Git quick reference for learners

A one-page reference of the Git and Composer commands a learner uses with Exponential - setting up clones, checking state, branching and committing, reading history, finding a breaking change, contributing through a...

Source file: `doc/eng-us/help/students/git-quick-reference-for-learners.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/students/git-quick-reference-for-learners.md

Every command on this page comes from Develop with Git or `CONTRIBUTING.md` unless another source is named. Print it or keep it next to your terminal. The explanations are in Using Git as a student.

Contents

1. Setting up
2. Where am I?
3. Branching and committing
4. Reading history
5. Finding a breaking change
6. Contributing through a fork
7. Staying up to date
8. When Composer refuses

Setting up

Command	Does
`composer config --global 'preferred-install.se7enxweb/*' source`	Install every se7enxweb/* package as a Git clone
`git clone https://github.com/se7enxweb/exponential.git exponential`	Get the kernel with its full history
`git switch -c my-site`	Start your own branch on `main`
`git switch -c my-site v6.0.14`	Start your own branch on a release tag instead
`git tag -l 'v*' --sort=version:refname`	List the release tags in version order
`composer install`	Clone the extensions into `extension/<name>`

Where am I?

Command	Does
`git status`	Changed and untracked files

Command	Does
`git diff`	The changes, line by line
`composer status -v`	Packages with uncommitted changes, unpushed commits or another version than the lock file
`ls -d extension/*/.git | wc -l`	Count the extensions that are clones
`git -C extension/<name> status`	The state of one extension ("HEAD detached at v..." is normal)

Branching and committing

Command	Does
`git switch -c <branch>`	A new branch from where you are
`git add <file>`	Stage one file (by name, not everything at once)
`git commit -m "Fixed: <what>"`	Commit with a subject in the project's style
`git log --oneline -5`	Your last five commits

Subjects start with `Added:` , `Updated:` , `Fixed:` , `Removed:` or `Renamed:` .

Reading history

Command	Does
`git log --oneline -10 -- <file>`	The last ten changes to one file
`git blame <file>`	Who last changed each line
`git log -S "<text>" --oneline`	Every commit that added or removed a piece of text
`git -C extension/<name> log --oneline -3`	The newest commits of one extension

Finding a breaking change

Shell

```
git bisect start
git bisect bad
git bisect good v6.0.13
# test, then answer: git bisect good   or   git bisect bad
git bisect reset
```

Contributing through a fork

Shell

```
cd extension/<name>
git remote add fork https://github.com/<you>/<name>.git
git switch -c <fix-branch>
git commit -am "Fixed: <what>"
git push -u fork <fix-branch>
# then open the pull request from the link GitHub prints, or: gh pr create
```

While the pull request is open, when reviews ask for it:

Shell

```
git fetch origin
git rebase -i origin/main
git push --force-with-lease     # only to your own pull request branch
```

For the kernel: `make hooks` once, and `make manifest-fix` to keep `share/filelist.md5` current.

Staying up to date

Command	Does
`git fetch origin --tags` then `git merge origin/main`	Bring the kernel up to date on your branch
`composer update`	Then the extensions to match
`composer update se7enxweb/<package>`	One extension only

Git first, then Composer. The full routine: Update with Git.

When Composer refuses

Composer says	Do
`... has uncommitted changes.`	Commit on a branch, or stash, update, and restore
`... has unpushed changes on the current branch`	Push the branch to your fork, then update
`You have version variations ...`	Expected while you work on a fix; `composer update <package>` returns to the tag
A package is still an archive	`composer reinstall <package>`; never `git clone` by hand into it

Source: When Composer refuses.

Related pages

- Using Git as a student
- Develop with Git
- Update with Git
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.