

[GIT](#)

May 05, 2023

Domantas G.

5min Read

# Basic GIT Commands



Need to learn some basic GIT commands? You've come to the right place. Read on to discover a cheat sheet that you can use for daily reference.

Let's get started!

[Download Complete Git Cheat Sheet](#)

## Understanding the GIT Workflow

GIT is the [most widely used](#) open-source VCS (version control system) that allows you to track changes made to files. Companies and programmers usually use GIT to collaborate on developing software applications.

- **git clone** is used to copy a repository. If the repository lies on a remote server, use:

```
git clone username@host:/path/to/repository
```

- Conversely, run the following basic command to copy a local repository:

```
git clone /path/to/repository
```

- **git add** is used to add files to the staging area. For example, the basic Git following command will index the temp.txt file:

```
git add <temp.txt>
```

- **git commit** will create a snapshot of the changes and save it to the git directory.

```
git commit -m "Message to go with the commit here"
```

## Pro Tip

Note that any committed changes won't make their way to the remote repository.

- **git config** can be used to set user-specific configuration values like email, username, file permissions, and so on. To illustrate, the command for setting up an email will look like this:

```
git config --global user.email youremail@example.com
```

- The `--global` flag tells GIT that you're going to use that email for all local repositories. If you want to use different emails for different repositories, use the command below:

```
git config --local user.email youremail@example.com
```

- **git status** displays the list of changed files together with the files that are yet to be staged and committed.

```
git checkout <branch-name>
```

- **git remote** lets you view all remote repositories. The following command will list all connections with their URLs:

```
git remote -v
```

- To connect the local repository to a remote server, use the command below:

```
git remote add origin <host-or-remoteURL>
```

- Meanwhile, the following command will delete a connection to a specified remote repository:

```
git remote rm <name-of-the-repository>
```

- **git branch** will list, create, or delete branches. For instance, if you want to list all the branches in the repository, the command should look like this:

```
git branch
```

- If you want to delete a branch, use:

```
git branch -d <branch-name>
```

- **git pull** merges all the changes present in the remote repository to the local working directory.

```
git pull
```

- **git merge** is used to merge a branch into the active one.

```
git merge <branch-name>
```

- **git diff** lists down conflicts. In order to view conflicts against the base file, use

```
git diff --base <file-name>
```

```
git reset --hard HEAD
```

- **git rm** can be used to remove files from the index and the working directory.

```
git rm filename.txt
```

- **git stash** command will temporarily save the changes that are not ready to be committed. you can go back to that project later on.

```
git stash
```

- **git show** is a command used to view information about any git object.

```
git show
```

- **git fetch** allows users to fetch all objects from the remote repository that don't currently exist in the local working directory.

```
git fetch origin
```

- **git ls-tree** allows you to view a tree object along with the name, the mode of each item, and the SHA-1 value. Let's say you want to see the HEAD, use:

```
git ls-tree HEAD
```

- **git cat-file** is used to view the type and the size information of a repository object. Use the object's SHA-1 value along with the object's SHA-1 value to view the information of a specific object, for example:

```
git cat-file -p d670460b4b4aece5915caf5c68d12f560a9fe3e4
```

- **git grep** lets users search through committed trees, working directory, and staging area for specific phrases and words. To search for `www.hostinger.com` in all files, use:

```
git grep "www.hostinger.com"
```

- **gitk** shows the graphical interface for a local repository. Simply run:

```
git fsck
```

- **git rebase** is used to apply certain changes from one branch to another. For instance:

```
git rebase master
```

## Basic GIT Commands Cheat Sheet in .pdf

If you are just starting out with GIT, it can be hard to remember even the basic commands. For this reason, we've put together a GIT cheat sheet to help you master the software. Save the file to your device so you'll always have it ready when you're stuck remembering GIT commands.

## Conclusion

Learning basic GIT commands will go a long way for developers as they can easily control their source code. It might take some time to commit to remembering all of them, but hopefully, our cheat sheet will be helpful for you.

Practice those commands and make the most of your developing skills! Good luck!

## Basic GIT Commands FAQ

**What Are the Most Used GIT Commands?**