

# Git Cheat Sheet

The most important GIT commands. More on [youssefbee.com](https://youssefbee.com).

## Account Setup

Set a name that is identifiable for credit when review version history

```
1 git config --global user.name "[firstname lastname]"
```

Set an email address that will be associated with each history marker

```
1 git config --global user.email "[valid-email]"
```

## Remote repositories

Add a remote

```
1 git remote add [alias] [url]
```

Fetch all the branches from that Git remote

```
1 git fetch [alias]
```

Update the current branch from its remote counterpart

```
1 git pull [alias]/[branch]
```

Transmit local branch commits to the remote branch

```
1 git push [alias] [branch]
```

## Repository Setup

Add a git repository to an existing local folder

```
1 cd path-to-folder
2 git init
```

Retrieve an entire repository from a hosted location via URL

```
1 git clone [url]
```

## Branches

List your branches. a \* will appear next to the currently active branch

```
1 git branch
```

Create a new branch

```
1 git branch [branch-name]
```

Create and switch to a new branch

```
1 git checkout -b [branch-name]
```

Switch to another branch

```
1 git checkout [branch-name]
```

Merge the specified branch's history into the current one

```
1 git merge [branch]
```

Show all commits in the current branch's history

```
1 git log
```

## Changes

Show modified files in working directory, staged for your next commit

```
1 git status
```

Select (stage) a file as for your next commit

```
1 git add [file]
```

Unstage a file from your next commit (while keeping the changes)

```
1 git reset [file]
```

Show the difference between staged and unstaged files

```
1 git diff
```

Show the difference between what is staged and what is not

```
1 git diff --staged
```

Commit your staged changes

```
1 git commit -m "[descriptive message]"
```

## Stashing

Save modified and staged changes

```
1 git stash
```



List the saved stash entries

```
1 git stash list
```



Apply the latest stash entry.

```
1 git stash pop
```



Discard the latest stash entry

```
1 git stash drop
```



## History

Show the commit history for the currently active branch

```
1 git log
```



Show the commits on branchA that are not on branchB

```
1 git log branchB..branchA
```



Show the commits that changed file, even across renames

```
1 git log --follow [file]
```



Show the diff of what is in branchA that is not in branchB

```
1 git diff branchB...branchA
```



Show any object (branch, hash...) in Git in a human-readable format

```
1 git show [SHA]
```



## Rewrite History

Rewriting branches, updating commits and clearing history

```
1 git rebase [branch]
```



Apply any commits of current branch ahead of specified one

```
1 git reset --hard [commit]
```



## Ignore patterns

Prevent unintentionally staging or committing files **globally**

```
1 git config --global core.excludesfile [file]
```



Prevent unintentionally staging or committing files **in a project** by adding it to `.gitignore`

```
1 echo [file] >> .gitignore
```



Pattern examples

```
1 # ignore all files inside the folder node_modules
2 node_modules/
3 # ignore all files ending with txt
4 *.txt
5 # ignore all folders starting with pattern
6 pattern*/
```

