An Introduction to GitHub

Alton Barbehenn

February 23, 2016

First Things First

First you need to create a free account on https://github.com/. If you want to use GitHub offline and still track your changes they have a desktop application available at https://desktop.github.com/. A great resource for learning this tool is https://guides.github.com/. This site walks you through a few simple procedures. If I don't define something well, https://help.github.com/articles/github-glossary/ is a good place to start if you don't understand a term.

Basics

GitHub allows people to collaborate on projects and it tracks the changes that you and others make to files. There are three main actions you can do: place pull requests, commit documents, and make and merge branches. GitHub tracks all these changes allowing people to take projects in a new direction or just to collaborate in an environment that allows for effective idea sharing.

The fundamental element of GitHub is the repository. A repository is a collection of files that make up the project that you are working on. Think of a repository as a folder that contains your project.

Branches

Being able to branch a repository is what allows multiple people to work on a project without getting in each other's way. It also lets you take somebody else's project in a new direction without stopping or confusing their project.

To create a new branch in the browser you'll need to click the branch button and either select an existing branch or type the name of a new one.

Branches can be merged back into the master branch or other branches when and if you are done with them. I have found that manipulating branches is easier in the browser version. In the browser you can go to the branches tab and select make a pull request. After this action, you can choose to merge branches, delete a branch, or you can do both actions. You can delete branches at any time when looking at the branches tab, not that you should need to do so.

Pull Requests

Pull requests are one of the most amazing things that GitHub does for your work. When you are done working in a direction, or want to see what you've done, a pull request allows you to compare versions of the repository and shows you character-for-character what changes have been made. In the case where conflicting changes have been made, a pull request will help you resolve these conflicts.

Commits

Commits are what GitHub calls all of the updates you make to your repository and files there in. They allow you to track your edits as you work on your project. While you only need to write a title for each commit, a short description is often nice so that when you want to go back in and document your work more thoroughly you can.

Working in somebody else's repository

This is important because we want a lab repository that holds all of the data we want in a useable environment. To do this you first need to for the repository that you want to modify. The easiest way to do this is in the browser interface, go to the repository you want to fork and click the fork button in the top right. This creates a copy of the repository in your account.

From here you can make all the changes you want in the normal way. The difference is that when you're done working you can place a pull request to the original repository. And with the correct permissions, you can complete the pull request and merge the two branches. Once the pull request is completed, you can merge the branches in the browser by having the owner of the original repository confirm and merge the pull request.

File Names and File Paths

To change a file's name or path select the file you want to edit and click the pencil icon in the top right hand of the window. This allows you to edit the file. At the top of the window, there is a file path box. Here you can type the desired file path or just change the file name by back spacing to the desired folder then typing out the proper extension. GitHub interprets the forward slashes as the end of a folder name.

Uplaoding Files

To upload a file to your repository you simply need to use the drag and drop function. Go to the desired location in your repository and drag the file from a directory into it and commit the changes. Once it is uploaded, you can either edit the code in the edit window or download the code, edit it, then re-upload it.