Lecture 5: Git and GitHub

Version Control:

Version control is a system that tracks changes to files over time, allowing collaboration, maintaining a history of revisions, and managing conflicts in projects like books, designs, websites, or software.

We generally use the following notation to indicate the changes on our project:

0.0.0, where each number means the following:

- x.0.0 : The first one indicates that there was a big change, a major update, of a fundamental improvement added to the project.
- 0.x.0 : The second one indicates that a minor update such an improvement in a feature, of a new feature happened in our project, but the same fundamentals are still the same.
- 0.0.x : The last one indicates that there was a small bug fix, a dispatch, or generally a change that not necessarily everyone or anyone at all should notice

To implement such functionality, we use what we call Version Control tools such as Git, to track, control, and document these changes in our project.

Git is a distributed version control system widely used for its speed, efficiency, and ability to track and document project progress. It helps developers work on projects collaboratively, revert to previous states, and maintain a clear history of changes.

Key Git Commands:

1. **`git init`**: Initializes a new Git repository in a directory to start the tracking of files in this directory.
   Example: `git init`
2. **`git add`**: Stages or adds files for a commit. Use this to track new or changed files.
   Example: `git add filename` or `git add .` (all files)
3. **`git commit`**: Saves staged changes with a message describing the changes.
   Example: `git commit -m "Initial commit"`

4. **`git log`**: Displays a history of commits.
   Example: `git log`
5. **`git status`**: Shows the current state of the working directory and staging area, to showcase the state of every file to see if it was staged/added or not.
   Example: `git status`
6. **`git checkout`**: Switches branches or restores files from a previous commit.
   Example: `git checkout branch-name` or `git checkout -- filename`
7. **`git clone`**: Copies an existing repository from a remote source to your local machine.
   Example: `git clone repo-url`
8. **`git push`**: Uploads local commits to a remote repository.
   Example: `git push origin branch-name`
9. **`git pull`**: Fetches and merges updates from a remote repository.
   Example: `git pull origin branch-name`

These commands help in efficiently managing project workflows and collaborating with others.

There are many platforms that allow us to use the Git tools remotely to be able to do the same tracking in the cloud, and make it easier to share our work with others and collaborate with them. One of these platforms that we will be covering and using is going to be GitHub/

GitHub is **a web-based interface that uses Git, the open source version control software that lets multiple people make separate changes to web pages at the same time**. As Carpenter notes, because it allows for real-time collaboration, GitHub encourages teams to work together to build and edit their site content.