

Context

1. Links you might need
 - a. [+ Computing in Context Resources](#)
 - b. [+ Computing in Context - Madelyn](#)
 - c. [Computing in Context GitHub Repo](#)
 - i. The .ipynbs will also exist in the Google Drive folder
2. At the time of requesting these resources from schools, the CIC defined computing in context in 2 ways:
 - a. integrating different disciplines or domains into computing curriculum (e.g., applying computing techniques to topics in the Humanities discipline), and/or
 - b. regularly updating curriculum and assignments to cover topics relevant to today's students (e.g., analyzing Twitter data to evaluate algorithmic bias in an election year)
3. We requested assignments and lectures from our partner schools that have a course that meets these definitions. The records of this is found on the first tab ('sources') of [+ Computing in Context Resources](#).
4. The 'resources' tab of this document is where you note specific information for *each* resource we received. Below the line (row 375) are resources that have not yet been analyzed. These might be links to individual assignments, they might lead to a whole class's worth of assignments. Each specific resource/file should be analyzed independently on its own row. Check past entries in the first 2 columns to know what kinds of things you should be looking for.
5. The entries in the column 'Resource Link' are the URLs to the *original* resources. We need to duplicate all of these in order to put them on our website and control the access permissions.
6. The 'download_resources' tab takes information from the 'resources' tab up until row 374 for easy download of necessary information to test the scripts that duplicate the resources. The formula in the very first cell of this tab will need to be updated as you analyze more resources.
7. In order to upload the resources to the CIC website, you need a WordPress account (check with Caitlin to set this up). The unpublished page can be found in Pages > Computing in Context Repository and the actual data table used for this page is in wpDataTables > Computing in Context Resources.
8. The new URLs (that link to our duplicates of the resources) must be in the format link||assignment_name (Ex. <https://colab.research.google.com/drive/1UoHhUMvAmP9MLsu5Cw1PIVBODEZxqZTX> ||Calculate the Cost of a Purchase) in order to be presented correctly on the website.
 - a. The assignment name is typically just the original title, but should be changed if it could be unclear to someone. This is what appears on the actual website table, so it should not be too long.
9. How the duplication scripts work:
 - a. From the CSV from the 'download_resources' tab,

- b. makes copies of each resource in a Google Drive folder you own,
 - c. stores the URL of the duplicated resource in the link||assignment_name format, along with all of the other recorded information for each resource in a dataframe,
 - d. saves the dataframe to an Excel file that can be directly uploaded to the CIC website through wpDataTables.
 - e. Once you are *done* duplicating resources, the folder you made should be moved into the CIC Google Drive. You cannot duplicate directly into the CIC Google Drive (due to access rules), which is why the folder must be associated with your work email's own drive initially. The excel file that you uploaded to WordPress should also live in the CIC Drive.
 - f. I finished a script for duplicating files given to us in a Google Colab format (typically single .ipynbs).
 - g. A lot of files are on GitHub repositories; the duplication of these is different from Colabs. One thing that makes these more complicated is the data sources. The links in [📄 Computing in Context Resources](#) are to a specific file on a GitHub repo, but a lot of them use data files that exist somewhere else in the repo. We need these data files in order for the resource to be useful to someone.
10. While testing if my scripts worked, I had a test Google Drive folder that I would continuously duplicate files into and delete them until I was confident that my script functioned as I desired.

Steps for continuing the project

Finish Analyzing Resources

1. Does not necessarily need to be done first.
2. If this is done after the scripts for duplicating Colab and Git files are run, whatever has not already gone through one of these scripts will need to. The results will then have to be entered into the .xlsx that gets uploaded to the website.
3. Keep track of which files you have already duplicated and exist in the .xlsx.
4. If you wait to finish analyzing resources after you move your folder into the CIC Drive, each additional resource will have to be moved into this location as well.

Colab Files

1. Obtain a .json file for your [client secret](#):
 - a. Visit the [Google API Console](#)
 - b. Navigate to the 'Credentials' tab and press '+ CREATE CREDENTIALS' to create an OAuth client ID
 - c. Save the client secret to the directory of your duplication scripts
2. In the google drive associated with your work email (not the CIC drive), create a folder. Save the folder ID to actual_folder_destination in the second cell of duplicate_files_COLAB.ipynb. The folder ID is the combination of numbers, letters, and symbols located after 'folders/' in the URL of the folder. Ex: if the link to the folder is

<https://drive.google.com/drive/u/2/folders/18w2oH-f-Lr6iDK1B7XQ6YvNilzpe1GTx>, then the folder ID is 18w2oH-f-Lr6iDK1B7XQ6YvNilzpe1GTx.

3. Run `duplicate_files_COLAB.ipynb` with a CSV from the 'download_resources' tab that only contains the Colab files (currently the first 297 rows).
4. Open `Computing_in_Context_Resources_COLAB.xlsx`
5. For each file that needs to be uploaded manually,
 - a. Open the link in your browser
 - b. File > Save copy in drive
 - c. Rename the file (delete 'Copy of')
 - d. File > Move > your resource folder
 - e. The resource information must be manually entered into `Computing_in_Context_Resources_COLAB.xlsx`. The value in the Resource column should be in the format `link||file name` (Ex. <https://colab.research.google.com/drive/1UoHhUMvAmP9MLsu5Cw1PIVBODEZxqZTX>||Calculate the Cost of a Purchase)

Git Repos

1. The script to duplicate Git files follows a similar thought process as the one for Colab files, but is not successful. The location of the data files makes it more complicated. The last 3 columns in the 'download_resources' tab are associated with my first attempt at duplicating git files.
2. How my first attempt works:
 - a. Sometimes, the data files for an assignment are stored in a folder that also contains the assignment. If this is the case, put the link to the folder in 'folder URL' and leave 'repo URL' blank
 - b. Other times, the common meeting point of the data and the assignment is the repo itself. If this is the case, put the link to the repo in 'repo URL' and leave 'folder URL' blank
 - c. Input the name of the assignment to 'file name' regardless of if a folder or repo URL was used
 - d. The script attempts to clone each assignment's repository, isolate the folder if it exists, and duplicate the assignments from there
3. ***I do not know if this is the best way to complete this task***, but if you decide to continue with this method, the last columns I:K will need to be filled out for all GitHub resources.
4. Whatever method you use will need to generate an excel file in the same format as `Computing_in_Context_Resources_COLAB.xlsx`. Since `wpDataTables` only allows one input file, the resulting .xlsx from the GitHub file duplication will need to be merged with `Computing_in_Context_Resources_COLAB.xlsx`.
 - a. Either copy and paste results into `Computing_in_Context_Resources_COLAB.xlsx`,
 - b. Or have a single script for both Colab and Git files that works off of the same dataframe and exports it to an Excel file at the end.
 - c. The first method will be easier for testing.