

Git cheat sheet

Git commands for everyday work and emergencies – searchable and copyable in one click.
With warnings on commands that discard work for good.

daniele.michelin //
cheat sheet

Everyday: status & commit

Short status with branch and tracking info

```
git status -sb
```

Show unstaged changes

```
git diff
```

Show staged changes

```
git diff --staged
```

Stage changes hunk by hunk

```
git add -p
```

Create a commit with a message

```
git commit -m "Fix login redirect"
```

! Add staged changes to the last commit

```
git commit --amend --no-edit
```

Create and switch to a new branch

```
git switch -c feature/login
```

Log & search

Last 20 commits as a compact graph

```
git log --oneline --graph --decorate -20
```

One author's commits from the last 2 weeks

```
git log --since='2 weeks ago' --author='Daniele'
```

Find commits that added or removed a string

```
git log -S 'password' --oneline
```

Who last changed lines 10-30?

```
git blame -L 10,30 file.py
```

Stash

Stash changes with a description

```
git stash push -m "wip login"
```

Also stash untracked files

```
git stash push -u
```

Apply a stash and keep it

```
git stash apply stash@{0}
```

Apply the newest stash and remove it

```
git stash pop
```

! Discard one stash

```
git stash drop stash@{0}
```

Rebase & tidying history

Replay the current branch onto main

```
git rebase main
```

! Interactively reorder, squash or reword the last 4 commits

```
git rebase -i HEAD~4
```

Abort the rebase, back to the previous state

```
git rebase --abort
```

Merge a branch with a merge commit

```
git merge --no-ff feature/login
```

Undoing things

! Discard unstaged changes of a file

```
git restore file.txt
```

Unstage a file, keep the changes

```
git restore --staged file.txt
```

Undo a commit with a new inverse commit (safe on shared branches)

```
git revert abc1234
```

Undo the last commit, keep changes staged

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

! Throw away the last commit and its changes

```
git reset --hard HEAD~1
```

! Permanently delete untracked files and folders

```
git clean -fd
```

Reflog & recovery

Show the history of all HEAD movements

```
git reflog
```

! Reset to an earlier reflog state

```
git reset --hard HEAD@{2}
```

Rescue a lost commit by hash as a branch

```
git switch -c rescue abc1234
```

Find dangling objects

```
git fsck --lost-found
```

! Undo the last reset, merge or rebase

```
git reset --hard ORIG_HEAD
```

Remotes & bisect

Fetch all remotes, prune deleted branches

```
git fetch --all --prune
```

Fetch and replay local commits on top

```
git pull --rebase
```

! Force push that won't overwrite others' new commits

```
git push --force-with-lease
```

Start a binary search for the breaking commit

```
git bisect start && git bisect bad && git bisect good v1.2
```

Run bisect automatically with a test script

```
git bisect run ./test.sh
```

End bisect, return to the original branch

```
git bisect reset
```