

Intro to Git/Github

Learning Objectives

- Understand Git and how to use it
- See the benefits of using Git/Github while at CCIS
- Learn primary git commands in order to be functional with Git
- Go over more advanced commands (merge, rebase, etc)
- Create barebones homepage using Github Pages

Materials:

- Laptop
- GitHub account
 - Must be made in advance, blackboard assignment to submit username
- The ability to use git commands (git shell for Windows, make sure git installed for Mac)
 - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
 - Xcode cmd line tools will install git, but you can install git separately as well for Mac
 - Windows alternatives include:
 - Git Shell(Click Windows): <https://desktop.github.com/>
 - Git-Bash: <https://git-scm.com/download/win>
 - Posh Git: <https://github.com/dahlbyk/posh-git>

Lesson Activities (with estimated times)

Git Basics (45 min)

- Allow students to get set up (5 min)
- Explain Git and Github/Bitbucket/etc (2 min)
 - Version Control System
 - Github/Bitbucket house repositories to manage version control
- Why is this important? (3 min)
 - Industry almost exclusively uses git
 - As a student, can get free private repos
 - CCIS also pays for enterprise, can join their org for more benefits
- Clone new repository (5 min)
 - Pre-created test-repo
 - May have to show them how to use terminal/shell commands
- Git Tasks: (30 min)
 - Create a new branch (5 min)
 - `git checkout -b [branch_name]`
 - `git push -u origin my_branch`
 - Explain branching/branches and why they're important
 - Create a new file and edit it (5 min)
 - Show text editors, how to use
 - Stage changes (5 min)

- git add -A (all but can specify file)
- Commit changes (5 min)
 - git commit -m "committing new file"
- Push changes to github (5 min)
 - git push origin [branch_name]
- Create Pull Request on github (5 min)
 - Go over merging, rebasing

Github Pages (40 min)

- What is Github Pages? (5 min)
 - Allows people to host static web page for their account and projects
 - Basically you can make a free homepage
- Create a new repository and name it [your_github_name].github.io (5 min)
- Clone it to your machine (2 min)
- Create a file named index.html (3 min)
- Edit it to add some text (5 min)
- Stage, Commit, Push the changes (5 min)
- Look in browser at [your_github_name].github.io (5 seconds)
- I'll give you guys [insert remaining time left in 40 min] to work on these a little bit (10 min)

Best Practices and Continuing Education

- Best Practices
 - You should probably not have a reason to force push
 - Use branches for larger bodies of work
 - Stage and commit frequently (can always fix later)
- Continuing Education
 - Code Academy has a tutorial, among many other websites
 - NU Scout is a design studio, they put on workshops for building homepages, etc.

Tip of the Week

- Join Computer Science Facebook Group