

# GIT

---

## Getting Started

| Task                   | Command         |
|------------------------|-----------------|
| Start a new repo       | git init        |
| Clone an existing repo | git clone <url> |

## Configure Git

| Task                                                  | Command                            |
|-------------------------------------------------------|------------------------------------|
| Set a config option                                   | git config user.name 'Your Name'   |
| Set a config option                                   | git config user.email 'Your email' |
| Set option system wide in /etc/gitconfig              | git config --system ...            |
| Set option globally in ~/.gitconfig                   | git config --global ...            |
| Set option locally in the repository .git/config file | git config --local ...             |
| Add an alias                                          | git config alias.st status         |
| See all possible config options                       | man git-config                     |

## Prepare to Commit

| Task                                                | Command                |
|-----------------------------------------------------|------------------------|
| Add untracked file or unstaged changes              | git add <file>         |
| Add all untracked files and unstaged changes        | git add .              |
| Choose which parts of a file to stage               | git add -p             |
| Move file                                           | git mv <old> <new>     |
| Delete file                                         | git rm <file>          |
| Tell Git to forget about a file without deleting it | git rm --cached <file> |
| Unstage one file                                    | git reset <file>       |
| Unstage everything                                  | git reset              |
| Check what you added                                | git status             |
| Check what you added, short format                  | git status -s          |

## Make Commits

| Task                                                  | Command                  |
|-------------------------------------------------------|--------------------------|
| Make a commit (and open text editor to write message) | git commit               |
| Make a commit                                         | git commit -m 'message'  |
| Commit all unstaged changes                           | git commit -am 'message' |

## Branches

| Task                                        | Command                          |
|---------------------------------------------|----------------------------------|
| List branches                               | git branch                       |
| List branches by most recently committed to | git branch --sort=-committerdate |

| Task                  | Command                                                                   |
|-----------------------|---------------------------------------------------------------------------|
| Switch branches       | git switch <name> OR<br>git checkout <name>                               |
| Create a branch       | git branch <name> OR<br>git switch -c <name> OR<br>git checkout -b <name> |
| Delete a branch       | git branch -d <name>                                                      |
| Force delete a branch | git branch -D <name>                                                      |

## Diff Staged/Unstaged Changes

| Task                                 | Command          |
|--------------------------------------|------------------|
| Diff all staged and unstaged changes | git diff HEAD    |
| Diff just staged changes             | git diff -staged |
| Diff just unstaged changes           | git diff         |

## Diff Commits

| Task                                      | Command                                            |
|-------------------------------------------|----------------------------------------------------|
| Show diff between a commit and its parent | git show <commit>                                  |
| Diff two commits                          | git diff <commit> <commit>                         |
| Diff one file since a commit              | git diff <commit> <file>                           |
| Show a summary of a diff                  | git diff <commit> -stat<br>git show <commit> -stat |

## Discard Your Changes

| Task                                               | Command                                                             |
|----------------------------------------------------|---------------------------------------------------------------------|
| Delete unstaged changes to one file                | git restore <file> OR<br>git checkout <file>                        |
| Delete all staged and unstaged changes to one file | git restore -staged -worktree <file> OR<br>git checkout HEAD <file> |
| Delete all staged and unstaged changes             | git reset -hard                                                     |
| Delete untracked files                             | git clean                                                           |
| 'Stash' all staged and unstaged changes            | git stash                                                           |

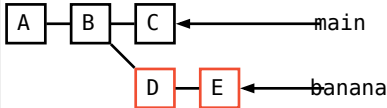
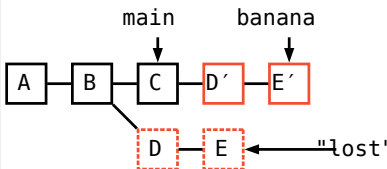
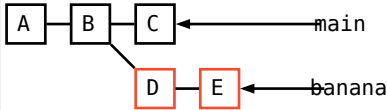
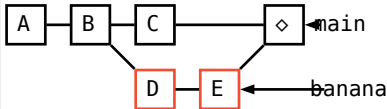
## Edit History

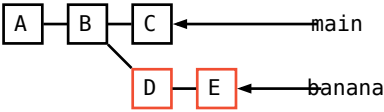
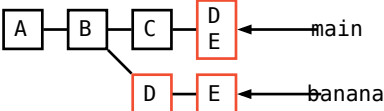
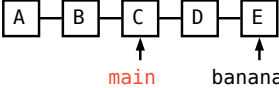
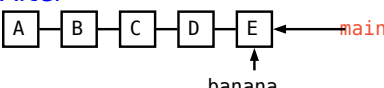
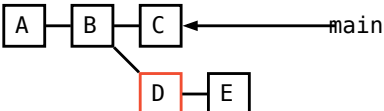
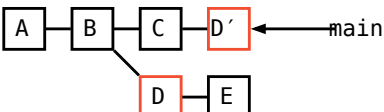
| Task                                                                 | Command                                                                                                              |
|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| "Undo" the most recent commit (keep your working directory the same) | git reset HEAD^                                                                                                      |
| Squash the last 5 commits into one                                   | git rebase -i HEAD~6<br>Then change "pick" to "fixup" for any commit you want to combine with the previous one       |
| Undo a failed rebase                                                 | git reflog BRANCHNAME<br>Then manually find the right commit ID in the reflog, then run:<br>git reset -hard <commit> |
| Change a commit message (or add a file you forgot)                   | git commit -amend                                                                                                    |

## Code Archaeology

| Task                                                                    | Command                                                                          |
|-------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Show the files under control                                            | git ls-files                                                                     |
| Look at a branch's history                                              | git log main<br>git log -all -graph<br>git log -graph <main><br>git log -oneline |
| Show every commit that modified a file                                  | git log <file>                                                                   |
| Show every commit that modified a file, including before it was renamed | git log -follow <file>                                                           |
| Find every commit that added or removed some text                       | git log -G banana                                                                |
| Show who last changed each line of a file                               | git blame <file>                                                                 |

## Combine Diverged Branches

| Task                | Command                                                                                                                                                                                                                                    |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Abort a merge       | git merge -abort                                                                                                                                                                                                                           |
| Combine with Rebase | git switch banana<br>git rebase main<br><br>Before<br><br>After<br> |
| Combine with Merge  | git switch main<br>git merge banana<br><br>Before<br><br>After<br> |

| Task                                                                     | Command                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Combine with Square Merge                                                | <pre>git switch main git merge -squash banana git commit</pre> <p>Before</p>  <p>After</p>  |
| Bring a branch up to date with another branch (aka "fast-forward merge") | <pre>git switch main git merge banana git commit</pre> <p>Before</p>  <p>After</p>         |
| Copy one commit onto the current branch                                  | <pre>git cherry-pick &lt;commit&gt;</pre> <p>Before</p>  <p>After</p>                   |

## Restore an Old File

| Task                                          | Command                                                                                                |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------|
| Get the version of a file from another commit | <pre>git checkout &lt;commit&gt; &lt;file&gt; OR git restore &lt;file&gt; -source &lt;commit&gt;</pre> |

## Remote Repositories

| Task                                                    | Command                                              |
|---------------------------------------------------------|------------------------------------------------------|
| Add a Remote                                            | <code>git remote add &lt;name&gt; &lt;url&gt;</code> |
| Push the main branch to the remote origin               | <code>git push origin main</code>                    |
| Push the current branch to its remote "tracking branch" | <code>git push</code>                                |
| Push a branch that you've never pushed before           | <code>git push -u origin &lt;name&gt;</code>         |
| Force push                                              | <code>git push -force-with-lease</code>              |

| Task                                                       | Command                             |
|------------------------------------------------------------|-------------------------------------|
| Push tags                                                  | git push -tags                      |
| Pull changes (but don't change any of your local branches) | git fetch origin main               |
| Pull changes and then rebase your current branch           | git pull -rebase                    |
| Pull changes and then merge them into your current branch  | git pull origin main OR<br>git pull |

## Important Files

| Item                    | Location     |
|-------------------------|--------------|
| Local git config        | .git/config  |
| Global git config       | ~/.gitconfig |
| List of files to ignore | .gitignore   |

## Merge Conflicts

Common merge error:

```
Auto-merging test.c
CONFLICT (content): Merge conflict in test.c
Automatic merge failed; fix conflicts and then commit the result.
```

Run command:

```
git log --merge --oneline
-----
1d91b52 (bugfix1) 2e bugfix
91d17b9 (HEAD -> master) New feature in Master
578b59d bugfix in bugfix branch
```

```
git status
-----
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   test.c
Untracked files:
  (use "git add <file>..." to include in what will be committed)
test_BACKUP_29859.c
test_BASE_29859.c
test_LOCAL_29859.c
test_REMOTE_29859.c
```

Edit the test.c file to resolve the conflict:

```
nano test.c
```

```
-----  
  
first version test.c file  
<<<<<<< HEAD  
New Feature added in Master branch  
=====  
Bugfix1 in bugfix branch  
  
Bugfix2 in bugfix branch  
>>>>>> bugfix1
```

## Links

- [Cheat Sheet](#)

From:  
<https://wiki.oscardegroot.nl/> - **HomeWiki**

Permanent link:  
<https://wiki.oscardegroot.nl/doku.php?id=computing:git&rev=1777302117>

Last update: **2026/04/27 15:01**

