

---

# Git Installation

---

Git was created as a command-line tool. To use it, we run various git commands in a Unix shell. This is not the most user friendly experience, but it's at the very core of Git!

Over the last few years, companies have created graphical user interfaces for Git that allow people to use Git without having to be a command-line expert. There are many examples, including Github Desktop, SourceTree, Tower, GitKraken, and Ungit. I use GitKraken (it's free to install)

Installing Git locally is slightly trickier, depending on your specific operating system. Git is intended to run on Unix-style systems like Linux and MacOS, so if you're on a Windows machine you may need to take a couple extra steps.

## macOS Install

1. It's typical that your Mac would have Git pre-installed. First, check to see if you have Git installed already using the command:

```
git --version
```

2. If not, or if you have an old version, download the latest Git installer package using [this link](#), the homebrew option is what I used.
3. Once you have it installed, you will need to update to this version of git. This is because macOS decides which program to run by searching through the directories listed in the PATH environment variable, in order, from left to right. And so, when you use git, your Mac finds a directory that contains the older Git installation, specifically `/usr/local/bin`, (use command `which git` to see for yourself) that would generally appear earlier in your PATH than Homebrew's directory, `/opt/homebrew/bin`. And so, to ensure Homebrew's bin directory is at the beginning of the PATH, paste in the command:

```
export PATH="/opt/homebrew/bin:$PATH"
```

4. Reload your shell with command `source ~/.zshrc`. Now, when you run `git --version`, you should have the latest version of git installed by homebrew.

## Windows Install

Git was designed to run on a Unix-based interface, and so the problem that we run into on a Windows machine is that they don't come with a Unix-based prompt by default. Windows comes with a different command line interface called Command Prompt, which is not Unix-based. So, we must download Git Bash.

1. Go to [this link](#) and download the standalone installer. Follow the instructions you are prompted with. Note that one of the prompts will ask you to choose your default editor used by Git. The default is set to Vim, which if you're not familiar with, can be a little tricky to use. I set my default editor to VS Code. It will prompt you to install VS Code if you haven't already.
2. After you launch Git Bash, you can run the command `git --version` to confirm it has been installed.

## Configuring Your Git Name

This is important especially if you have multiple people working on the same project. By configuring your name, you'll be able to see who made what changes on your project by combing through `git log`. To see you already have a name configured, paste the command:

```
git config user.name
```

To configure the name that Git will associate with your work, run this command:

```
git config --global user.name "Your Name"
```

You can also do the same for your email, using the following command. When you get to Github, you'll want your Git email address to match your Github account.

```
git config --global user.email email@here.com
```