



Scan me
to watch these
commands animate



Initial Commit
made with git-sim on initialcommit.com

The Essential Git Cheat Sheet

Page 1 of 3

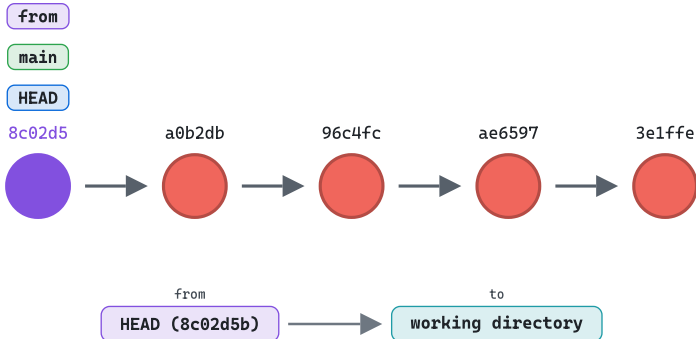
The Essential Git Cheat Sheet

See unstaged changes

● SAFE

```
git diff
```

Shows the lines you've changed but haven't staged yet.



Unstaged changes

2 file(s), +2 -2

M	app.py	+1 -1	■ ■ ■ ■ ■
M	styles.css	+1 -1	■ ■ ■ ■ ■

See what's changed

● SAFE

```
git status
```

Lists what's staged for the next commit, what's changed but not staged, and files Git isn't tracking yet.

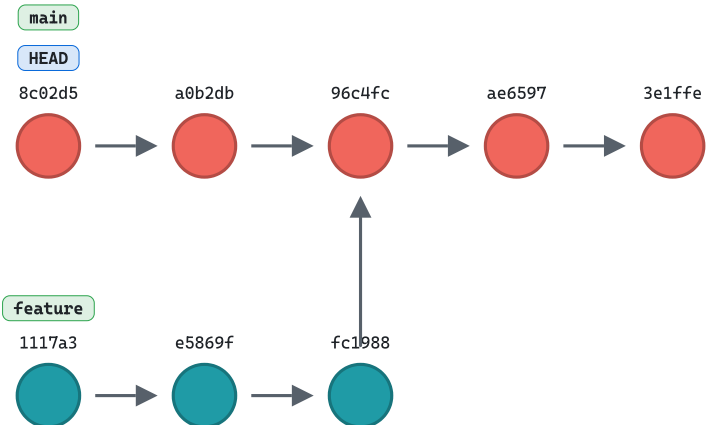
Untracked files	Modified files	Staged files
notes.txt	app.py	README.md

See every branch as a graph

● SAFE

```
git log --oneline --graph --all
```

Draws the history of every branch as a text graph, one line per commit, with the branch labels on it.

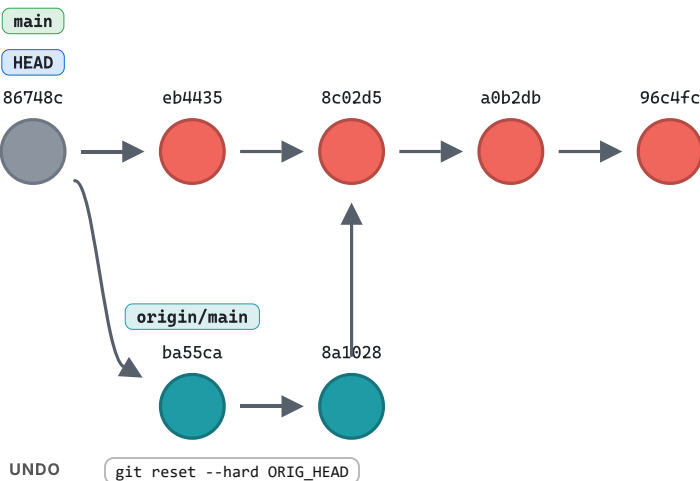


Pull and merge remote changes

● SAFE

```
git pull
```

Fetches, then merges the remote branch into yours. Makes a merge commit when both sides have new commits.

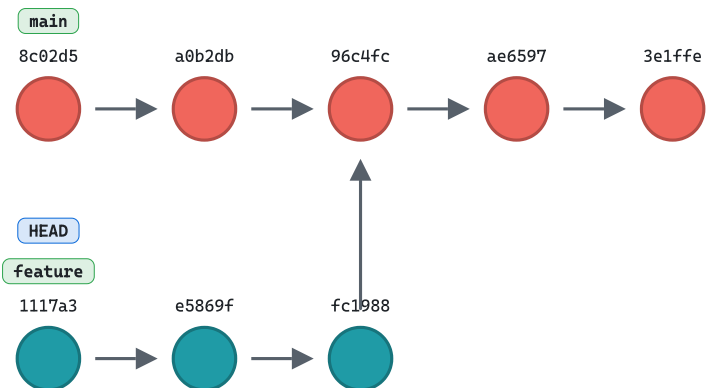


Switch branches

● SAFE

```
git switch <branch>
```

Moves HEAD to the branch and updates your files to match its latest commit.





Scan me
to watch these
commands animate



Initial Commit

made with git-sim on initialcommit.com

The Essential Git Cheat Sheet

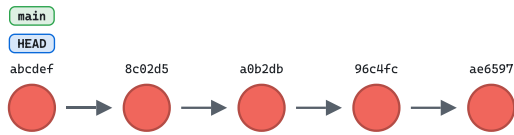
Page 2 of 3

Commit your staged changes

SAFE

```
git commit -m "<message>"
```

Saves everything in the staging area as a new commit, and moves the current branch to it.



Working directory	Staged files	New commit
app.py	README.md	README.md

UNDO `git reset --soft HEAD~1`

Copy a repository

SAFE

```
git clone <url>
```

Downloads a repository with its whole history into a new folder, and remembers where it came from as `origin`.

Successfully cloned from `../your_project.git` into `your_project_copy`
Cloned repo log:

origin/main

main

HEAD

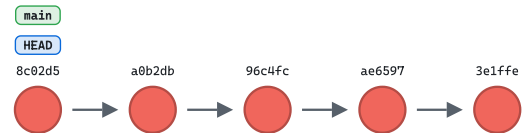


Apply and remove the latest stash

SAFE

```
git stash pop
```

Applies the newest entry to your files and removes it from the stash.



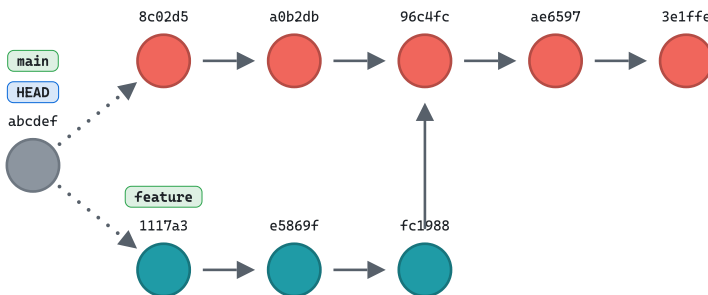
Stashed changes	Working directory	Staging area
app.py	app.py	

Merge a branch into yours

SAFE

```
git merge <branch>
```

Brings the branch's commits into yours with a merge commit that has two parents.



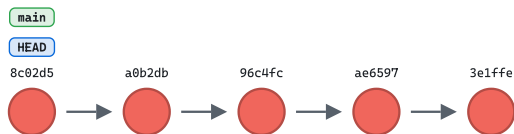
UNDO `git reset --hard ORIG_HEAD`

Stash your changes

SAFE

```
git stash
```

Saves your staged and unstaged changes on the stash and resets the working directory to the last commit.



Stashed changes	Working directory	Staging area
app.py	app.py	
README.md		README.md

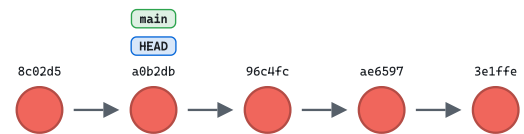
UNDO `git stash pop`

Undo a commit, keep its changes staged

CAUTION

```
git reset --soft HEAD~1
```

Moves the branch back one commit. The commit's changes wait in the staging area, ready to commit again.



Modified files	Staged files	Undone commits
	requirements.txt (new file)	requirements.txt

UNDO `git reset --soft ORIG_HEAD`



Scan me
to watch these
commands animate



Initial Commit

made with git-sim on initialcommit.com

The Essential Git Cheat Sheet

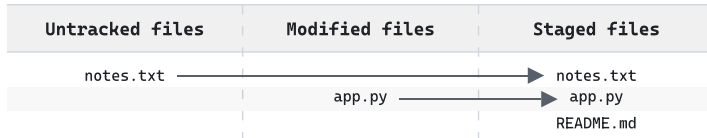
Page 3 of 3

Stage files

● SAFE

```
git add <file>
```

Puts the file's current changes in the staging area, ready for the next commit.

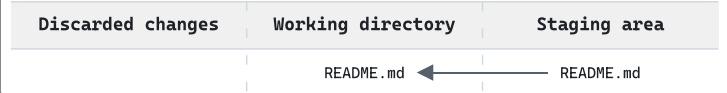


Unstage a file

● SAFE

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

Takes the file out of the next commit. Your edits stay in the file.

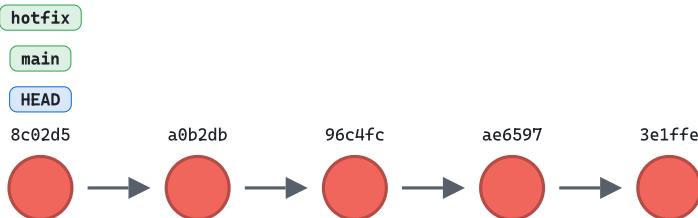


Create a branch and switch to it

● SAFE

```
git switch -c <name>
```

Creates the branch on the current commit and switches to it in one step.

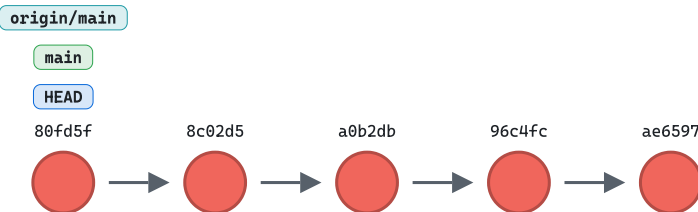


Push your commits

● SAFE

```
git push
```

Sends your branch's new commits to the remote branch it tracks.



Discard changes to a file

● DESTRUCTIVE

```
git restore <file>
```

Puts the file back the way it was at the last commit. `git restore .` does every file.



UNDO No way back

WATCH The edits were never committed, so Git can't bring them back. Run `git stash` first if you're not sure.