

First Time Setup

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

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

Setup your first repository

1. Create a folder `mkdir <my cod e> && cd <my cod e>`
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 rep o.g it> && cd <my rep o.g it>`
2. `git init --bare`

Clone An Existing Repository

`git clone <ur l_o r_f ile _pa th>`

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 Messag e"`

Seeing What Changed

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

Open LVCompare `git difftool`

What changed in a commit? `git show <co mmi t>`

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

Undoing Changes

Throw away all `git restore .`

Throw away one file `git restore <fi le>`

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

Go back to commit `git reset --hard <co mmi t>`

Reset file to commit `git checkout <co mmi t> -- <fi -le>`

Undo published commit `git revert <co mmi t>`

Branches

New branch `git switch -C <ne w_b ran ch>`

Change branch `git switch <ot her _br anc h>`

Combine changes `git merge <ot her _br anc h>`



By **sasworkshops**

cheatography.com/sasworkshops/

Published 7th November, 2025.

Last updated 16th September, 2026.

Page 1 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Syncing With Remotes

Push changes up	<code>git push</code>
Pull changes down	<code>git pull</code>
Push a new branch	<code>git push -u origin <my branch></code>

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. Beware of merging them. Consider using a .gitattributes file to mark them as binary

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

Consider multiple lvproj to avoid conflicts.



By **sasworkshops**

cheatography.com/sasworkshops/

Published 7th November, 2025.

Last updated 16th September, 2026.

Page 2 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>