ME30_Lab1_18JUL18

August 29, 2018

1 ME 30 Lab 1 - Introduction to Anaconda, JupyterLab, and Python

ME 30 ReDev Team
2018-07-18

Description and Summary: This lab introduces Anaconda, JupyterLab, and Python.

Anaconda is a free, open source, software platform oriented toward data science and scientific computing. It provides an easy way to get started with Python and comes with a bunch of very powerful software packages (libraries) for scientific computing.

You will need to have installed Anaconda in order to do the exercises in this lab assignment. The computers in E407 and E390 already have Anaconda installed. If you want to work on your own machine, download the Python 3.x version for your operating system from: https://www.anaconda.com/download/

One of the software tools that comes with Anaconda is called JupyterLab. It is the next-generation user interface for Project Jupyter, an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. We will use the JupyterLab environment to write, run, and document Python code for homework and lab work in ME 30. For more in-depth information, see the References at the end of this document. ***

1.1 Getting Familiar with JupyterLab Notebooks

1.1.1 Step 1 - Launching the Anaconda Navigator

If you are working on a Windows machine, left-click in the Windows start search box and start typing Anaconda. It should locate the Anaconda Navigator icon, which you can left-click on to launch Anaconda. After a few moments of initialization, you should see:

1.1.2 Step 2 - Launching a JupyterLab Notebook

Click on the 'Launch' button for JupyterLab as indicated by the red arrow above. This will launch JupyterLab in a window in your default web browser. Once you launch JupyterLab, you should see:

```
<img src="JupyterLab_launch_window_new_folder.PNG"
alt="Anaconda Navigator Start Menu icon"
width="800px"/><br>
```

The list of files on the left will be different for the lab computers or if you are using the machine you installed Anaconda on. Left-click on the icon in the upper-left (shown with the blue arrow above) that looks like a folder with a '+' sign in its center, to create a new folder. The new folder will show up as, 'Untitled Folder' in the list of files. Right-click on this folder and select, 'Rename'. Rename it to, **ME30_Lab1_Jupyter_Intro**. Double-left click on this folder, so that your subsequent work from this lab session will be contained within it.

1.1.3 Step 3 - Working with Python code cells and Markdown cells

Click on the first icon for Python 3 under the Notebook icon:

```
<img src="JupyterLab_launch_window_w_arrow.PNG"
alt="Anaconda Navigator Start Menu icon"
width="180px"/><br>
```

This will start a JupyterLab notebook, which will look like:

```
<img src="UntitledJupyterLabNotebook.PNG"
alt="Anaconda Navigator Start Menu icon"
width="400px"/><br>
```

You should see a gray cell region, which is labeled to the left with **In [ ]:**. This is a Python *code* cell. Python code you type in this cell will be executed when you press SHIFT-Enter. To get started typing Python code, first press Enter (or left-click in the gray cell), and the cursor will be placed in the input cell. Try something like 2 + 2, and then press SHIFT-Enter.

Notice several things. First the Python code you just entered was *executed*, and the result was placed in an Output cell, labeled, Out [1]:, and another Python code cell appeared below the output cell:

The blue vertical line to the left of the new input cell indicates which cell has *focus*. To see this, click on the *first* input cell, and note that the focus of the cell changes. Click back in the empty input cell, and then select '**Markdown**' from the drop-down menu on the ribbon where it says 'Code' (shown by the red arrow below). [Markdown is a very simple syntax for writing for display in web browsers]:

Notice how the cell changes. The In [ ]: label disappeared, and now the cell is grayed out. With focus on this Markdown cell (blue bar to the left), press the Enter key (or left-click in the cell), and the cursor should now be in the cell. Go ahead and type some text in the cell, like:

ME 30 rocks and Jupyter is cool!

You now have a way to include text in your notebook. Basically, by placing Markdown cells above and below your Python code cells, you can include observations and make notes about the code you wrote, and do all of this without leaving the notebook. There is more to learn about Markdown, but we'll leave that to your exploration of web resources which cover it more extensively, like: + https://daringfireball.net/projects/markdown/syntax +

https://jupyter.brynmawr.edu/services/public/dblank/Jupyter%20Notebook%20Users%20Manual.ipynb + https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

Another way to find out how to use Markdown effectively is to look at other JupyterLab notebooks, such as this one, which you can find in the online resources for ME 30 or at many websites, such as this one.

1.2 Step 4 - Save your notebook!

When you launch a new JupyterLab notebook, the name defaults to 'Untitled.ipynb' as you should see in to the left in the Files view or by the tab at the top of the window. You can and should change the file name to something descriptive. It is best practice to NOT include spaces in your file name. Right-click on the tab showing the file name, and rename it to something like, `ME30Lab1Report.ipynb`.

JupyterLab notebooks can also be exported in other formats. For example, to save as pdf, select 'File' from the main ribbon menu in JupyterLab, and choose, 'Export Notebook As...', then choose PDF. Ask your lab instructor which format(s) you need to save your notebook to and upload to Canvas.

---

2 Exercise #1 - Practice with cells

Add Markdown cells to the ***beginning*** of your newly saved file (before the cells you were just working with), and format them like the beginning of this guide, so you can begin building your report for Lab 1. When you finish Exercise #1, you should have the following lines at the start of your notebook:

- A Title (using Header level 1. See the Markdown syntax reference for more information about headers.)
- Author name (using **bold** emphasis. See the Markdown referencence for more information about emphasis.)
- The date
- A sub heading for Description and Summary (using Header level 4)
- A paragraph describing what this lab is about
  - Three paragraphs that describe:

1. What you did
2. How you did it
3. What you learned in the lab exercise

You might need to refer to the links at the end of Step 3 to help with your formatting using Markdown.

Helpful Hints: 1. You can copy cells within a notebook or between notebooks. Right-click on the blue focus line to the left of any cell and explore the menu options. 2. You can add cells below a cell (that currently has focus) using the icon on the horizontal ribbon at the top of the notebook that looks like a '+' sign, and remove cells using the icon that looks like scissors. There are also

many keyboard shortcuts that you can find to the left of the screen with the vertical tab labeled 'Commands'.

2.1 Step 5 - Run some Python code and explore some other notebook commands

So far we have been looking at some of the features of the JupyterLab notebook. To whet your appetite for some of the cool things you can do with Python, copy the following block of code into a Python code cell and run it (either CNTRL-Enter or choose the right-pointing triangle from the ribbon menu at the top of the notebook). [Note: if you copy-paste the code below, make sure that all the lines have the same level of indentation in the code cell, preferably no indentation in this case. You will learn later that indentation in Python code cells is important as to how the code is interpreted]

```
%matplotlib inline
from matplotlib import pyplot as plt
import numpy as np
import os, sys
x = np.linspace(0, 5. * np.pi, 1000)
y = np.sin(x)
plt.plot(x, y)
plt.plot(x, y + 2)
plt.grid(True)
```

Neat, huh? Suppose you wanted to label the axes, so you can clearly show what is being plotted. Add these lines to the block of code you just entered:

```
plt.xlabel("Time, s")
plt.ylabel("Voltage, V")
```

Is the graph getting better? Might as well give it a title. Add the following line to the code block:

```
plt.title("Sine Waves vs. Time")
```

Now it looks pretty sharp! If you want to save it as an image file, add:

```
plt.savefig("sinewave plot.svg")
```

What you have just been doing (writing a little code, running it, adding more code, running it..., etc.) illustrates the ability of Python to run *interactively*. This interactivity allows you explore and build functionality of a program incrementally.

Look to see that the file **`sinewave plot.svg`** has shown up in the list of files to the left. Once it shows up, double-click on it. Doing so will open it in the JupyterLab image viewer. The JupyterLab notebook has the ability to view image files and manipulate them. Click on the vertical tab on the left labeled Commands. Scroll down until you get to the Image Viewer section. Try some of these commands when the sinewave plot window has focus. Document what you explored by describing it in a Markdown cell.

While we are on the topic of image files, the notebook has the capability to display images, your own (stored in the directory where the notebook is stored or externally, from the web). Create a new Markdown cell and copy-paste the following code in the cell, then run it:

```
<img src="http://tinyurl.com/y96k4q99" width="500" height="500" alt="surfer at Pipeline" title=
```

In a new Markdown cell below the one you just made, copy-paste the following code, which shows how you can include a graphic from the folder that the notebook is in:

```
<img src="sinewave plot.svg" width="800" height="800" alt="dual sinewave plot" title="dual sine
```

---

3 Exercise #2 - Practice including images in a notebook

1). Change the width and height parameters in the line of code you just placed from Step 5 above. Vary them between about 200 and 1000, and describe how the parameters affect the image size by adding your comments in a Markdown cell below the image.

2). Go back to the Python code cell that generated the plots of the two sine waves, and add this line at the end:

```
plt.savefig("sinewave plot.png")
```

[Note: **.png** is the 'Portable Network Graphics' file format. It is a 'bitmap' image, which means the image is made up of a *fixed* number of pixels. You first saved the graph to .svg, 'Scalable Vector Graphics' file format, which is a vector image file format, which means that it stores the image as a text-based XML file, which gives instructions for how to construct the image. How an image is stored has important implications when it comes to scaling, as you'll see next.]

3). Create a new Markdown cell that will display both the sinewave plot.svg and sinewave plot.png graphs (using the method for displaying an image that was described at the end of step 5). Use width and height parameters of 1000 for each image. How do the two images compare? Which one is clearer? So, if you have to scale images in the future, which file format would you want to use? Answer these questions in a Markdown cell below the images.

4). Save your notebook (Ctrl+s or left click the diskette icon on the horizontal ribbon at the top of your notebook).

You can export a notebook in different file formats. Use the dropdown menu under File on the main horizontal ribbon, and hover the cursor over the bottom-most entry, 'Export Notebook As...' We will us HTML format. Other formats are possible, such as PDF.

5). Export your notebook as a .html file. [File --> Export Notebook As..., and select HTML]. Locate the exported .html file using the Windows File Explorer and double-click on it to open it in a web-browser to see how it looks.

---

4 Exercise #3 - Additional Formatting with Markdown

1). Visit Adam Pritchard's Git Hub repository, the Markdown Cheatsheet, which is a quick reference and showcase for using Markdown. Find the section that describes Code and Syntax Highlighting.

2). Create a Markdown cell, copy-paste the code you worked with in Step 5 (that generated the two sine waves) into the Markdown cell, but use Python syntax highlighting to make the code more readable in the Markdown cell.

3). See what changes if you remove the keyword, *'python'* after the first set of three back ticks.

4). Create a new Markdown cell below the one you just generated. Include a Horizontal Rule to begin with, then add a blockquote with your favorite quote or saying.

5). Create a new Markdown cell below the one with your quote and include a link to your favorite YouTube video. Display the 0.jpg image for the video in your Markdown cell. Add some text before the image to indicate that clicking on the image will launch the video. Verify that you can access the video from within your notebook.

---

5 What to Turn In

Ask your lab instructor what he or she wants you to submit to Canvas for this lab session.

---

6 References

6.0.1 JupyterLab related

+ JupyterLab Docs http://JupyterLab.readthedocs.io/en/stable/index.html
+ Jupyter Notebook Users Manual (from Bryn Mawr College) https://jupyter.brynmawr.edu/services/public/dblank/Jupyter%20Notebook%20Users%20Manual.ipynb
+ Adam Pritchard's Markdown Cheatsheet https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet