

# Student Command Handout

Quick reference for the Playwright + TypeScript course. Commands are shown for Windows (PowerShell) and macOS/Linux (bash/zsh).

## 1) Terminal, Navigation, Files

Task	Windows (PowerShell)	macOS/Linux (bash/zsh)	What it's for
Show current folder	Get-Location	pwd	Confirm where you are
List files/folders	dir	ls -la	See project files
Change folder	cd .\path\to\proj	cd ./path/to/proj	Move into project
Clear screen	cls	clear	Clean terminal view
Create folder	mkdir training	mkdir training	Create a directory
Create file	New-Item README.md -Type File	touch README.md	Create empty file
View file contents	Get-Content package.json	cat package.json	Inspect configs/scripts
Copy file	Copy-Item a.txt b.txt	cp a.txt b.txt	Duplicate a file
Move/rename	Move-Item old.ts new.ts	mv old.ts new.ts	Rename or move
Delete file/folder	Remove-Item tmp -Recurse -Force	rm -rf tmp	Remove generated files
Zip / Unzip	Compress-Archive .\tests tests.zip Expand-Archive .\tests.zip .tests	zip -r tests.zip ./tests unzip tests.zip -d ./tests	Share labs / extract labs
Stop running command	Ctrl+C	Ctrl+C	Cancel a run (safe)

## 2) Search, Environment Variables, Ports

Task	Windows (PowerShell)	macOS/Linux (bash/zsh)	What it's for
Search text in files	Select-String -Pattern "login" -Path .\tests\*.ts	grep -R "login" ./tests	Find usages fast
Find files by name	Get-ChildItem -Recurse -Filter "*.spec.ts"	find . -name "*.spec.ts"	Locate specs/helpers
Read env var	\$env:BASE_URL	echo \$BASE_URL	Check configuration
Set env var (session)	\$env:BASE_URL= "https://staging"	export BASE_URL= "https://staging"	Set config for this terminal
Env var for one run	\$env:BASE_URL= "https://staging"; npx playwright test	BASE_URL="https://staging" npx playwright test	Temporary override
Who uses port 3000	netstat -ano   findstr :3000	lsof -i :3000	Fix 'port in use'

Note: In PowerShell, run commands on separate lines if '&&' is not supported in your version.

### 3) Git Daily Commands

Task	Command	Example / Notes	Deliverable
Clone repo	git clone	Run once per project	Local repo created
Check status	git status	See modified/untracked files	Know what changed
Create branch	git checkout -b feature/day2-lab	Use a new branch per lab	Safe isolated work
Stage + commit	git add . git commit -m "Day 2 lab"	Commit small & often	Changes saved
Pull / push	git pull git push -u origin feature/day2-lab	Sync with remote	Work backed up

### 4) Node/NPM/NPX Essentials

Task	Windows + macOS/Linux	Example	What it's for
Check versions	node -v   npm -v	node -v	Verify toolchain
Install deps	npm install	npm i	Install project packages
Install dev dep	npm i -D @playwright/test	adds to package.json	Add Playwright test runner
Run npm script	npm run test	script from package.json	Standardized run command
Use npx	npx playwright --version	npx runs local binaries	Avoid global installs

### 5) Playwright Commands You Will Use Every Day

Task	Command	Example	What it's for
Initialize project	npm init playwright@latest	Choose TypeScript	Scaffold test project
Install browsers	npx playwright install	Run once per machine	Download browsers
Run all tests	npx playwright test		Execute suite
Run one file	npx playwright test tests/login.spec.ts		Target a spec
Run by test name	npx playwright test -g "should login"		Run matching tests
Headed mode	npx playwright test --headed		Watch browser
Debug mode	npx playwright test --debug		Inspector + step through
Open report	npx playwright show-report		View HTML report

*Tip:* Keep tests stable with `getByRole/getByLabel/getById` and avoid hard waits like `waitForTimeout` unless absolutely necessary.