

# Git Basics

A Crash Course

Feb. 16, 2021



## 1 History

Git is the distributed version control system (VCS) designed and developed in 2005 by

Linux Torvalds, the creator of the Linux Kernel. Its goals included speed, data integrity, and support for non-linear work-flows (branching on different systems) [4].

## 2 Terminology

- remote → A repository on another machine/server
- branch → A split from the main branch
- untracked → A new file that is not tracked by Git
- tracked → Changes are being tracked by Git
- staged → The file is added but changes are not yet comited

## 3 Installation

There is a lot of information related to Git and the various commands. Below we will cover a simple work flow but any of the more specific commands can be found with a search or through the official documentation pages [1].

### 3.1 Linux

These instructions should work on the distro's and their WSL counterparts.

#### 3.1.1 Debian-based

- `sudo apt update`
- `sudo apt upgrade`
- `sudo apt install git`

#### 3.1.2 Arch-based

- `sudo pacman -S git`

#### 3.1.3 Redhat-based

- `sudo yum upgrade`
- `sudo yum install git`

### 3.2 Windows

- Install following this link

I recommend installing git-bash, but the normal Windows cmd-line or Powershell will work

### 3.3 Mac

Disclaimer: Am not a Mac user, this was found doing some research. Homebrew makes installing developer easier (Check references for more details).

- install Homebrew through the macOS terminal [2]
- `brew install git`

## 4 Git Configuration

Now that Git is installed, we are ready to configure Git so it knows who we are. In the command line type:

1. `git config -- global user.name ' < insert name here > '`
2. `git config -- global user.email ' < insert email here > '`

## 5 A Visual Aid

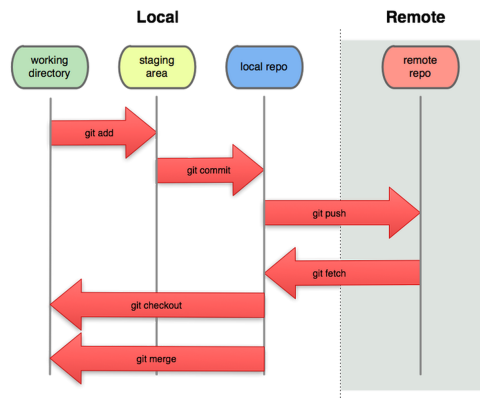


Figure 1: A normal Git progression [3]

## 6 Basic Work-flow

Project initialization begins with the following command being run inside the project directory/folder

- **git init** → initialize the project
- **git add < filename >** → add a single file to be staged for a commit
- or **git add \*** → add all modified or untracked files to be staged
- **git commit -m 'message here'** → Commit (save) the changes of the project
- **git remote add origin <server>** → Add a remote repository
- **git push origin master** → A standard push to the master/main branch

## References

- [1] Git-SCM. <https://git-scm.com/book/en/v2/getting-started-about-version-control>.
- [2] Homebrew. <https://brew.sh/>.
- [3] Kevin Kononenko. Git flow char, 2021.
- [4] Wikipedia. Git, Feb 2021.