Git is a version control system that allows you to commit code and restore different revisions of a project as required. Github doesn't care what language you use as it sees it all as text files. Github builds upon git by providing the ability to work with a team of developers within a remote repository. On top of committing code with git, Github allows any member of your team to push and pull code from a remote repository. Github also keeps track of a "tree" of edits with the main branch being the "trunk." Branches can deviate from and merge back into the main branch allowing separately-written segments of code to be merged.

Commits are a way of saving your current code separately from what is being stored locally. This provides a backup that can be restored later should something happen, like a hard drive failure. These commits can be pushed to github, which will allow you to put code into a project's remote repository as opposed to having to rely on saving it locally. You may need to open a pull request so that someone managing the repo can review and merge the code you submit; this prevents any unwanted or unfinished code from being merged into the main branch and causing long-term bugs. When code is merged, if there are no conflicts, it's combined into the master branch. In the event of a merge conflict, someone often needs to edit the merging branch to resolve the conflict. Each different branch should be kept close to the main branch so that merge conflicts are less likely to occur. I've experienced them most frequently when 2 people push a commit that includes changes to the same file. The benefit of using different branches is that instead of on the main branch directly, multiple developers can work simultaneously on different features in the same sections of code.

Forking a repo is a great way to obtain a copy of a repository that you can work on, push to, and build in order to add and test a new feature; you can then use that forked repo in order to create a pull request to the main repository as mentioned above, and get the code you wrote merged.

Atlassian. “Git Workflow: Atlassian Git Tutorial.” Atlassian, www.atlassian.com/git/tutorials/comparing-workflows. Accessed 14 Nov. 2023.