Install Git

What is Git?

Git is a version control system that keeps track of changes to your files over time. It allows you to create commits, which are snapshots of a codebase at a moment in time.

A helpful analogy to Git is maintaining different *drafts* of an essay. Git allows you to return to any of those drafts/snapshots to explore what the code was like and to see how it evolved over time.

You do *not* need to know Git to be effective with Rust. However, Git is a common technology for *any* modern programming project. It also does not take too much time to learn (<1 hour).

Setup

We're not going to be using Git too much in the course but:

- the Cargo tool depends on it
- it will enable you to easily download the course's materials.

For macOS: The XCode Command Line tools we setup should include Git. Execute:

1. git --version

in your Terminal to confirm.

For Windows: The Visual Studio editor we setup should include Git. Execute

1. git --version

in your PowerShell to confirm.

If the git command does not work, head to https://git-scm.com/. The web browser will detect the operating system for your computer and offer you the correct download option.

Working with Git

In Git, we make commits. A commit is like a "checkpoint" or "save point" for a project in its current state. We can then return to any one of those commits to see what the project looked like at that point.

You can then interact with Git using the `git` command.

If it's your first time using Git, I recommend Googling the following commands:

- git init
- git status
- git add
- git commit
- git checkout
- git restore

I recommend making a commit at the end of every course lesson so you can return to the last working version of the code if something breaks.