

Creating a Jupyter Book, based on Callysto Curriculum Notebook.

By Michael Lamoureux - November 17, 2020

Introduction.

Converting a bunch of Jupyter Notebooks into a Jupyter Book is a three step process:

1. Create a local copy of the JBook on your PC, using the JBook tools
2. Publish on the web in a new repo on Github
3. View your book online

Here is a more detailed explanation, for a simple JBook we created at a hackathon in October. Here we show the terminal commands (Linux) used to do all the steps. We show the directories explicitly, as it seems to matter **where** you run the command. We are doing this on a NUC Linux machine running Mint, in the Documents directory.

Step 1. Create a new book (as a directory), clone the jupyter notebooks into it, create the table of contents, and build

```
NUC:~/Documents$ jb create FEC-jbook
NUC:~/Documents$ cd FEC-jbook/
NUC:~/Documents/FEC-jbook$ git clone https://github.com/kcdoo/pac.git
NUC:~/Documents/FEC-jbook$ cd ..
NUC:~/Documents$ jb toc FEC-jbook/
NUC:~/Documents$ jb build FEC-jbook/
```

-- at this point, you can read the book locally. You can also edit various pieces to make it look nice.

Step 2. Publish on the web.

-- First I go to github to recreate a private, empty repo. Next I clone it, copy files into it, and use ghp-import to push everything back to github, where it automatically gets published.

```
NUC:~/Documents$ git clone https://github.com/callysto/FEC-jbook-online.git
NUC:~/Documents$ cp -r FEC-jbook/* FEC-jbook-online/
NUC:~/Documents$ cd FEC-jbook-online/
NUC:~/Documents/FEC-jbook-online$ git add .*
NUC:~/Documents/FEC-jbook-online$ git commit -m "First load of the book"
NUC:~/Documents/FEC-jbook-online$ git push
NUC:~/Documents/FEC-jbook-online$ ghp-import -n -p -f
~/Documents/FEC-jbook-online/_build/html/
```

Step 3. go to github and click on the gitpages activity icon to see the book.

In this case the book is here: <https://callysto.github.io/FEC-jbook-online>

Setting up your environment

You will need to install and update some tools on your unix system. I did these updates recently:

```
pip install -U pip (gave me pip-20.2.3)
pip install -U jupyter-book (gave me jupyter-book-0.8.1)
pip install -U nbformat (gave me nbformat-5.0.7)
pip install -U sphinx-togglebutton (gave sphinx-togglebutton-0.2.2)
```

I did have an error earlier on running the command jupyter-book, having to do with secure_write not importing. If you have the same problem, you might try this solution I found online, which involved running this command:

```
pip install -U jupyter_client
```

You may also need to pip install some Python packages, if the Jupyter Notebooks you are running require them. (Check the error messages when you do a build.)

Problems with the Curriculum Notebooks, that need tweaking

The Callysto curriculum notebooks were written by many authors, over a period of a few years. So the styles are quirky, and each one needs to be examined individually to fix anything that might block the functioning of JBook. Some common problems:

1. HTML formatting in Markdown cells -- this seems to mess up JBook, so avoid anything like centerline, bold, left, right alignment commands.
2. Header levels, indicated by #, ##, ### should go in sequence. So if one section is headed with a ##two-level-header, its sub-sections should be a ###three-level-header
3. Only one #top-level-header should be used. And this will become the title of that chapter in the Jupyter book. (So make the title short and informative.)
4. Latex equations in display mode should be separated by a blank line above and below. I don't know why, this seems to be a bug in JBook.
5. Latex equations require a line in the _config.yml file (see below)
6. Our method of hiding/showing code is incompatible with JBook. So, to hide a code cell, you need to edit the cell metadata and include "tags": ["hide-input"] (see below for more notes)
7. Notebooks that pause execution while waiting for user input will interrupt the JBook creation. Best to use text widgets for input, or something that doesn't pause execution. However, it is also possible to set up the _config.yml so that the JBook does not try to re-run the notebook.

8. To block execution of a notebook when building the Jupyter Book, you can either list it in the `_config.yml` file as a notebook to not execute, or you can tell the config file to not execute any of the notebooks. (The default behaviour is to run the notebook anytime there are changes. This can be slow, and can mess up your hard work. I decided in the end not to execute any notebooks. You can see how in the `_config.yml` file shown at the end of this document.)
9. Geogebra apps need to be hosted on the Geogebra website, and loaded from there as part of the code. Locally hosted Geogebra apps do not seem to work in JBook. I had to remake several Geogebra apps and host them online to get them to work.
10. Helper files (images, data, python code, etc.) that are stored in the notebook folder don't seem to get copied over to the JBook directories. So you will need to copy them over yourself. Unfortunately, lots of our notebooks use these helper files.
11. Some notebook files create too much output. For instance, our Shakespeare notebook prints out ALL of Macbeth. It is better to limit outputs in all cells (e.g. just print the first 100 lines of Macbeth.)
12. To get widgets, YouTube and Audio files to work, you need to add a line to the `_confil.yml` file (see below)
13. D3 animations are flakey. I tried several methods to get them to work, sometimes I was successful. I am not sure that I can explain what methods work.
14. Note that your browser will cache old versions of the JBook, so when you are making changes to the JBook, the browser may be in some half-assed state that doesn't always reflect your changes. You should flush the browser cache.

Detailed record: Creating a Jupyter Book from Callysto curriculum notebooks.

These are my notes on creating the Jupyter Book, step-by-step, including problems and some fixes.

1. Create a new folder on the Linux box:

```
:~/Documents$ jb create curriculum-jbook-local
```

2. cd to that new directory, and GIT the repo

```
:~/Documents/curriculum-book-local$ git clone  
https://github.com/callysto/curriculum-notebooks.git
```

3. cd to that directory, and create a new branch

```
:~/Documents/curriculum-book-local/curriculum-notebooks$ git checkout -b jbook
```

4. Add a markdown file to each folder that has subfolders, so we get the TOC to be created correctly. I put in a short .md file in each of these folders:

- Arts
- EnglishLanguageArts
- Health
- Language
- Mathematics
- Science
- SocialStudies
- TechnologyStudies

The subfolders don't seem to need anything.

5. Create a Table of Contents:

```
:~/Documents$ jb toc curriculum-jbook-local/
```

6. Edit the TOC to get rid of references to unwanted files.

7. Use the "sort" function to order the items in the TOC, alphabetically (by section)

8. Remove the notebook "notebooks-button-creator.ipynb" as it will mess up our other files!!
DANGEROUS!!

9. Checked the notebook "Science/Climatograph" to ensure it runs. Last time it crashed (HTTP Error 502: Bad Gateway). This time it seems to work fine.

10. Fixed Languages/French-Verb-Coding.ipynb because it calls StdIn, which halts the JB build. Replaced it with a TextWidget

11. The file "TechnologyStudies/IntroductionToPython/introduction-to-python.ipynb" uses raw input and will kill JB build.

So I edited the _config.yml file to exclude execution on this one. Also electrical-conductivity. This is the added lines to the _config.yml file:

```
execute:  
  exclude_patterns:  
    - 'introduction-to-python.ipynb'  
    - 'electrical-conductivity.ipynb'
```

12. Updated _config.yml to proper author, title, logo. Include the Callysto logo file in the folder.

13. Tried jb build. Had some errors:

- the _toc file had to be formatted correctly
- have to pip install cufflinks
- have to pip install folium

14. Had to copy all the subfolders like images/, data/ etc from the curriculum repo into the corresponding folders in the "_build" folder.

15. Problems:

- Science/Newton's First Law. The D3 animations fail. They work in Third Law notebook.
- Widgets like sliders and text boxes are active, but do not run any Python code as they should.
- English/Shakespeare - it prints out all of MacBeth. Yikes.
- Math.analyzing-radical-function Had to fix the display Latex and the MD list. Removed the html-style formatting.
- Math/appreciation-of-mathematics-in-society. Ditto.
- Math/binary-decimal-conversion. The inline apps only sort of work. Formatting issues. Might be fixable...
- Math/combined-log-law has wacky Latex (probably fixable, but it is in the code!)
- Math/counting-triangles The Geogebra items don't seem to work. Why? (They work in TriangleArea.) The Triangle Area uses a call to ggb.material('QgWDn3pt').draw(), which suggests the code is hidden somewhere on a Geogebra site. And that works! The counting triangle tries to load in a local ggb file, gb.file('sources/triangleAnimation.ggb').draw(), and that does NOT work.

UPDATES TO THE SOFTWARE

Did a bunch of pip installs

```
pip install -U pip (gave me pip-20.2.3)
pip install -U jupyter-book (gave me jupyter-book-0.8.1)
pip install -U nbformat (gave me nbformat-5.0.7)
pip install -U sphinx-togglebutton (gave sphinx-togglebutton-0.2.2)
```

GEOGEBRA FIXES

Trying to get Geogebra to imbed correctly.

This seems to work (in a code cell):

```
%%html
<iframe scrolling="no" title="TriangleAnimation"
src="https://www.geogebra.org/material/iframe/id/cwbxjw9c/width/716/height/272/border/888888
/sfsb/true/smb/false/stb/false/stbh/false/ai/false/asb/false/sri/false/rc/false/ld/false/sdz/false/ctl/false" width="716px" height="272px" style="border:0px;"> </iframe>
```

You get this from the Geogebra website, going to Details/Share. Then select Embed (as HTML). It will get the dimensions correct, but you can also edit it in the iframe above.

I had been using: `ggb.material('QgWDn3pt').draw()`
which sometimes works, and sometimes did not. I did not find a pattern to this.
We also had code like `ggb.file('sources/PentaAnimation.ggd').draw()` but that never seemed to work in the J-Book.

Notebooks with geogebra:
triangle-area (fixed)
CountingAreas (fixed)
BinaryDecimal conversion
Data and sinusoidal functions
Inductive reasoning and deductive reasoning
Parallelogram area
Reflections of graphs

HEADER FIXES

- only the title of the notebook should have a level 1 header (using one #)
- all other sections should have level or higher (use two ## or more)
- all done (many files needed fixing)

HIDING CODE.

- our Callysto method of toggling on and off the code does not work.
- We need to edit the cell metadata, looks like this:

```
{
  "scrolled": true,
  "tags": ["hide-input"],
  "trusted": true
}
```
- I think I have to do this for every code cell we want to hide.
- we can use "hide-cell" to hide both input and output. Useful for long outputs that the reader doesn't need.
- we could use "remove-input" if we don't want the reader to ever see the code.

CONSISTENT HIDE BUTTON

- use this code, for all the notebooks that hide code. Consistent with Jupyter book buttons.
- ```
%%html
```

```
<script>
function code_toggle() {
 if (code_shown){
 $('div.input').hide('500');
 $('#toggleButton').val('Show Code')
 } else {
 $('div.input').show('500');
```

```

 $('#toggleButton').val('Hide Code')
 }
 code_shown = !code_shown
}

$(document).ready(function(){
 code_shown=false;
 $('div.input').hide()
});
</script>
<p> Code is hidden for ease of viewing. Click the Show/Hide button to see. </>
<form action="javascript:code_toggle()"><input type="submit" id="toggleButton" value="Show
Code"></form>

```

## GETTING WIDGETS, YOUTUBE TO WORK.

- I don't know why I needed to do this, but JupyterBook .8 suggested I add this to the `_config.yml` file

```

sphinx:
 config:
 html_js_files:
 - https://cdnjs.cloudflare.com/ajax/libs/require.js/2.3.4/require.min.js

```

This seems to fix a lot of problems with text boxes, Audio widgets, even YouTube.

## LATEX EQUATION MODE

To get stuff like this to work: `\begin{equation}`, `\begin{align}` etc.

- Need to add the following to `_config.yml` file

```

parse:
 myst_extended_syntax: true

```

ALSO, LATEX. Display equations should be on three lines, like this. Leave spaces above and below

....

```

$$
\int_0^{\infty} f(x) dx = 0
$$

```

....

## PUSHING TO GITHUB FOR PUBLISHING

- I have another folder for this, called curriculum-jbook

- put everything in curriculum-jbook-local/\_build/html into this folder curriculum-jbook
- cd into curriculum-jbook
- git status
- git add \*
- git commit -m "current version or whatever"
- git push origin master
- will ask for my name and password
- it will take several minutes to upload, into <https://github.com/callysto/curriculum-jbook> (a private repo)

Now do this:

```
mikel@mikel-NUC:~/Documents/curriculum-jbook$ ghp-import -n -p -f
~/Documents/curriculum-jbook-local/_build/html
```

Note which directory I had to be in, in order to run this.

PUSHING TO GITHUB FOR PUBLISHING -- from scratch!

1. Create a new github repo on github/callysto. No README file. Private
2. Clone it onto my hard drive
  - git clone <https://github.com/callysto/curriculum-jb>
3. Copy the files from my local book into the newly cloned folder
  - cp -r curriculum-jbook-local/\* Curriculum-jb/
4. Push the content onto Github
  - cd Curriculum-jb
  - git add ./\*
  - git commit -m "trying the second book!"
  - git push

(This should take a few minutes, as it is uploading everything.)
5. From within this Curriculum-jb directory, run the ghp import command
  - ghp-import -n -p -f ~/Documents/curriculum-jb/\_build/html

(This should also take a few minutes.)
6. go to <https://callysto.github.io/curriculum-jb/intro.html> to see the new book

LET'S TRY PUSHING TO GITHUB ANOTHER WAY (also works)

Let's try again with the old callysto-jbook repo

1. clear out the folder curriculum-jbook
2. copy files to it
  - cp -r curriculum-jbook-local/\* curriculum-jbook
3. Push the content onto Github
  - cd curriculum-jbook
  - git add ./\*
  - git commit -m "trying the second book!"
  - git push

(This should take a few minutes, as it is uploading everything.)



4. From within this curriculum-jbook directory, run the ghp import command  
- ghp-import -n -p -f ~/Documents/curriculum-jbook/\_build/html  
(This should also take a few minutes.)
5. go to <https://callysto.github.io/curriculum-jbook/intro.html> to see the new book

## The \_config.yml file

There is a lot of magic in the config file. Below is the final one that worked for me. Notice:

- We can set the title and author here
- We can set the logo as a png to include. Be sure the png file is in the book directory
- We setup this for latex
- We can decide whether or not to execute notebooks before building the book
- Launch buttons, the repo storage, stuff about JS and parsing Latex is at the end

# File \_config.yml, book settings

title: Callysto Curriculum Notebooks

author: Callysto.ca (A joint project of Cybera Inc and the Pacific Institute for the Mathematical Sciences)

logo: CallystoLogo.png

latex:

  latex\_documents:

    targetname: book.tex

execute:

  execute\_notebooks: 'off'

  exclude\_patterns:

    - 'introduction-to-python.ipynb'

    - 'electrical-conductivity.ipynb'

repository:

  url : <https://github.com/callysto/curriculum-notebooks>

launch\_buttons:

  notebook\_interface : classic

  binderhub\_url : <https://mybinder.org>

  jupyterhub\_url : "<https://hub.callysto.ca>"

  thebelab : false

sphinx:

  config:

    html\_js\_files:

      - <https://cdnjs.cloudflare.com/ajax/libs/require.js/2.3.4/require.min.js>

parse:  
myst\_extended\_syntax: true

## Post-publishing HTML hacking

This sections will list bugs found and solutions that worked in those instances.

Bug: sound autoplayed from audio elements when page loaded.

Solution: ensure all `<audio>` tags do not contain the word `autoplay`. `<audio autoplay=false>` still auto-plays

Bug: custom html elements became heavily left adjusted

Solution: the generated code opened many `<div>` elements but did not close them. Adding sufficient `</div>` closing tags below the offending elements solved the issue

Bug: Image not appearing

Solution: Usually the generated code did not change the filepath from the source ipynb and did not place the image in the images folder. Copying the needed media from the curriculum notebooks repo into the images folder then adding the file path solved the issue.

bug : LaTeX rendering

Solution: if there are many LaTeX elements or they aren't formatted in the ipynb as specified above, or there is code that generates LaTeX it fails to render properly. Numerous times it starts rendering text as LaTeX and vice versa. Adjust the html so that:

`class="math notranslate nohighlight"`

Is inside the span or div element you wish to render as LaTeX. Additionally, if you want your LaTeX rendered inline surround it by `\(....\)` and `\[...]` if you want it rendered on a new line.

Example: `<span class="math notranslate nohighlight">\(f(x)\)</span>` and `<span class="math notranslate nohighlight">\(g(x)\)</span>` defined as follows