Github Tutorial

Narayani Vedam

2023-09-02

GitHub is an online platform that serves as a central hub for software development and collaboration. It's a place where individuals and teams can store, manage, and share their code.

Key points to understand:

1. **Repositories:** GitHub uses repositories (often referred to as "repos") to store code projects. Think of a repository as a folder that holds all the files, code, and documentation for a particular software project.
2. **Version Control:** GitHub uses a version control system called Git, which allows you to track changes to your code over time. This makes it easy to collaborate with others, review changes, and revert to previous versions if needed.
3. **Collaboration:** GitHub is designed for teamwork. You can invite others to work on your projects, manage access permissions, and track issues and tasks related to your code.
4. **Open Source:** Many projects on GitHub are open source, meaning their code is freely available for anyone to view, use, and contribute to. It's a great place to learn from others and contribute to the global software development community.
5. **User-Friendly Interface:** GitHub provides a user-friendly web interface where you can explore code, create and manage repositories, and engage with other developers through discussions, issues, and pull requests.

In essence, GitHub is a platform that simplifies the process of working on software projects, encourages collaboration, and promotes transparency in code development. Whether you're an individual coder, part of a team, or just starting to explore the world of programming, GitHub is a valuable tool to have in your toolkit.

Step-by-Step Guide

Step-1: Sign Up for a GitHub Account

- Open your web browser and go to GitHub's website, https://github.com/
- Click on the "Sign Up" button in the upper right corner.
- Follow the prompts to create your GitHub account.
- You'll need to choose a username, provide an email address, and set a password.

Step-2: Install Git

GitHub uses Git for version control. If you don't have Git installed on your computer, follow these steps:

For Windows

- Download the Git installer for Windows from the Git website.
- Run the installer and follow the installation instructions.

For macOS

- Open Terminal (you can find it in the Utilities folder within the Applications folder)
- Install Git using Homebrew by running the command: brew install git
- Follow any additional prompts

For Linux (Ubuntu)

- Open a terminal
- Install Git with the command: `sudo apt-get install git`

Step 3: Set Up Git

- Before you start using Git, you need to configure your identity
- Open a terminal or command prompt and run the following commands
- Use the same user name and email that you used in Step 1

```
git config --global user.name "Your Name"
git config --global user.email "youremail@example.com"
```

Step 4: Create a New Repository

- Log in to your GitHub account.
- Click on the "+" icon in the top-right corner and select "New repository."
- Fill in the repository name, description, and other settings.
- Choose to make the repository public or private (public repositories are visible to everyone, while private ones are only visible to you and collaborators).
- Click "Create repository."
- See here for illustrations, https://towardsdatascience.com/the-easiest-github-tutorial-ever-4a3aa0396039

Step 5: Clone a Repository

Cloning allows you to download a copy of a repository to your local computer.

Method: A

If you prefer a graphical interface, you can use GitHub Desktop to clone repositories.

- Download and install GitHub Desktop for your operating system.
- Open GitHub Desktop and sign in with your GitHub account that you created in Step-1
- Click on the "File" menu and select "Clone Repository."
- In the "URL" field, paste the repository's URL (e.g., https://github.com/username/repository-name.git).
- Choose a local directory where you want to store the cloned repository
- Click the "Clone" button

Method: B

- There is an alternate way to clone a repository, mentioned below:
- Open the repository you want to clone on github.com
- Click the "Code" button and copy the repository's URL (e.g., https://github.com/username/repository-name.git).
- Open a terminal or command prompt.
- Navigate to the directory where you want to clone the repository using the cd command, `cd Insert_Path_to_the_Folder_Here`
- Run the following command to clone the repository by replacing the URL with the one you copied,
  `git clone https://github.com/username/repository-name.git`

Step 6: Make Changes

- Open the cloned repository on your computer
- Make changes to the files inside the repository

Step 7: Commit and Sync Changes

Method: A

- In GitHub Desktop, you'll see a list of the changes you've made
- Enter a brief summary of your changes in the "Summary" field
- Click the "Commit to main" (or other branch name) button to save your changes locally

Method: B

- Open a terminal or command prompt
- Navigate to your repository's directory using the cd command
- Run the following commands to commit your changes
- Replace "Your commit message here" with a brief description of your changes.

```
git add
git commit -m "Your commit message here"
```

Step 8: Push Changes to GitHub

Method: A

- To push your changes to GitHub, click the "Push origin" button

Method: B

- Run the following command to push your changes to GitHub:

```
git push origin main
```

- This command pushes your changes to the "main" branch. If your repository uses a different branch, replace "main" with the appropriate branch name.

Step 9: can ignore since our work is indiv

If you're working on a collaborative project, you can create a pull request to propose changes to a repository

Method: A

- In the GitHub Desktop app, click the "Current Branch" dropdown and select the branch you want to create a pull request for
- Click the "Create Pull Request" button
- Enter a title and description for your pull request
- Click the "Create Pull Request" button again to submit it on GitHub

Method: B

- Go to the repository on github.com
- Click on the "Pull Requests" tab
- Click the "New Pull Request" button
- Select the branch with your changes and the branch you want to merge your changes into
- Add a title and description for your pull request
- Click "Create Pull Request."