$ apt update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission den

## Using the bang-bang command to append the last command after sudo
mauro_codes@DESKTOP-HIQ7662:~$ sudo !!
sudo apt update
[sudo] password for mauro_codes:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
...
```

Working with files and directories

Create a new file (without open it)

Type `touch [name-of-your-file]` to create a new file without open it on a text editor. This is useful if you just want to create an empty file but don't need to change it right now.

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ pwd
/home/mauro_codes/projects/landing-page

## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md

## Create an empty js file
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ touch main.js

## List the content for the working directory (including your new file)
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  main.js
```

Create a new file using a text editor

Type `nano [name-of-your-file]` to create a new file and open it using the text editor nano. If you want to learn more about nano, you can Type `man nano` on your terminal to display the nano user manual.

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ pwd
/home/mauro_codes/projects/landing-page

## List the content for the working directory

mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  main.js

mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ nano index.html
```

After running the last command, you'll be able to edit the file using nano:

Copy a file

You can use the `cp` (Copy) command to copy files and directories
Type `cp [source-path-of-your-file] [destination-path-for-your-file]` to copy a file into a new destination.

```
## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index.html  main.js  temp

## Copy the README.md file into the temp directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ cp README.md temp/README.md

## List the content for the working directory and check that your file is still there.
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index.html  main.js  temp

## List the temp directory's content and check if your file was copied.
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls temp/
```

```
README.md  index-copy.html
```

Create a new directory

Type `mkdir [new-directory-name]` to create a new directory in your current working directory

```
## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index-empty-copy.html  index.html  main.js

## Create a new directory called "scripts"
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ mkdir scripts

## List the content to check if our new directory was created
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index-empty-copy.html  index.html  main.js  scripts
```

Remove an empty directory

Type `rmdir [name-of-the-directory-you-want-to-remove]` to remove an empty directory. Please note that this command will only work with empty directories.

```
## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index.html  main.js  temp

## Remove the "temp" empty directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ rmdir temp

## List the content and check that the directory was removed

mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index.html  main.js
```

Remove command (rm)

Remove a file

Type `rm [name-of-your-file]` to remove a file

```
## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page/temp$ ls
README.md  index-copy.html

## Remove the index-copy.html file
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page/temp$ rm index-copy.html

## List the content for the working directory and check that the file was removed
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page/temp$ ls
README.md
```

Remove a directory recursively

Type `rm -rfi [name-of-your-directory]` to recursively remove a directory with all its files and sub-directories.

> **Please be careful!** This is one of the most dangerous commands you can run. If you run `rm -rfi /`, you'll erase your entire root partition. Be sure to specify the path for the directory you want to delete. In this example, In this example, I include the `-i` argument to ask for confirmation.

```
## List the content of the temp folder (It has one file)
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls temp/
total 0
drwxr-xr-x 1 mauro_codes mauro_codes 512 Jan 24 19:45 .
drwxr-xr-x 1 mauro_codes mauro_codes 512 Jan 24 19:44 ..
-rw-r--r-- 1 mauro_codes mauro_codes   8 Jan 24 19:45 file.txt

## Recursively remove the temp folder
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ rm -rf temp/

## Check that the temp folder was removed
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls temp/
ls: cannot access 'temp/': No such file or directory
```

Concatenate command (cat)

You can use the `cat` (concatenate) command to read data from a file and print their content as output

View a single file

Type `cat [name-of-your-file]`:

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ pwd
/home/mauro_codes/projects/landing-page


## Print the content of the index.html file
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ cat index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>My Website</title>
  </head>

  <body>
    <script src="js/main.js"></script>
  </body>
</html>
```

View a single file including the line numbers

Type `cat -n [name-of-your-file]`:

```
## Check the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ pwd
/home/mauro_codes/projects/landing-page

## Print the content of the index.html file
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ cat -n index.html
     1  <!DOCTYPE html>
     2  <html lang="en">
     3    <head>
     4      <meta charset="utf-8" />
     5      <meta http-equiv="x-ua-compatible" content="ie=edge" />
     6      <meta name="viewport" content="width=device-width, initial-scale=1" />
     7
     8      <title>My Website</title>
     9    </head>
    10
    11    <body>
    12      <script src="js/main.js"></script>
    13    </body>
    14  </html>
```

Copy the content of one file to another file

Type `cat [filename-whose-contents-is-to-be-copied] > [destination-filename]`:

```
## Create an empty file called index-empty-copy.html
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ touch index-empty-copy.html

## Copy the content of index.html to index-empty-copy.html
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ cat index.html > index-empty-copy.html

## Print the content of the index-empty-copy.html file
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ cat index-empty-copy.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>My Website</title>
  </head>

  <body>
    <script src="js/main.js"></script>
  </body>
</html>
```

Learn more about the `cat` command

Type `man cat` to display the user manual for the `cat` command

Move command (mv)

You can use the `mv` (move) command for moving and renaming files

Move a file

Type `mv [source-path-of-your-file] [destination-path-for-your-file]` to move a file into a new directory

```
## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index-empty-copy.html  index.html  main.js  temp

## Move the index-empty-copy.html file to the temp directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ mv index-empty-copy.html temp/index-empty-

## List the content again and check that the file is no longer in the current working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls
README.md  index.html  main.js  temp

## List the temp folder and check that the file is now there.
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page$ ls temp/
index-empty-copy.html
```

Rename a file

Type `mv [name-of-your-file] [new name-of-your-file]` to rename a file

```
## List the content for the working directory
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page/temp$ ls
index-empty-copy.html

## Rename the index-empty-copy.html file
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page/temp$ mv index-empty-copy.html index-copy.h

## List the content for the working directory (check if your file's name was updated)
mauro_codes@DESKTOP-HIQ7662:~/projects/landing-page/temp$ ls
index-copy.html
```