

## Introduction

Before starting a project with Git commands, let's first understand a little bit more about Git.

## How does Git track changes to code files?

I previously mentioned that Git is a version control tool, so in a nutshell, once Git is initialized in a folder, it will track changes that you make to your files. For example, I have the *index.html* file. How does Git attract changes to your code files?

The way Git tracks changes made to *index.html* is by moving it through three states:

- **Modified** - the file had some changes made to it and you still might be working on making a few more changes.
- **Staged**. After you're done changing the file, it will become staged. If you weren't using a version control tool like Git you'd save your *index.html* file with modifications under a new name to keep the previous version of the file and the new modified version. With Git things are much simpler. Your file with modifications is staged, meaning that it waits for Git to take a snapshot of these changes.
- **Committed** - a snapshot of the current state of your *index.html* file has been taken.

As your file moves through three different states, these three different states occur in three main sections of a Git project. By Git project I mean a folder with all your code files that has been initialized with Git. The three main sections:

- **Working Directory** - this is where you are currently doing all your work. It can be thought of as the folder you're currently working in. This contains files in the Modified state.
- **Staging Area**. So you've made a change to your file and saved it locally - the file moves on to the staging section. You haven't as yet indicated to Git that you'd like to keep a record of this change, and the way you do that is by adding your modified file to the Staging Area. You can think of the Staging Area as a photo booth.
- **Git Repository**. The next step for Git is to take a snapshot of the file change, that is, commit the file change. Once that's been done, your file now exists in the Git repository.

I previously mentioned that a repository contains your commit objects. This is what I meant: Git repo contains files that have been committed in the current estate. It consists of snapshots of your file changes.

## Git Commands

**Git commands** are structured by the use of the `git` keyword followed by the action that you want to perform. For example, `git add` will add your file to the Staging Area. `git commit` will take a snapshot of the current state of your file and so on.

## Git Workflow

A **workflow** is just a sequence of steps to complete a task. Let's create a simple Git workflow: you modify your files, then you stage them, and lastly, you commit them. Once that's done, the current state of your modified file or files now exists in the Git repository.