

# Git Commands for Playwright TypeScript Course

A quick, end-to-end reference you will use daily during this training.

## 1) Daily Checklist

Before you start (once)	Every training day	Collaboration / PR	Fixing mistakes
<ul style="list-style-type: none"> <li>• git --version</li> <li>• git config user.name / user.email</li> <li>• (Recommended) SSH key</li> <li>• git clone</li> </ul>	<ul style="list-style-type: none"> <li>• git checkout main</li> <li>• git pull --rebase</li> <li>• git checkout -b</li> <li>• git status</li> <li>• git add .</li> <li>• git commit -m "..."</li> <li>• git push -u origin</li> </ul>	<ul style="list-style-type: none"> <li>• git fetch origin</li> <li>• git rebase origin/main</li> <li>• resolve conflicts -&gt; add -&gt; rebase --continue</li> <li>• open PR in Git UI</li> </ul>	<ul style="list-style-type: none"> <li>• git restore --staged</li> <li>• git restore</li> <li>• git commit --amend</li> <li>• git log --oneline</li> </ul>

## 2) Branch Naming (Recommended)

Use clear, searchable branch names (pick one style):

- day02-locators-and-fixtures
- feature/login-fixture
- bugfix/flaky-selector-homepage
- perf/jmeter-baseline-notes

## 3) Copy/Paste Daily Routine

```
# Start your day
git checkout main
git pull --rebase
git checkout -b dayXX-topic-name

# Do your work (edit files) ...

git status
git add .
git commit -m "Day XX: <short description>"
git push -u origin dayXX-topic-name
```

## 4) Update Your Branch from main (Before You Push / Before PR)

```
git fetch origin
git rebase origin/main    # or: origin/master

# If conflicts happen:
git status
# open conflicted file(s), fix markers, save
git add <fixed_file>
git rebase --continue
```

## 5) Conflict Markers (What You Will See)

Remove the markers and keep the correct code:

```
<<<<< HEAD
your changes
=====
incoming changes from main
>>>>> origin/main
```

## 6) Undo / Fix Mistakes (Most Common Cases)

Goal	Command	Example
Unstage a file (keep edits)	git restore --staged <file>	git restore --staged tests/login.spec.ts
Discard local edits (danger)	git restore <file>	git restore playwright.config.ts
Fix last commit (message/files)	git commit --amend	git add . then: git commit --amend -m "..."
See compact history	git log --oneline --graph --decorate --all	git log --oneline --graph --decorate --all

## 7) Housekeeping (After PR Merge)

```
# Delete local branch
git branch -d dayXX-topic-name

# Delete remote branch
git push origin --delete dayXX-topic-name

# Optional: tag a milestone
git tag -a v1.0 -m "End of Playwright Fundamentals"
git push origin v1.0
```

## 8) Daily Submission Standard (Recommended)

- Create a branch for the day/topic.
- Commit with a clear message.
- Push so instructor can review.
- Include run evidence when requested (Playwright HTML report, trace, screenshot, or console output).