**GIT Init**

The git init command creates a new Git repository. It initialize a new, empty repository or convert an existing, unversioned project to a Git repository. This is usually the first command you'll run in a new project.

Executing git init creates a .git subdirectory in the current working directory, which contains all of the necessary Git metadata for the new repository.

By default, git init will initialize the Git configuration to the .git subdirectory path

Git init

Transform the current directory into a Git repository. This adds a .git subdirectory to the current directory and makes it possible to start recording revisions of the project.

git init <directory>

Create an empty Git repository in the specified directory. Running this command will create a new subdirectory called containing nothing but the .git subdirectory.

If you've already run git init on a project directory and it contains a .git subdirectory, you can safely run git init again on the same project directory. It will not override an existing . gitconfiguration.

Git init has a few other command line options.

--BARE

Creates a bare repository. The --bare flag creates a repository that doesn’t have a working directory, making it impossible to edit files and commit changes in that repository.

-TEMPLATE=<TEMPLATEDIRECTORY>

Specifies the directory from which templates will be used

**GIT FORK**

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.