Git/Github Lab

Objectives:

- Becoming familiar with Git and Github
- Create an account
- Add repository and make contributions
- Create a Git Page for a project

Activities

After instructions in the class, you must be familiar with most concepts in this lab activities. If not clear try to search it up yourself and if still not successful then seek help. An important skill of a developer is to find solutions by searching and try and error.

We'll work in pairs: User A and User B

First make sure you have created your accounts and have logged in.

1. User A

- Set up a new repository, locally (let's say in a folder called "TestRepo")

```
mkdir TestRepo

cd TestRepo

git init
```

- Create or copy over a file or two and add them to the repository.

(create a ReadMe.md file) [edit the file]

```
git add ReadMe.md

git commit
```

- Go to your GitHub account and create a new repository
- Connect your local repository to GitHub

```
git remote add origin https://github.com/userA/TestRepo
```

Push your local repository to GitHub:

```
git push -u origin master
```

2. User B

- Fork user A's repository on GitHub: go to http://github.com/userA/TestRepo and click the "Fork" button.
- Clone your version of that repository locally

```
git clone https://github.com/userB/TestRepo
```

- Change a file, and add another file

  ```
  cd TestRepo
  [change/copy files]
  git add [filenames]
  git commit
  ```

- Push the changes to GitHub

  ```
  Git push
  ```

- Make a pull request:
  - Go to your repository on GitHub (http://github.com/userB/TestRepo)
  - Click “Pull requests”
  - Click “New pull request”
  - Click “Create pull request”
  - Optionally add a comment
  - Click “Create pull request”

3. User A

- Connect to User B’s repository

  ```
  git remote add userB git://github.com/userB/TestRepo
  ```

- Fetch the changes from User B

  ```
  git fetch userB
  ```

- Checkout their version of the repository as a local branch

  ```
  git checkout -b userB userB/master
  ```

- Check that you like the changes
- Use git branch to see your branches; the asterisk indicates the one you’re currently on.
- Switch back to (“checkout”) your master branch

  ```
  git checkout master
  ```

- Note that the files are in the state that you left them.
- Merge their work into your master branch.

  ```
  git merge userB
  ```

- Push the work to github.

  ```
  Git push
  ```

    - Make another change to the file; then add, commit, and push.

4. User B

    - Add a connection to User A’s repository

      ```
      git remote add userA git://github.com/userA/TestRepo
      ```

    - Fetch User A’s latest

      ```
      git fetch userA
      ```

    - Check it out as a local branch

      ```
      git checkout -b userA userA/master
      ```

- Test things
- Checkout your master, merge the change from User A, and push to github.

  ```
  git checkout master
  git merge userA
  git push
  ```

5. Users A and B

    - Make simulateneous changes, then add, commit, and push.

6. User B

    - Pull User A’s change

      ```
      git checkout userA
      git pull userA master
      ```

    - Go back to your master branch and merge the change from User A.

      ```
      git checkout master
      git merge userA
      ```

    - Fix the merge conflict; then add, commit, push.
    - Make another pull request

7. User A

- Fetch User B’s repository

  ```
  git checkout userB
  git pull userB master
  ```

- Merge into your master branch

  ```
  git checkout master
  git merge userB
  ```

- Push back to github

  ```
  git push
  ```

8. Both Users

Create a project page on the web

- Create a simple html and css files (or download from somewhere)
- Make sure main page is index.html
- Add it to your repo
- Publish it and provide a link
- You can pull a simple site from (https://github.com/Douglas1280W21/Project1-PersonalPage.git) try to modify it.