

# 1. Create a repository, make changes

GITHUB DESKTOP APP	COMMAND LINE
<b>Install / Set up tools</b>	
Download from <a href="https://desktop.github.com">desktop.github.com</a> Under "Preferences..." sign in to your GitHub account.	Install Git command line via <a href="https://git-scm.com/downloads">git-scm.com/downloads</a>  <pre>git config --global user.email "you@youremail.com" git config --global user.name "Your Name"</pre>
<b>Create a repository</b>	
(1) Create some files in a folder on your computer	
(2) File → Add New Repository. Include a Readme and a License. (3) Publish Repository	(2) Create an empty repository on GitHub. Include a Readme and a License. (3) <pre>cd &lt;folder where your files are&gt; git init git add &lt;my file(s)&gt; git commit -m 'initial commit' git remote add origin https://github.com/&lt;me&gt;/&lt;repo name&gt;.git git push -u origin master</pre>
<b>Modify files</b>	
(1) Add new files, edit existing files, remove files	
(2) Observe the "Changes" tab. See the changes in each file.	(2) <pre>git status git diff</pre>
<b>Stage and commit changes</b>	
(1) Check the boxes next to the files you want to stage. (2) Add a brief comment (3) Press Commit to master (4) View the changes on GitHub	(1) <pre>git add &lt;changed files&gt;</pre> (2) <pre>git commit -m 'Describe this set of changes here'</pre> (3) <pre>git push origin master</pre> (4) View the changes on GitHub
<b>View the commit history on Github</b>	

## 2. Branching and Collaborating

GITHUB DESKTOP APP	COMMAND LINE
One Person: On Github, create a repository. Add collaborators (in Settings → Collaborators)	
<b>Everyone: Clone the repository:</b>	
File → Clone Repository (by URL)	<code>git clone https://github.com/&lt;user&gt;/&lt;repo name&gt;.git</code>
<b>Create some issues</b>	
<b>Assign issues to people</b>	
<b>Each person: Create a branch</b>	
Switch branch to New Branch	<code>git branch &lt;name of the new branch&gt;</code> <code>git checkout &lt;name of the new branch&gt;</code>
<b>Make changes to files to fix an issue.</b>	
<b>Check in your changes to the branch:</b>	
(1) Check the boxes next to the files you want to stage. (2) Add a brief comment, for example "Adds title, axis labels to plot. Fixes #1" (3) Press Commit to <name of your branch> (4) View the changes on GitHub	(1) <code>git add &lt;changed files&gt;</code> (2) <code>git commit -m 'Adds title, axis labels to plot. Fixes #1'</code> (3) <code>git push origin &lt;name of your branch&gt;</code> (4) View the changes on GitHub
On GitHub, <b>create a new Pull Request</b> for the new branch. Assign the pull request to a collaborator. To review the pull request assigned to you:	
(1) Press "Fetch Origin" (2) Switch to the branch you're reviewing	(1) <code>git fetch origin</code> (2) <code>git checkout -b &lt;branch name&gt; origin/&lt;branch name&gt;</code>
If everything looks good to you, press the green "Merge Pull Request" button (if it's not green, you have merge conflicts!) Now you can delete the branch (optional)	
On Github, view <b>Insights</b> → <b>Network</b> for a visual representation of what happened	

### 3. Make a Release of your Software

- On Github, in the Code tab, click on "0 releases"
- Create a new release from the master branch.
- Call the version 1.0 and write something for the release notes.
- Publish your release.

Tip: If you're going to publish your code on Zenodo.org, then Zenodo will need a code release to ingest your GitHub repository!

### 4. Fork a Repository

- Let's use <https://github.com/gwu-libraries/git-sandbox>
- On the repository's page, press the Fork button.
- Go to your own repositories - notice that you now have a fork of **git-sandbox**.
- Create a branch.
- On your branch, modify some file(s).
- Create a pull request with this branch.