AI/HUB

Lesson 1

Implementation & Jupyter

OVERVIEW

For this lesson, we will be using the Anaconda distribution of Python. To gain an understanding of the basic principles and functions that the Python coding language provides, the first step is understanding how to implement Python on your computer.

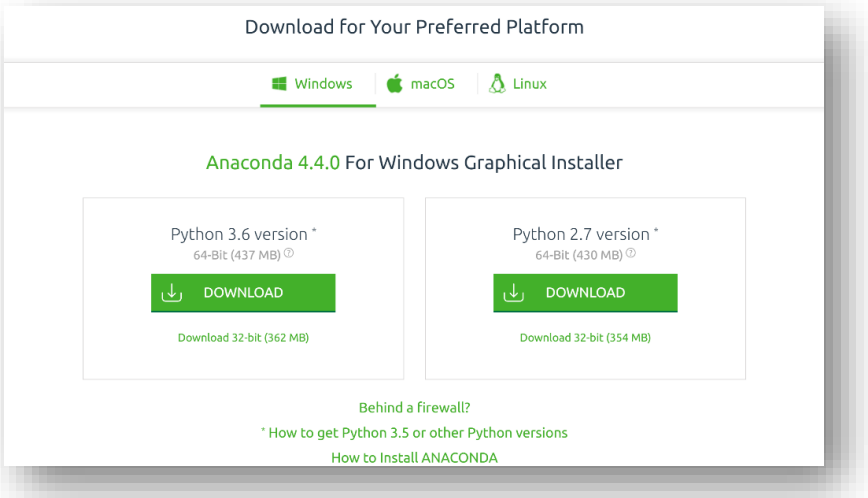
DOWNLOAD & INSTALL ANACONDA

Download Anaconda

1. Open a web browser and go to the following link:

   https://www.continuum.io/downloads

2. Scroll down and select your current Operating System.
3. Select the newest Python version available for download. (i.e. 3.x over 2.x)

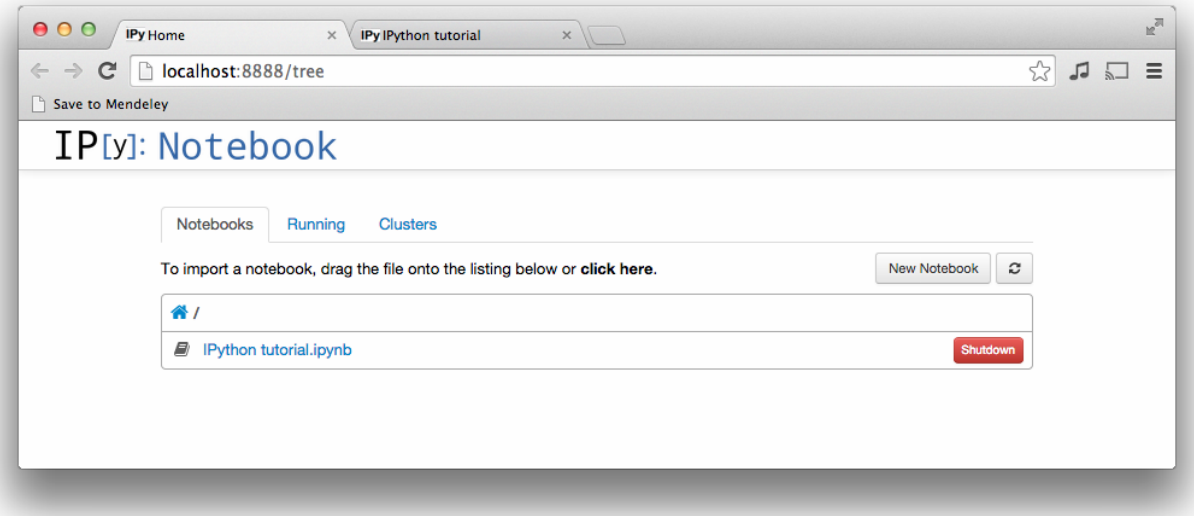


Install Anaconda

1. Run the install file once downloaded.
    a. Leave all installation settings as default!
    b. Ensure that Anaconda is selected as your default Python 3.x provider near the dend of the installer steps, just before commencing installation.

USING JUPYTER NOTEBOOKS

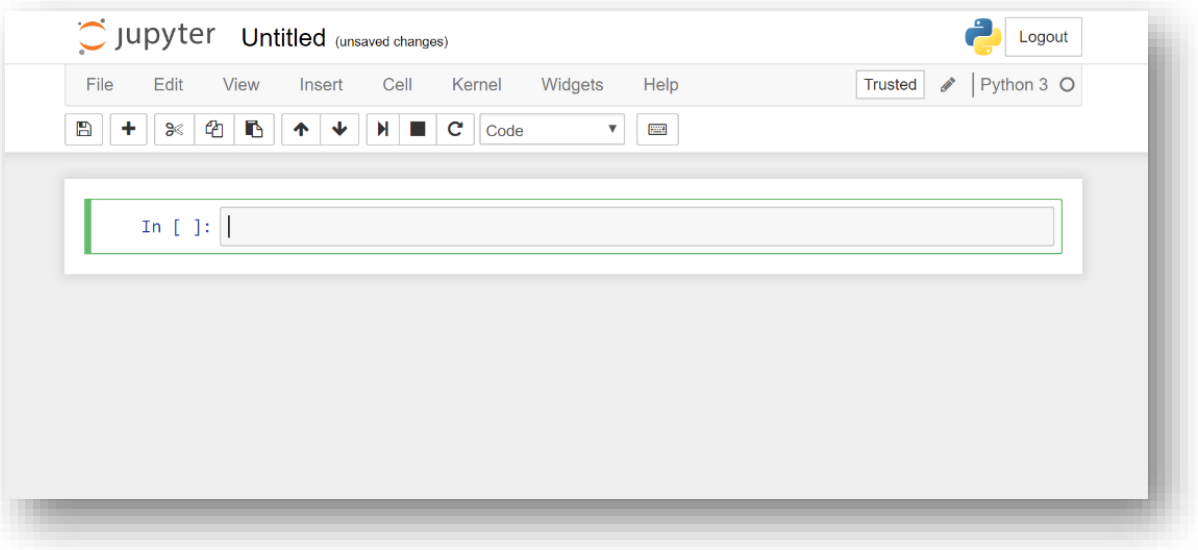
1. **Open the Start menu and search 'Jupyter Notebook'**
    a. *If it does not show up, navigate to the Anaconda folder on the start menu.*
2. **Right-click and add 'Jupyter Notebook' to your Start menu or Taskbar for easy access.**
3. **Alternatively, open the command prompt and type 'jupyter notebook', press enter.**
    a. *Wait a few moments and a new browser window should pop up that will look something like the following:*



NOTE

This is your directory structure for all your Jupyter notebooks. You can create and modify your notebooks from here. Jupyter 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.

4. **Click the 'New Notebook' button, to create your first Notebook.**
    a. *This should open a new browser tab that looks something like this:*



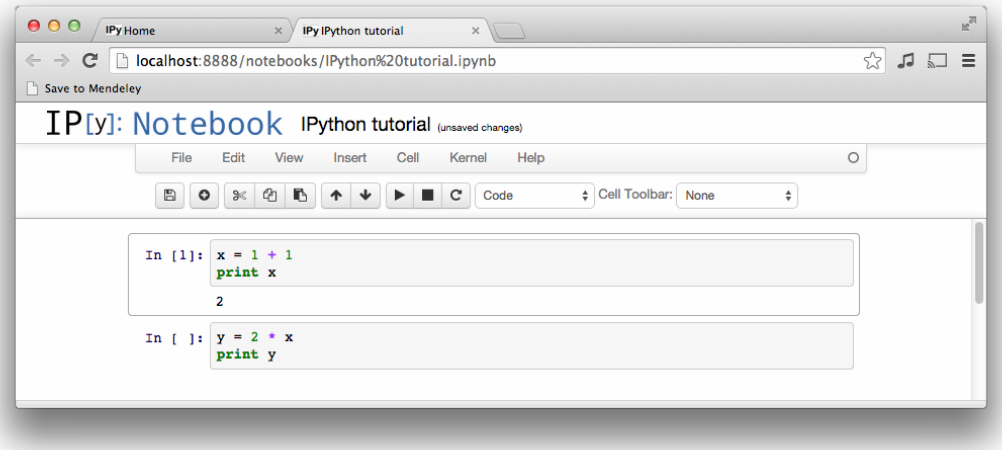
5. **Notice it says 'Untitled' at the top, click the text to rename your Notebook**

6. **You can use 'Ctrl + S' to save your progress at any time.**

NOTE

It is important to remember that any code you write for the **Python – Basics** boot camp, should be written in Jupyter for easier testing. Once your code is functional, copy it into a new text file and change the file extension to '**.py**'. *Double click that file to run your Python script!*

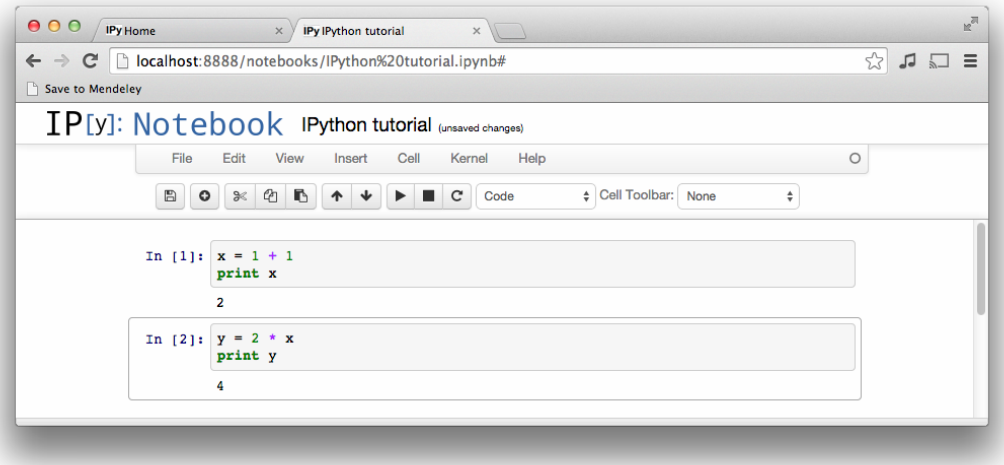
7. **A Jupyter notebook is made up of a number of cells. Each cell can contain Python code as well as Markdown (which is for styling).**
    a. *You can execute a cell by clicking on it and pressing **'Shift + Enter'**.*
    b. *When you do so, the code in the cell will run, and the output of the cell with be displayed beneath the cell.*
    c. *For example, after running the first cell in the example it looks like this:*



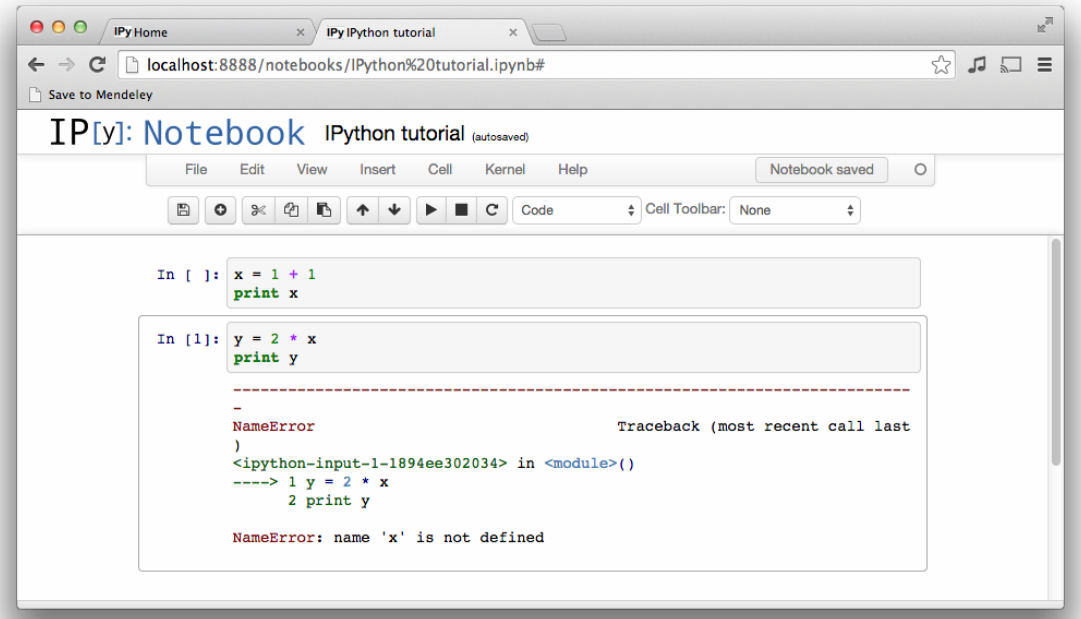
NOTE

Notice how the output was printed directly below the first cell. Saving your notebook will also save your outputs for future reference.

8. **Global Variables are shared between cells.**
    a. Executing the second cell in the example therefore provides the following result:



9. **By convention, Jupyter notebooks are expected to be run from <u>top to bottom</u>.**
    a. Failing to execute some cells, or executing out of order can result in errors like this one:



SUMMARY

1. Create a new Jupyter 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 library and can reference its functionality anywhere in your code with the keyword `**np**`.
4. Open and start the following tutorial:
    - https://www.kdnuggets.com/2017/03/working-numpy-matrices.html

NOTE

- We are going to rely heavily upon NumPy for the rest of the lessons, so it is especially important you understand the ins and outs right now.
- You can stop after completing the "**string arrays**" section in the 'kdnuggets' tutorial above. We will not be requiring the functionality demonstrated thereafter.