command-line-tricks

Contents

Practical Terminal Commands Every Developer Should Know

Level up your command-line skills with these practical terminal tricks. These tips can save you time, fix mistakes quickly, or help debug without extra tools. Whether you're a beginner or a power user, there's something here for everyone.

---

1. `!!` → Repeat the Last Command

Forget to use `sudo`? No need to retype everything.

```
$ apt-get install curl
Permission denied
$ sudo !!
```

`!!` re-runs your last command. Handy for correcting permission issues.

---

2. `!$` → Reuse the Last Argument

Automatically inserts the last argument from your previous command.

```
$ mkdir logs
$ cd !$
```

Here, `!$` becomes `logs`.

---

3. `^old^new` → Fix Typos Quickly

Fixes the first occurrence of a typo and re-runs the command.

```
$ cat fole.txt
$ ^fole^file
$ cat file.txt
```

Great for minor typo corrections.

---

4. `Alt + .` (Mac: `Esc + .`) → Cycle Through Last Arguments

Pulls the last argument from previous commands and cycles through them.

```
$ vim server_config.yml
$ cat <Alt+.>
```

Inserts `server_config.yml`.

---

5. `xargs` → Turn Output into Arguments

Pass a list of items into another command.

```
$ ls *.log | xargs rm
```

Deletes all `.log` files using `rm`.

---

6. `tee` → Save Output While Still Seeing It

Splits output to both screen and file.

```
$ make build | tee build.log
```

Useful for logging while watching build progress.

---

7. `grep -R "pattern" .` → Search Inside Files Recursively

Finds all occurrences of a pattern in the current directory.

```
$ grep -R "TODO" .
```

Ideal for scanning source code or config files.

---

8. `fc` → Edit and Rerun Your Last Command

Opens the last command in your default editor for modification.

```
$ fc
```

Great for editing long or complex commands.

---

9. `!!:n` → Reuse a Specific Argument

Extract a specific argument from the last command.

```
$ echo one two three
$ echo !!:2
two
```

`!!:2` uses the second argument.

---

10. `Ctrl + a` / `Ctrl + e` → Jump to Line Start/End

- `Ctrl + a`: Move to beginning of line.
- `Ctrl + e`: Move to end of line.

Boosts navigation speed.

---

11. `Ctrl + w` / `Ctrl + u` → Delete Words or Lines

- `Ctrl + w`: Delete word before the cursor.
- `Ctrl + u`: Delete entire line before the cursor.

Helpful for cleaning up commands.

---

12. `!!:gs/old/new/` → Global Replace in Last Command

Replaces **all** instances of `old` with `new`.

```
$ echo world world
$ !!:gs/world/universe/
universe universe
```

Perfect for mass substitutions.

---

13. `df -h` / `du -sh *` → Check Disk Usage

Find out what's eating your storage:

```
$ df -h       # Disk free space (human readable)
$ du -sh *    # Size of all files/folders in current dir
```

---

14. `lsof -i :<port>` → See What's Using a Port

Find the process locking a port.

```
$ lsof -i :8080
```

Now you know what to kill or restart.

---

15. `nc -zv <host> <port>` → Test if a Port is Open

Quick connectivity check:

```
$ nc -zv google.com 443
```

Useful for debugging networking issues.

---

16. `cd -` → Jump Back to Previous Directory

Like an “undo” for `cd`.

```
$ cd projectA
$ cd ../projectB
$ cd -
```

Jumps you back to `projectA`.

---

17. `Ctrl + l` → Clear the Screen Instantly

Forget typing `clear`. Just use:

```
Ctrl + l
```

Bonus Tips

`history` → View Command History

```
$ history
```

See all past commands. Use `!123` to re-run command #123.

`pushd` / `popd` → Stack Navigation for Directories

```
$ pushd folderA
$ pushd folderB
$ popd
```

Let you switch between directories using a stack.

`watch` → Repeatedly Run a Command

```
$ watch -n 2 ls -l
```

Runs `ls -l` every 2 seconds. Great for monitoring file changes.