

Collaborative Software Development

Learning Objectives

- Understand the key features of GitHub.com
- Implement a collaborative software development workflow using Git and GitHub

GitHub as CSD Tool

- Although there are various flows and methodologies for developing software collaboratively, GitHub is starting to see major use at software companies
- This is due to its project management features in addition to the ability to store multiple versions of code

GitHub Features for CSD

a tour of github.com

- [Issues](#)
 - Milestones
 - Labels
 - Commit Keywords (`closes`, `fixes`, etc.)
- [Pull Requests](#)
- [Diffs](#) (View the difference in a commit compared with another commit)
- Releases

Exercise: Issues

- Using one of the repositories you've pushed to GitHub, open up a new issue
- Make a change and then a commit locally including the words "closes #1", then push up to GitHub
- The issue you created, as long as it was the first issue in the repo, should now be closed and reference your commit/code
- You can also use closes, closed, fix, fixes, fixed, resolve, resolves, resolved before the issue number to close a commit

More on Issues

- The issue will only be closed if the commit you've made is done on the default branch, typically called master
- You can also assign other collaborators in your repository to an issue, making it a great platform for project management
- Issues can be part of a milestone, a larger organizational goal or perhaps an actual "release" of your software

Resources

Codecademy

Learn Git - Git Teamwork

TeamTreeHouse

Get Together with Git