Evolve Full Stack Developer

Tech Appendix 1 - Git and GitHub Basics

INCEPTIONU

Air conditioning...I knew I forgot something...

INCEPTIONU

Setup and Prerequisites

Prerequisites

- Quality of Life Tips for a Developer
- Files, directories and naming conventions
- Command Line Basics
  - Command line practice: Follow the white rabbit

Git and GitHub Setup

- Installing Git for the first time
- Setting your Identity
- Optional: see Setting up SSH keys on the last slide if you're getting an authentication error when pushing to GitHub.

What is Git?

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

- from Wikipedia
- See also: What is Git?

Why do we use it?

- **Clone third party projects**: Popular projects include lodash, TailwindCSS and Font Awesome, but there's a whole world of projects you can clone.
- **Code backups**: Git allows you to take snapshots of a directory (commits) which we can rollback to in case we royally screw up.
- **Syncing code**: If you work on two machines, like a home laptop and office system, you can use Git to sync your projects.
- **Collaboration with teammates**: Git allows multiple developers to work on the same file at the same time. Conflicts will happen but Git helps resolve them.
- **Deployment**: In industry, we use Git to sync our project code with a server to make it live.

Terminology

- **Version Control:** A category of software tools that help a software team manage changes to source code over time.
- **Repository (aka repo):** The root directory of a project that Git tracks. It contains a hidden `.git` directory at the top of the project.
- **Local Repository:** A repo that is located on your local machine.
- **Remote Repository:** A repo that is located on another machine or server. For the purposes of this course, all of your remote repos will be located on GitHub.
- **To commit changes**: Saving a snapshot of the current state of your project.
- **To add changes:** An extra step you need to complete before you can commit a change (for safety).
- **To push changes:** Sending code from your local repo to the remote repo.
- **To pull changes**: Retrieving code from the remote repo to your local repo.

How does it work?

- At the end of the day, Git keeps track of each line of code in your project, including:
  - who created the line(s) of code;
  - every change made to a line(s) of code;
  - who made each change and when.

- A typical sequence of events in a developer's day:
  - A developer will work locally on their own system in a Local Repo, making changes to code.
  - When code is finally working, they `add` the changes to "staging" (this must be done before you can commit the changes);
  - They then `commit` the changes to create a "snapshot" of your project;
  - They `push` the changes to a Remote Repo (i.e. GitHub) when the changes are ready to be shared or deployed.
    - Note: you must `pull` any new changes from the Remote Repo before you can `push`. This is when dreaded merge conflicts need to be handled.

- At any point during development, you can view the `status` of your repo to find helpful information of the status of your project.

Key Takeaways

Some things to think about when first learning Git:

- It's very important that you're in correct directory (usually the root of your project) when running Git commands. Practice your command line skills and always keep in mind which folder you're in.

- It's a pain, but you always have to `add` your changes before you `commit` them. This is to prevent you from accidentally committing code you didn't mean to.

- Don't forget to add the `-m "commit message here"` when using the `git commit` command. Otherwise, your terminal will open a `vim` window so you can add the mandatory commit message. It's the default command line text editor on most systems and is not the most intuitive application to use. See this Vim cheatsheet if you need a command (`:q!` will quit without saving so you can try committing again with the `-m` flag).

INCEPTIONU

Git and GitHub Basics

Starter Exercises:

- Clone Happy
- Publish a webpage with Git and GitHub Pages
  a. Create a GitHub repository
  b. Clone a GitHub repository to your workspace
  c. Commit a new index.html file
  d. Commit new changes to index.html
  e. Push your changes to GitHub
  f. Deploy your GitHub repo to GitHub Pages!

Extra Activities

Collaborating with Git

- Activity: Create a merge conflict (on purpose)
- Activity: Commit Catch

Setting up SSH keys

GitHub has recently made changes to their authentication policy and they may force you to connect to your repos via SSH (instead of logging in via your GitHub account using HTTPS). If this happens, follow the steps below to set up your SSH keys (it's a pain but only has to be done once per machine).

- About SSH
- Check for existing SSH keys
- Generate a new SSH key and add it to the ssh-agent
- Add your SSH key to your GitHub account
- Test your connection