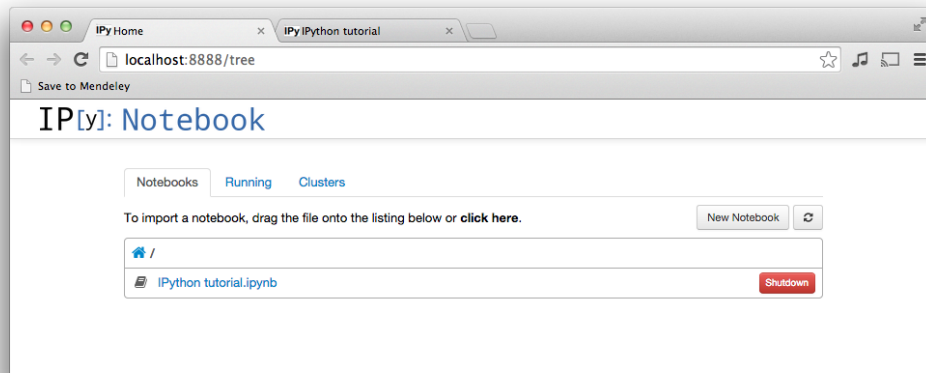


Lesson 2 – IPython & Numpy

Background: Now that you're familiar with Python, we will begin working with a widely used interactive Python environment (IPython Notebook, sometimes also called Jupyter Notebook) and Python library (NumPy) used for number manipulation. This lesson will require that you have installed Anaconda from Lesson 1.

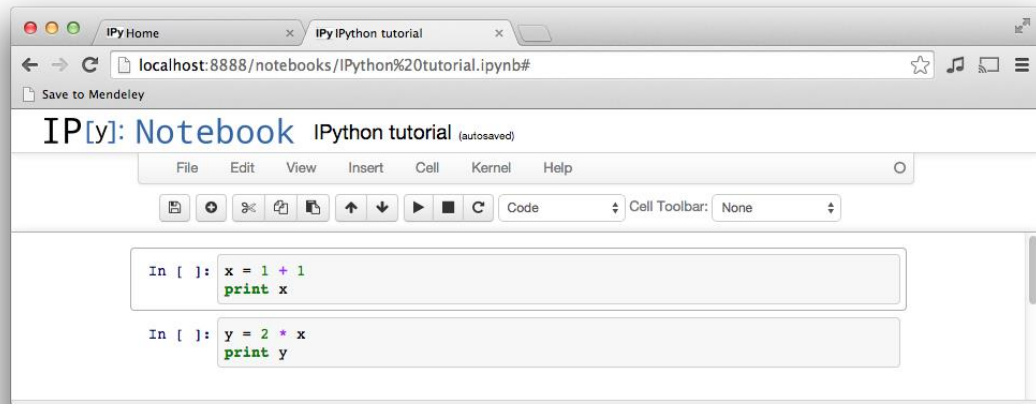
IPython Notebooks:

1. Open the Start menu and search 'Jupyter Notebook'
 - a. If it does not show up, navigate to the Anaconda folder on the start menu.
2. Alternatively, type `jupyter notebook` into the command prompt and press enter. Wait a few moments and a new web browser will popup with something that looks like the following:

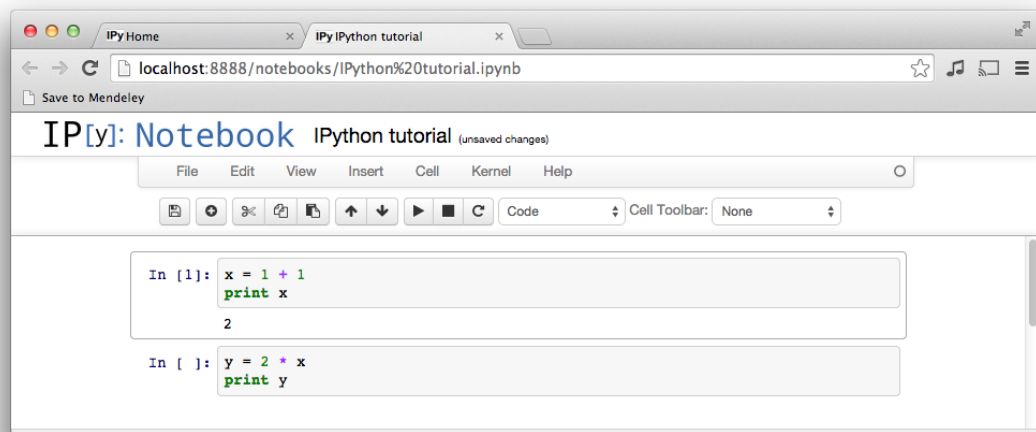


Note: This is your directory structure for all your IPython notebooks. You can create and modify your notebooks from here. IPython notebooks are great for making your Python programs along with all the documentation needed. It also saves output so you can reference back in the future.

3. Go ahead and click the “New Notebook” button to create your first notebook. Wait a moment and you will be greeted with a new screen which looks like the following:

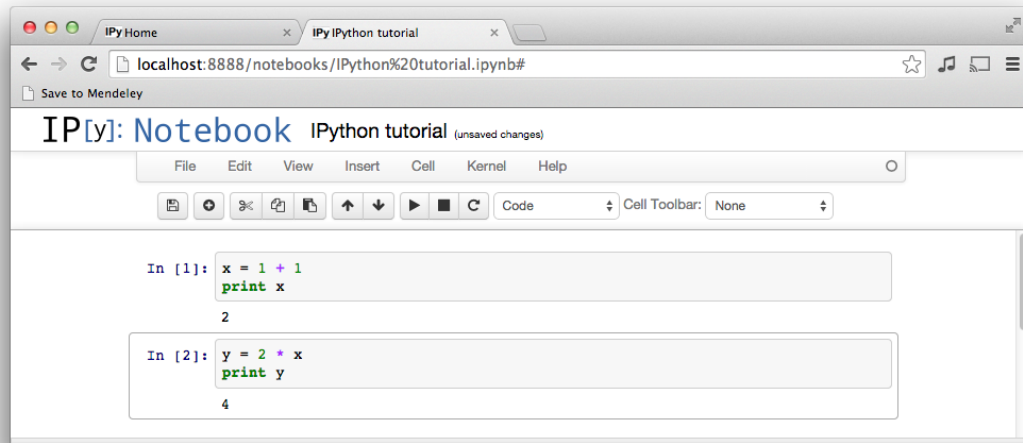


4. Give your IPython notebook a name by editing the title at the top. You can click ctrl + S to save your notebook at any time.
5. An IPython notebook is made up of a number of cells. Each cell can contain Python code. You can execute a cell by clicking on it and pressing 'Shift + Enter'. When you do so, the code in the cell will run, and the output of the cell will be displayed beneath the cell. For example, after running the first cell the notebook looks like this:



Note: Note the output below the first cell. You can save your notebook and pull it back up at any point in time in the future for reference.

6. Global variables are shared between cells. Executing the second cell thus gives the following result:

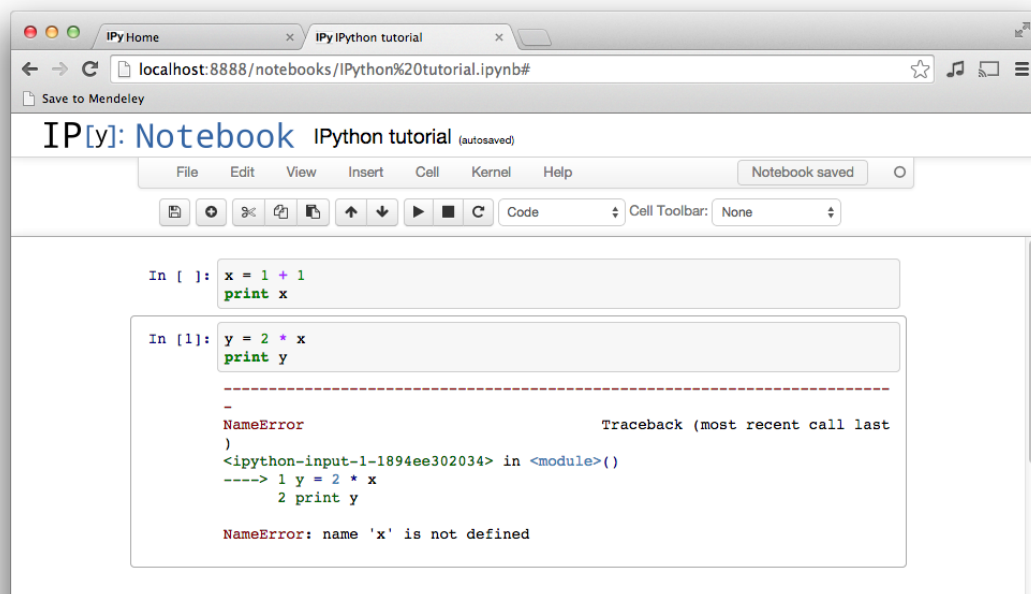


```
IPython Notebook IPython tutorial (unsaved changes)
File Edit View Insert Cell Kernel Help
[Icons] Code Cell Toolbar: None

In [1]: x = 1 + 1
        print x
        2

In [2]: y = 2 * x
        print y
        4
```

7. By convention, IPython notebooks are expected to be run from top to bottom. Failing to execute some cells or executing cells out of order can result in errors as seen below:



```
IPython Notebook IPython tutorial (autosaved)
File Edit View Insert Cell Kernel Help Notebook saved
[Icons] Code Cell Toolbar: None

In [ ]: x = 1 + 1
        print x

In [1]: y = 2 * x
        print y
        -----
        -
        NameError                                Traceback (most recent call last)
        <ipython-input-1-1894ee302034> in <module>()
        ----> 1 y = 2 * x
               2 print y
        NameError: name 'x' is not defined
```

Great! You can now begin coding all your programs in IPython notebooks to save your output.

NumPy:

1. Create a new IPython notebook and name it `numpy-practice.ipynb`.
2. Open the notebook and in the first cell, type the following: ``import numpy as np``.
3. Run the cell by pressing `shift + enter`. You have just successfully imported the entire NumPy.
4. library and can reference its functionality anywhere in your code with the keyword `np`.
5. Open the following tutorial: <http://www.kdnuggets.com/2017/03/working-numpy-matrices.html> and go through it.
6. Go through each section and type all the code in the black windows into their own respective cells. Run each cell once you've finished typing in the code. This will give you a good idea of how NumPy works and some of the things you can do with it.

Note: We are going to rely heavily upon NumPy for the rest of the lessons so it is very important you understand the ins and outs right now.

Note: You can stop after the "string arrays" section. We will not be requiring the functionality thereafter.