CSC-450, Github Assignment

Introduction

Github (https://github.com) is a development platform and is probably the most popular web-based source code management system. Github uses uses *git* for revision control and source code management. In addition, Github implements collaborative features such as bug tracking and feature requests. Github is free to use for public repositories (a monthly charge is required for private repositories). Other source code management systems such as BitBucket (https://bitbucket.org) are available as well. Both command line *git* tools and a GUI tool called Github Desktop are available.

Why use Github?

- Github is a version control system:
  - you can easily revert to a previous version of code, if needed
  - you can easily see how files change from one version to another
  - branches can be used to separate major versions of code (e.g., release and development versions; see figure below)
- Github makes it easy to work with existing code and to collaborate:
  - a user can (a) *fork* (copy) code from anyone's repository, (b) clone the code (i.e. make a local copy for development), (c) make changes to the code, (d) commit (record) the changes and (e) push (save) the changes to the remote repository (i.e., the respository on github.com).
  - if appropriate, the user can submit a *pull request*, which is a request to merge his/her code with another user's version which was originally forked.
  - pull requests can be automatically merged if there are no conflicts; otherwise conflicts are noted and must be manually resolved.
- Github makes it easy to automate workflows, such as testing and deployment, through Github actions (https://docs.github.com/en/actions)

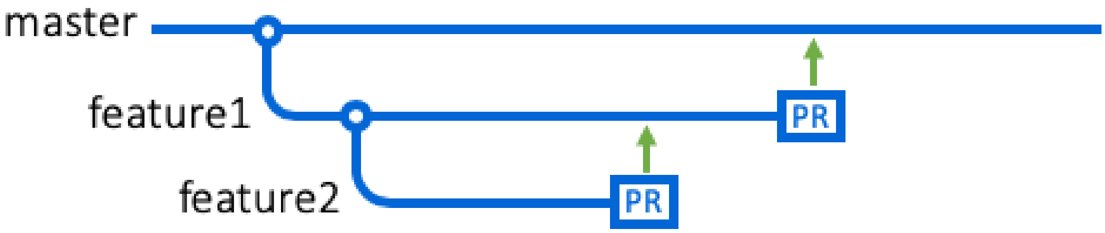


Figure 1. Example Github branches, source: http://nvie.com/posts/a-successful-git-branching-model/

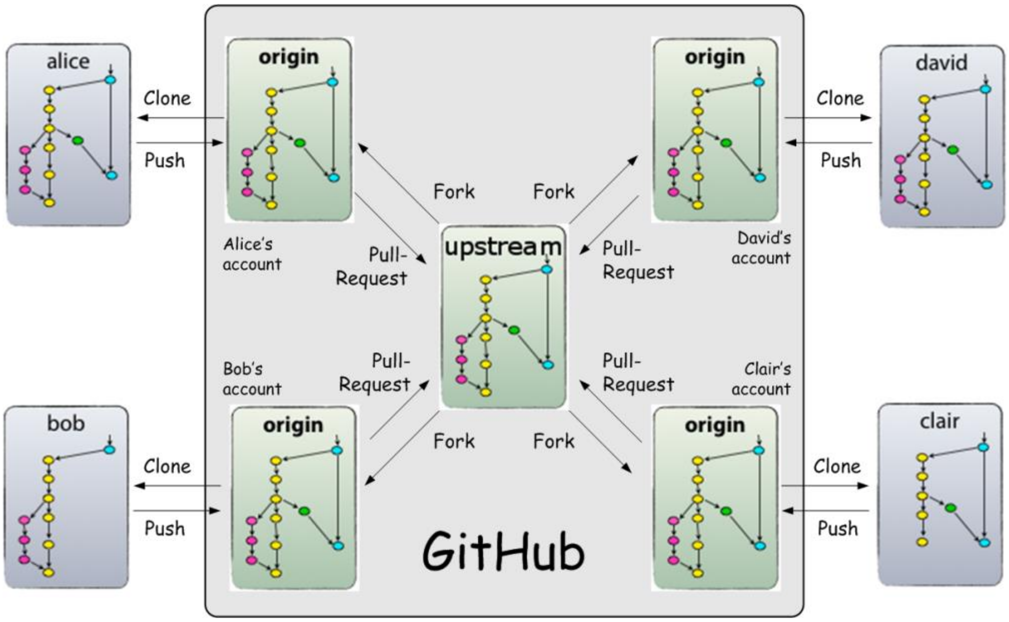


Figure 2. Collaboration on Github, source: http://www.dalescott.net/2012/09/14/using-gitflow-with-githubs-fork-pull-model/

Reproducible research

Reproducible research is fundamental to scientific research. For research involving code or data, access to source code and data is necessary to reproduce and verify published results. By making your software and data available, others will be able to reproduce (i.e., validate) your findings, and can expand on what you have done, either by asking different questions about your data and/or modifying the code you have developed.

As part of the project requirement, you must make your source code and data available. We will use Github for this purpose (other resources for hosting your code/data can be used with permission). In addition to supporting your research project, having a Github account is a great way to promote the work you have done and will look great on your resume!

We will use Github Desktop, available here: https://desktop.github.com/

Note: you do not need to use Github Desktop in this class, but you will need to be able to fork and modify one of my repositories, and submit a Pull Request.

Exercise #1

1. Create a Github (https://github.com/) account if you do not already have one

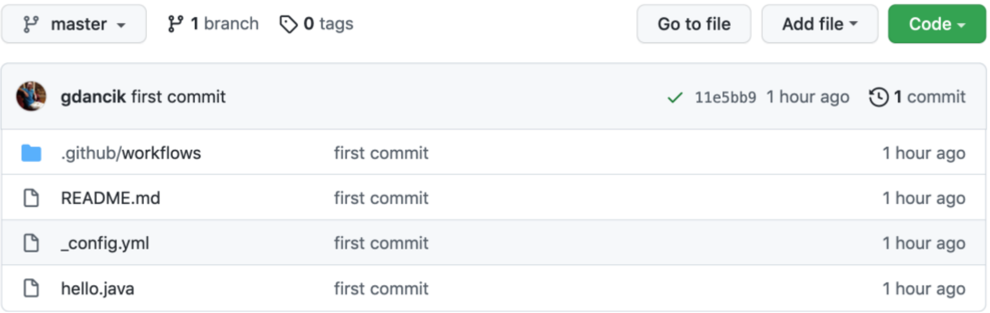
2. Download and set up Github Desktop: https://desktop.github.com/

3. Fork my *hello_world* program, available here, by clicking the *Fork* button on the top right of the page: https://github.com/gdancik/hello_world. This will create a new repository in your Github account with the name *hello_world*.

4. Open Github Desktop, select the *hello_world* repository, and click the ***Clone*** button, and specify the folder where *hello_world* should be cloned (copied) to.

5. Edit the main program (using a text editor of your choice) to change the value of *firstName* to your first name. Change the print statement so that the program outputs a greeting in the form, "Hello, my name is Garrett ", where the print statement outputs the value of *firstName*. Note that you may test your code locally using an IDE such as Eclipse or can run your code through an online compiler such as https://www.tutorialspoint.com/compile_java_online.php

6. When you are happy with your changes, *commit* them and then *push* them to your repository. On your repository's page, click on Actions and enable the *Run hello.java* workflow. This will allow you to launch a Github Action that will run compile and run the *hello.java* program. If the program runs without any errors, you will see a checkmark by the commit, as shown below (Note: this only checks that the program successfully runs, not that the output follows the format in question (5)).



gdancik first commit ✓ 11e5bb9 1 hour ago 1 commit

.github/workflows	first commit	1 hour ago
README.md	first commit	1 hour ago
_config.yml	first commit	1 hour ago
hello.java	first commit	1 hour ago

7. Submit a pull request by clicking on the appropriate links from the *hello_world* repository on your Github page. Note: you can check whether you have successfully submitted a Pull Request by looking for your repository at https://github.com/gdancik/hello_world/pulls. If you do not see your pull request here, you have not successfully completed the assignment!

8. For full credit on this assignment, you must (1) correctly make changes to the code, (2) run the Github Action to test the code on your repository, and (3) submit a Pull Request. You will not receive any credit if you do not submit the Pull Request.

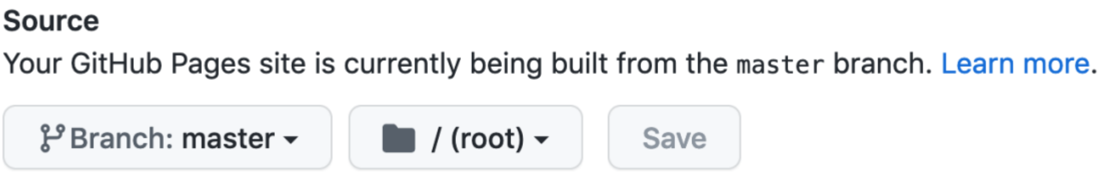
Exercise #2

1. Create a repository for your Senior Research project by doing the following (you can skip this step if you already have one). Open Github Desktop, and from the "Let's get started page" click "Create a New Repository on Your Hard Drive". Check "Initialize this repository with a README" and then Click "Create Repository".

2. Edit your README.md file so that it describes your project. Your description must include at least the objective of your project. Note that the README file is a *markdown* file that is displayed in a formatted way. For example, using a single hashtag (# Heading) will display a large header. A markdown cheatsheet is available here: https://www.markdownguide.org/cheat-sheet/. To see how your README file will be displayed, you can copy and paste the text of your README file here: https://dillinger.io/

3. Commit and push your changes to update your project repository on Github.

4. Create a web page for your project repository by following the instructions for the "*Project site*" at the following link: https://pages.github.com/ (make sure to click on "Project site" near the middle of the page). When creating your web page on Github pages under Settings, select Branch: master under source, so that the page will be built from the master branch, as shown in the screenshot below. If you do this, then the page will generated based on the README file from your repository. Alternatively, you can use gh-pages as the source, but your page needs to include a description and objective of your project.



Source

Your GitHub Pages site is currently being built from the `master` branch. Learn more.

5. When completed, submit through Blackboard the **two** links for your project page and for your Github repository page. Your links will have the format below (these assume that the name of the repository is *senior_research*):

   a. your project web page (the URL will be similar to http://gdancik.github.io/senior_research/)

   b. your Github repository page (the URL will be similar to http://www.github.com/gdancik/senior_research)