Formal Language Update (May 13th 2018)

This is the most updated document as of May 13th 2018 for OpenDSA formal language material. Before reading this you must be familiar with the documentation written by Duke student that describe all files and usage of this tool.

Pre-Requisite:
before reading this document, you must read and understand the formal language documentation. You should also download, and get familiar with JFLAP.

NewFiles library addition:

- Html2canvas – this file allows the application to capture and save the machine as a jpeg file.

Current Directory Structure:
The following files listed bellow are the core libraries of the formal language materials. They do not reference any other files in their code and are referenced by almost all other files to implement algorithms and designs.

- Commands.js
- CustomPrompt.css
- CustomPrompt.js
- FA.css
- FA.js
- serializableGraph.js
- TraverseAccepter.js
- TraverseTransducer.js
- underscore-min.js
- html2canvas.js

These files and all other future files that will not be referenced by any other files within formal language should be moved into "OpenDSA/DataStructures/FL_resources" directory until the end of development of materials.
All other files that will reference these files should be located within their respective project folder.

Next Step:
The next step in the development will depend on the need of the formal language class. However, up till this update, the next logical step is to modify view for "Minimize DFA", "Convert to RLG", and "Convert To RE" menu option to mimic JFLAP. Then after, tie/implement algorithm to their respective view elements. Note that there are multiple algorithm already implemented that are worth taking look at before coding.

I strongly recommend to keep the FLA directory, within AV, intact until the end of the development of the materials. This directory contain sub directory that have multiple files in common. I do not encourage deleting those files as well. The linking, within codes, to those files makes it difficult to use one instance of the file (unless you create create a common directory of course). Also be careful, just because files have the same name; it does not mean they have the same content. Assuming this directory is kept alive, developer can always go back to reflect on the old view or copy and paste code for old functionality that might have been deleted.

Development Guide:

If developer is new or unfamiliar with formal language material, it will be extremely beneficial to download and get familiar with JFLAP. Try running examples provided with the jar file to explore all the provided functionality. Then play with the latest version of this applications and compare it to JFLAP. Read all the documentation for this material before starting to code.

During coding, it is useful to use the debug functionality of your browser to debug your application. Developer can make live code and style changes withing the browser during development. This is extremely useful as it saves a lot of time. You can also set up break point to catch functionality event. You can find multiple videos about different browser debugging on youtube.