Assessment: Git for Teams

To earn a micro-badge for this workshop, complete the prompts on the following page. You will need to do your work across a few different platforms (a command line editor of your choosing and your GitHub account). This assessment does not ask you to create any code or data; rather, you will be working from your computer and GitHub on a project of your choosing and taking screenshots of the results. We encourage you to select a repository for one of your own projects, but you may also create a test repository if need be.

For questions marked "Short answer", record your notes in a plain text file titled "answers.txt". Be sure to note which question you're responding to (e.g. 1b, 3c, etc.). When you're finished with everything, create a new folder. Move your screenshots to this folder along with answers.txt. Zip it up and submit it to GradPathways.

Links

- GradPathways badge
- Event page
- Workshop reader

Rubric

1. Working code: were you able to complete each prompt successfully?
2. Understanding your actions: can you explain what your actions do and why you implemented them?
3. Critical reflection: do your short answers provide context (conceptual, domain-specific, etc.) for your actions?

Prompts

1. Making a project
    a. First, select one of your own Git repositories to use for the assessment. If you don't have a repository on hand, feel free to make a simple example. If you go the latter route, put a few files in this repository and commit them to Git.
    b. Within your repository, create a data folder. Prevent Git from tracking it (hint: this is a special file).
    c. (Short answer) In a sentence or two, explain why it's a good idea to prevent certain files from being tracked.
    d. Screenshot your repository as viewed from your computer's finder window. Name this screenshot "1d.png".
2. Tracking the project on GitHub
    a. Now, create a corresponding remote repository for your project on GitHub. If you don't already have a README for this project, make that as well. Your README should have some descriptive information about what the repository contains.
        - **Note**: if you're already tracking this repository on GitHub, you don't need to make another one, but do make sure there's a README.
        - Want some pointers on how to write a good README? See the DataLab's reader on project organization
    b. Link the local and remote versions of your repository. If you've made any changes on GitHub, pull them down to your own computer.
    c. Screenshot the repository as it exists on GitHub. Name this screenshot "2c.png".
3. Making an issue
    a. On GitHub, create a new issue that describes some change you'd like to do to your project. Your issue should have a short, informational title and a sentence or two of description about the changes you want to make. Assign yourself to the issue.
    b. Screenshot the issue. Name this screenshot "3b.png".
    c. On your local computer, checkout a new branch. Make whatever changes you need to make and commit them on your local computer.
    d. Push your new branch to GitHub. On GitHub, open a pull request to merge the new branch into your main branch.
    e. Screenshot the pull request. Name this screenshot "3e.png".
    f. Close your issue, using a brief comment to describe your changes.
4. Forking a repository
    a. (Short answer) Finally, search for another repository on GitHub. Try to find something that is interesting to you. In a few sentences, explain why you've chosen this repository.
    b. Fork this repository into your own GitHub account.
    c. Screenshot the forked repository as it exists in your own account. Name this screenshot "4c.png". With that, you're done!
5. Finishing up
    a. To finish, make a new folder and put your screenshots inside it, along with answers.txt. Zip this folder and submit it to GradPathways.