

WSL / Linux Cheat Sheet

Development edition for Astro, Node.js, Git, GitHub and Cloudflare Pages

Safety rule: run `pwd` and `ls -la` before destructive commands. Treat `rm -rf` as irreversible.

1. Navigation and inspection

Goal	Command	Notes
Show current directory	<code>pwd</code>	Confirm location before destructive operations.
Detailed listing	<code>ls -la</code>	Includes hidden files.
Human-readable sizes	<code>ls -lh</code>	Useful for downloads and build output.
Enter a directory	<code>cd ~/Projects</code>	
Go up one level	<code>cd ..</code>	
Return home	<code>cd ~</code>	
Previous directory	<code>cd -</code>	Switch between two locations.
Project tree	<code>tree -L 2</code>	Install with <code>sudo apt install tree</code> .
Find package.json	<code>find . -name "package.json"</code>	

2. Files, folders and cleanup

Goal	Command	Notes
Create nested folders	<code>mkdir -p src/content/articles</code>	
Create a file	<code>touch notes.md</code>	
Copy a file	<code>cp source.txt copy.txt</code>	
Copy a directory	<code>cp -r source/ backup/</code>	
Move or rename	<code>mv old-name new-name</code>	
Delete a file	<code>rm file.txt</code>	Irreversible.
Delete a directory	<code>rm -r folder</code>	
Clean build files	<code>rm -rf node_modules dist .astro</code>	Check <code>pwd</code> first.
Remove Zone.Identifier	<code>find . -name "*Zone.Identifier*" -delete</code>	

3. ZIP and downloads

Goal	Command	Notes
Inspect ZIP	<code>unzip -l package.zip head -30</code>	
Extract ZIP	<code>unzip package.zip</code>	
Extract to destination	<code>unzip package.zip -d target-folder</code>	
Create ZIP	<code>zip -r release.zip project-folder/</code>	
Move Windows download	<code>mv /mnt/c/Users/NAME/Downloads/file.zip ~/</code>	Extract in WSL.

4. WSL and Windows

Goal	Command	Notes
List distributions	<code>wsl -l -v</code>	Run in PowerShell.
Show status	<code>wsl --status</code>	Run in PowerShell.
Restart WSL	<code>wsl --shutdown</code>	Run in PowerShell.
Start Ubuntu	<code>wsl -d Ubuntu</code>	Run in PowerShell.
Open Explorer	<code>explorer.exe .</code>	Inside WSL.
Open VS Code	<code>code .</code>	Inside project folder.
Show Linux user	<code>whoami</code>	
Show uid and groups	<code>id</code>	

5. Node.js and npm

Goal	Command	Notes
Node version	<code>node -v</code>	Need $\geq 22.12.0$.
npm version	<code>npm -v</code>	
Locate Node	<code>which node</code>	Must be a Linux path.
Locate npm	<code>which npm</code>	Must not point to <code>/mnt/c</code> .
Install Node	<code>nvm install 22.12.0</code>	
Use Node	<code>nvm use 22.12.0</code>	
Set default	<code>nvm alias default 22.12.0</code>	
Install dependencies	<code>npm install</code>	
Clean CI install	<code>npm ci</code>	Requires <code>package-lock.json</code> .
Check script shell	<code>npm config get script-shell</code>	Normally null.

6. Astro

Goal	Command	Notes
Development server	<code>npm run dev</code>	
Network-accessible dev server	<code>npm run dev -- --host</code>	
Production build	<code>npm run build</code>	Creates <code>dist/</code> .
Preview build	<code>npm run preview</code>	
Project check	<code>npm run check</code>	
Reset build state	<code>rm -rf dist .astro</code>	

7. Git fundamentals

Goal	Command	Notes
Initialise	<code>git init</code>	
Main branch	<code>git branch -M main</code>	
Status	<code>git status</code>	Run frequently.
Unstaged changes	<code>git diff</code>	
Staged changes	<code>git diff --staged</code>	
Stage all	<code>git add .</code>	Review before commit.

Goal	Command	Notes
Commit	<code>git commit -m "style: refine homepage"</code>	
Compact history	<code>git log --oneline --decorate --graph -15</code>	
Discard file edits	<code>git restore path/to/file</code>	
Unstage file	<code>git restore --staged path/to/file</code>	

8. Branches and remote

Goal	Command	Notes
Create branch	<code>git switch -c feature/homepage</code>	
Switch main	<code>git switch main</code>	
Update safely	<code>git pull --ff-only</code>	
Push new branch	<code>git push -u origin feature/homepage</code>	
Delete local branch	<code>git branch -d feature/homepage</code>	
Compare to main	<code>git diff main...HEAD</code>	
Add GitHub remote	<code>git remote add origin https://github.com/USER/REPO.git</code>	
Show remote	<code>git remote -v</code>	
First push	<code>git push -u origin main</code>	

9. Tags and releases

Goal	Command	Notes
Create release tag	<code>git tag -a v0.1.0 -m "Initial foundation"</code>	
Push tag	<code>git push origin v0.1.0</code>	
List tags	<code>git tag --list</code>	
Delete local tag	<code>git tag -d v0.1.0</code>	Use only to correct a mistake.

10. Cloudflare Pages

Goal	Command	Notes
Local preflight	<code>node -v && npm ci && npm run build</code>	
Build command	<code>npm run build</code>	Cloudflare setting.
Output directory	<code>dist</code>	Cloudflare setting.
Node variable	<code>NODE_VERSION=22.12.0</code>	Cloudflare environment variable.

11. Permissions and recovery

Goal	Command	Notes
Ownership	<code>ls -la</code>	
Fix project ownership	<code>sudo chown -R "\$USER": "\$USER" project-folder</code>	Only for root-owned files.
Make executable	<code>chmod +x script.sh</code>	
Disk usage	<code>df -h</code>	
Folder sizes	<code>du -sh *</code>	

Goal	Command	Notes
Port 4321 process	<code>lsof -i :4321</code>	
Stop process	<code>kill PID</code>	

12. Daily workflow for Designing Clarity

Use one short-lived branch per feature, case study or content change.

```
cd ~/Projects/huanxie-designing-clarity

git switch main
git pull --ff-only
git switch -c content/new-article

npm install
npm run dev

git status
git diff

npm run check
npm run build

git add .
git commit -m "content: add article on compliance by design"
git push -u origin content/new-article
```

Cloudflare deployment flow

```
feature branch -> Pull Request -> main -> Cloudflare Pages automatic build
```

Failure-check order

- Confirm Node version.
- Run npm ci locally.
- Run npm run build locally.
- Check missing environment variables.
- Check filename case and import paths.
- Read the Cloudflare build log from the first error upward.

Destructive-command checklist

Before rm -rf, git restore, deleting a tag or rewriting history:

```
pwd
ls -la
git status
```

- Am I in the correct directory?
- Is the content committed or backed up?
- Am I deleting generated output or source files?
- Is there a safer command?

Five commands to remember first

```
pwd
ls -la
git status
npm run dev
npm run build
```