# Collaborating With GitHub v2.0 - GW Libraries Workshop

## **Part 1: Create your own repository**: Two Ways

Starting from github:

- On github.com, create a repository, call it "hello-world"
- In GitHub Desktop, go to File → Clone Repository, and clone hello-world -- pay attention to the Local Path you've chosen.

OR:

Starting from existing local files:

- Create some files (and optionally, files in subfolders)
- In GitHub Desktop, go to File → Add a New Repository.
- Give the repository a name, description, etc.
- For the Local Path, select the folder where you created the files.
- In GitHub Desktop, press Publish Repository
- Observe that your repository is now on GitHub.

## **Collaboration: Issues, branches, pull requests**

- Create a repository, call it "hello-world"
- Create an **issue** to "Add a new story feature"
- Create a **branch** called "`some-feature`". Work on the branch:
    - Add a "`story.txt`" file in the `some-feature` branch (using the Create New File button)
    - Edit the file (add some text), then make sure to write a commit message that includes "Fixes #1". Commit the file.
    - Look at the network graph (under Insights → Network)
    - Look at the commit history
- Open a new pull request, select the `some-feature` branch
    - View the changes on the Compare page, then press Create Pull Request.
    - Give the pull request a title and description
- Merge your pull request
    - Look at the network graph
    - Look at the commit history
    - Look at your issues list
    - Delete the `some-feature` branch
- Look at your profile

## **Part 2: Collaborate on a repository**, AND use git at the command line.

We'll be using https://github.com/kerchner/git-workshop for this section.

- Send me your git username so I can invite you to be a collaborator.
- Accept the invitation.
- I'll create some issues and assign them to people. (I'll also add them to a milestone)
- Connect to the server (see instructions!)
- Some quick git setup - use the email that you used to register your github account:

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

- Clone the repository - copy from the "Clone or Download" button on github, and type:

```
git clone https://github.com/kerchner/git-workshop
```

- Let's see what we have:

```
ls -al                  (ls means "list directory contents")
cd git-workshop         (cd means "change directory")
ls -al
git status
```

- Make a branch and check out that branch:

```
git branch name-of-my-branch
git checkout name-of-my-branch
```

- Create and edit a myname.txt file (e.g. dan.txt)
  Use your editor of choice (nano is just one)

```
nano some-file-name.txt
```

- What does git make of this new file?

```
git status
```

- Add the new file

```
git add some-file-name.txt
```

  The change is now in a "staging area."
- Now what does git status say?

```
git status
```

- Let's also look at

```
git diff
git diff --staged
```

- Commit the file. Find the issue assigned to you and use that number (instead of 30)

```
git commit -m 'Add some comment here. Fixes #30'
```

- Now what does git status say?

```
git status
```

- Push this branch to the repository on github:

```
git push origin your-branch-name
```

- View the log of changes in this branch:

```
git log
```

  and in the master branch:

```
git checkout master
git log
```

You can compare against the commit history for each branch up on GitHub.

- On GitHub, create a new pull request for your branch.
- Take someone else's pull request and merge it.
- Now update your local repo with the changes that have been merged

```
git pull origin master
```

and note that you have the latest changes

```
git log
```

- Let's now look together at the commit history and some of the Graphs.

## **Part 3: Fork a repository** and contribute a pull request

- Fork the repository **https://github.com/gwu-libraries/git-sandbox**
- Add or modify a file in your fork of **git-sandbox**. For now, do this in the "master" branch.
- Create a pull request. This will be a pull request against the repository you branched from.
- Look at the Network graph for the main repo, under the Insights tab.
- I'll merge your pull requests.