

First Time Setup

Name `git config --global user.name "<name>"`

Email `git config --global user.email "<email>"`

Setup your first repository

1. Create a folder `mkdir <my code> && cd <my code>`

2. Initialize `git init -b main`

3. Add files to track `vi readme.md`

4. Track files `git add .`

5. Initial Commit `git commit -am "Initial Commit "`

Create a Bare Repository (to use as a remote)

1. `mkdir <my repository> && cd <my repository>`

2. `git init --bare`

Clone An Existing Repository

`git clone <url of file path>`

Getting Around

I need help. `git help`

What is going on in my repo? `git status`

What happened in the past? `git log`

Can I see a history graph? `git log --oneline --graph`

What changed? `git diff`

How to configure things? `git config`

How to manage branches? `git branch`

How to manage remotes? `git remote`

Resources

<https://ohshitgit.com>

<https://git-scm.com/book/en/v2>

<https://blog.sasworkshops.com/the-easiest-way-to-setup-lvcompare-and-lvmerge/>

<https://sasworkshops.thinkific.com/courses/using-git-effectively>

<https://blog.sasworkshops.com/tag/scc/>

<https://forums.ni.com/t5/Git-User-Group/bd-p/grp-2368>

Creating a Commit

1. Add all changes `git add .`

2. Make a commit `git commit -am "Commit Message"`

Seeing What Changed

List changed files `git diff --name-only`

Open LVCompare `git difftool`

What changed in a commit? `git show <commit>`

Which commits changed a file? `git log -- <file>`

Undoing Changes

Throw away all `git restore .`

Throw away one file `git restore <file>`

Undo add file `git restore --staged <file>`

Go back to commit `git reset --hard <commit>`

Reset file to commit `git checkout <commit> -- <file>`

Undo published commit `git revert <commit>`

Branches

New branch `git switch -C <new branch>`

Change branch `git switch <other branch>`

Combine changes `git merge <other branch>`

Syncing With Remotes

Push changes up `git push`

Pull changes down `git pull`

Push a new branch `git push -u origin <my branch>`

LabVIEW Specific Tips

Ignore *.aliases, *.lvps, *.UserState files using .gitignore

Separate Compiled Code.

Merge conflicts are painful. Avoid changing the same file in parallel.

*.lvclass, *.lvlib and *.lvproj can contain binary data.

Close your lvproj before switching branches or reverting changes on old versions of LV - pre 2026

Consider multiple lvproj to avoid conflicts.



By **sasworkshops**

Published 7th November, 2025.
Last updated 27th June, 2026.
Page 1 of 1.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>