

First Handout - GitHub Workflow

For this project, you and your partner(s) will be creating a code repository that is a copy of the main Indian Country repository. Since everyone in the group will be working on the same repository, somebody will have to create and be the owner of that repository. This person will be referred to as Partner A in this worksheet, and will have a few more responsibilities that Partner B(s) will be unable to perform. But don't worry, everyone will have a chance to code! Tasks specific to Partner A will have an "(A)" in front of them.

BEFORE ANYTHING - Make a folder on their desktop for their repository! This is very important, you can save all of their information and data there and will not have to go searching for it

Part A - GitHub.com

1. Go to GitHub.com
2. Search for **indian-country-ss24** in the search bar at the top left
3. This should bring you to our master repo:
msu-anthropology/indian-country-ss24
 - a. Explain:
 - i. <Code>: The main page of your repo
 - ii. <Wiki>: A Wiki for this project is available in this tab
 - iii. This is a convenient place to start troubleshooting.
 - *[Troubleshoot: The Wiki for this project will only be available on the main msu-anthropology repository, and not your individually created repos]*
4. Fork your repo (**Decide who is partner A and who is(are) partner B**). Partner A will be the owner of the repository, which means they'll be responsible for a few tasks Partner B will be unable to perform such as the Settings, Pull Requests, etc.
 - a. (A) In the top right hand corner of the repo, press the Fork button
 - b. This should create a new forked repo named:
“(A)username/indian-country-ss24”
 - i. Underneath it will state: forked from
msu-anthropology/indian-country-ss24
 - *Tip: You can click on this to to get back to the original repo*
5. (A) Add a collaborator (Give your group members permission to work in the repo)
 - a. (A) Go to <Settings>
 - i. *[Remember, only partner A will have access to settings]*
 - b. (A) Go to the <collaborator> tab to the left of the page
 - c. (A) Click the green <Invite teams or people> button
 - d. (A) Search for your teammate's GitHub username and add them
 - e. (B) Partner B(s) should get a notification in their email (check your junk folder) asking them to join the repo - accept the notification

- f. Now all teammates are able work with the repository
 - i. (A) Partner A will be able to tell if the link was accepted by checking the <collaborator> tab (refresh the page)
6. (A) The repository owner will then set up *GitHub Pages*.
 - a. Go to <Settings>
 - b. Scroll down to the “GitHub Pages:” section
 - c. Change the **source** to **main branch** (not **gh-pages**)
 - d. This should automatically save & create a link to the website for the forked repository you have created (*takes a while to update*).
7. (A & B) Both Partners Clone the new repository owned by Partner A to their local computer
 - a. Go to the <Code> tab
 - b. Choose **Open with GitHub desktop**, not ZIP

Part B - Github Desktop (A & B)

1. Go to your GitHub Desktop Application
 - a. Make sure that students save their repositories into the folder that they made for this project
 - b. Click the “Parent Project” option in the pop-up box
2. Things to explain:
 - a. **Fetch origin** – clicking makes sure that you’re working in the most current form of your forked repo – good for avoiding conflicts
 - b. **Changes & History** lets you keep track of progress in your repository
3. Click **Current Repository**:
 - a. Unless you have worked in GitHub before, only one repository should show up: **indian-country-ss24**
4. Make sure VS Code is the default text editor
 - a. **PC**: File -> Options -> Advanced -> External Editor set to Visual Studio Code
 - b. **MAC**: Github Desktop -> Preferences -> Integrations-> External Editor set to Visual Studio Code
 - c. Save
5. There’s a button titled “Open in Visual Studio Code” that you can use to open your text editor

Part C - Workflow in VS Code

Now that each of you have a copy of the repository on your computer, you can start making changes to the project. These initial steps could be completed by either partner, but to get used to things, only one person should be making changes at a time.

1. Starting in VS Code
 - a. Notice that the folders and files are the same here as on GitHub.com
 - b. Now each partner will take turns changing the repository
2. Under the Project tab, in the **sites** folder.

- a. (A) Partner A will start by creating the project folder:
 - i. Right click the **AA-template folder** – select **Duplicate**
 - **Do not delete or rename this folder!**
 - ii. Give the new folder the name of your project (can change later)
 - Right click on the new folder, select rename
 - **No spaces or capital letters**
 - E.g. Tel el-Amarna = tel-el-amarna
 - Must press enter after typing in name
 - iii. Click on your new “*project-name*” folder to open it
 - Right click on **AA-template.html** and rename it to *project-name.html* (the same as the folder name)
 - Should now have two items within *project-name* folder:
 - **img** (folder) & *project-name.html*
 - The **img folder** has a placeholder image in it.
 - The **video folder** has a placeholder video in it as well.
 - iv. Now push these changes using the following directions:
 - open up **GitHub Desktop**
 - **Fetch & Pull** any changes - There shouldn't be any since this is the first change to the repo, but it is a good habit to get into. (**Pull** will not appear this time with no changes to pull).
 - Enter a commit message like “creating project folders”
 - click the blue **Commit to Main** button
 - click the **Push** button that appears in the **GitHub Desktop**
 - *[troubleshoot: Try restarting VS Code & GH Desktop. Put a red sticky note up if this doesn't work.]*
- b. (B) Partner B - editing *project-name.html*
 - i. In order to edit the folder and files created by partner A, you will have to *pull* what they did onto your computer:
 - Click the **Fetch** button located at the bottom-right
 - If the changes from before were pushed correctly, then the Fetch button should change to **Pull** this time. Click **Pull**.
 - ii. Now, navigate to your newly appeared *project-name* folder, and click on *project-name.html*.
 - Notice the <body> section that begins at line 22
 - In the <body> section, you will change the text (italicized here) in between the following tags with any new information you know (at the very least you can change the authors):
 - <h1>*Subject Title*</h1>
 - <h3>*Community Name*</h3>
 - <h4>*Author(s)*</h4>
 - iii. Partner B will also push their changes in the **Git** tab:
 - **SAVE!** - git will not notice your changes unless you save them!
 - **Fetch & Pull** any changes - still a good habit. **Pull** should not appear since you've already pulled changes.

- In the future, it's a good idea to coordinate what sections you are working on and when changes are being submitted in order to avoid conflicts!
 - Enter a commit message like "editing project HTML"
 - click the blue **Commit to Main** button
 - click the **Push** button that appears at the bottom
3. **(A & B)** You can check to see if your repository is updated by checking the code on github.com, checking out the GitHub Pages link you made, or opening your HTML file on your computer.
 - a. Fill in the link (*Replace "project-name" twice and don't forget ".html"*):
[https://username\(A\).github.io/indian-country-ss20/sites/project-name/project-name.html](https://username(A).github.io/indian-country-ss20/sites/project-name/project-name.html)
 - i. You can get most of the link by going to the online GitHub.com Repository you are working on. Just add:
sites/project-name/project-name.html.
 - b. **You can also open your project's html using your web browser.** We recommend checking most of your work using this method.
 - i. The file should be in your documents folder -> GitHub -> Indian Country -> projects folder -> "*project-name*" folder -> *project-name.html*
 - ii. You should be able to right click and open this file using your standard browser. This will usually allow you to see local, SAVED changes made to the file immediately.
 4. You can see the new changes from Partner B are reflected on the GitHub.com repository, but Partner A will still need to **fetch** and **pull** the changes made by partner B to see the changes in their local repository. Partner A should do that now.

Part D - Create a codecademy.com Account (Everybody)

Now that you have a better understanding of the GitHub workflow, we can shift gears to learning how to write in HTML

1. Create an account for codecademy.com
2. Search for the following courses in codecademy.com: **Learn HTML** and **Learn CSS**
 - a. Run through the first lesson (16 exercises) of the HTML course as well as the first lesson (8 exercises) of the CSS course
 - i. Feel free to continue the course if you want to learn more!