How to install Python?

There are two approaches to install Python?

- You can download Python directly from its project site "python.org/downloads/" and install individual components and libraries you want.

- Alternately, you can download and install a package, which comes with pre-installed libraries (Anaconda, anaconda.com/distribution). This method provides hassle free installation.

Choosing a development environment

Once you have installed Python, there are various options for choosing an environment. Following are the most common options.

- Terminal / Shell based
- IDLE
- iPython/Jupyter notebook

Running the Jupyter Notebook

One of the major components of the Jupyter project is the notebook, a type of interactive document for code, text, data visualizations, and other output.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R.

Running the Jupyter Notebook

To start up Jupyter, run the command jupyter notebook in a terminal

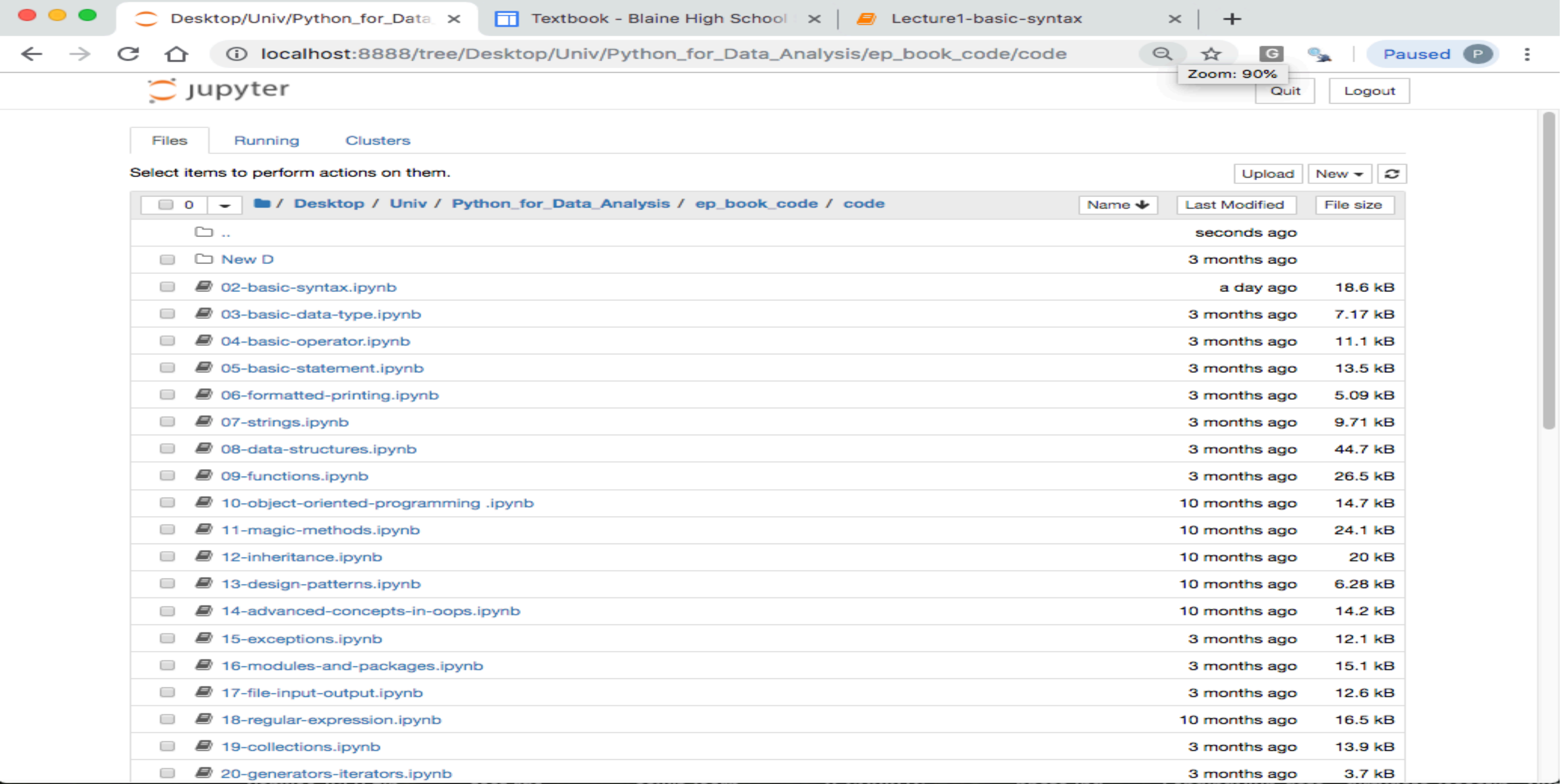
```
$ jupyter notebook
```

```
homes-MBP:~ home$ jupyter notebook
/Users/home/anaconda3/lib/python3.6/site-
packages/notebook/services/kernels/kernelmanager.py:19:
VisibleDeprecationWarning: zmq.eventloop.minitornado is deprecated in pyzmq
14.0 and will be removed.
    Install tornado itself to use zmq with the tornado IOLoop.

  from jupyter_client.session import Session
[11:30:32.666 NotebookApp]
```

Running the Jupyter Notebook

On many platforms, jupyter will automatically open up in your web browser. Otherwise, you can navigate to the http address printed when you started the notebook.



localhost:8888/tree/Desktop/Univ/Python_for_Data_Analysis/ep_book_code/code

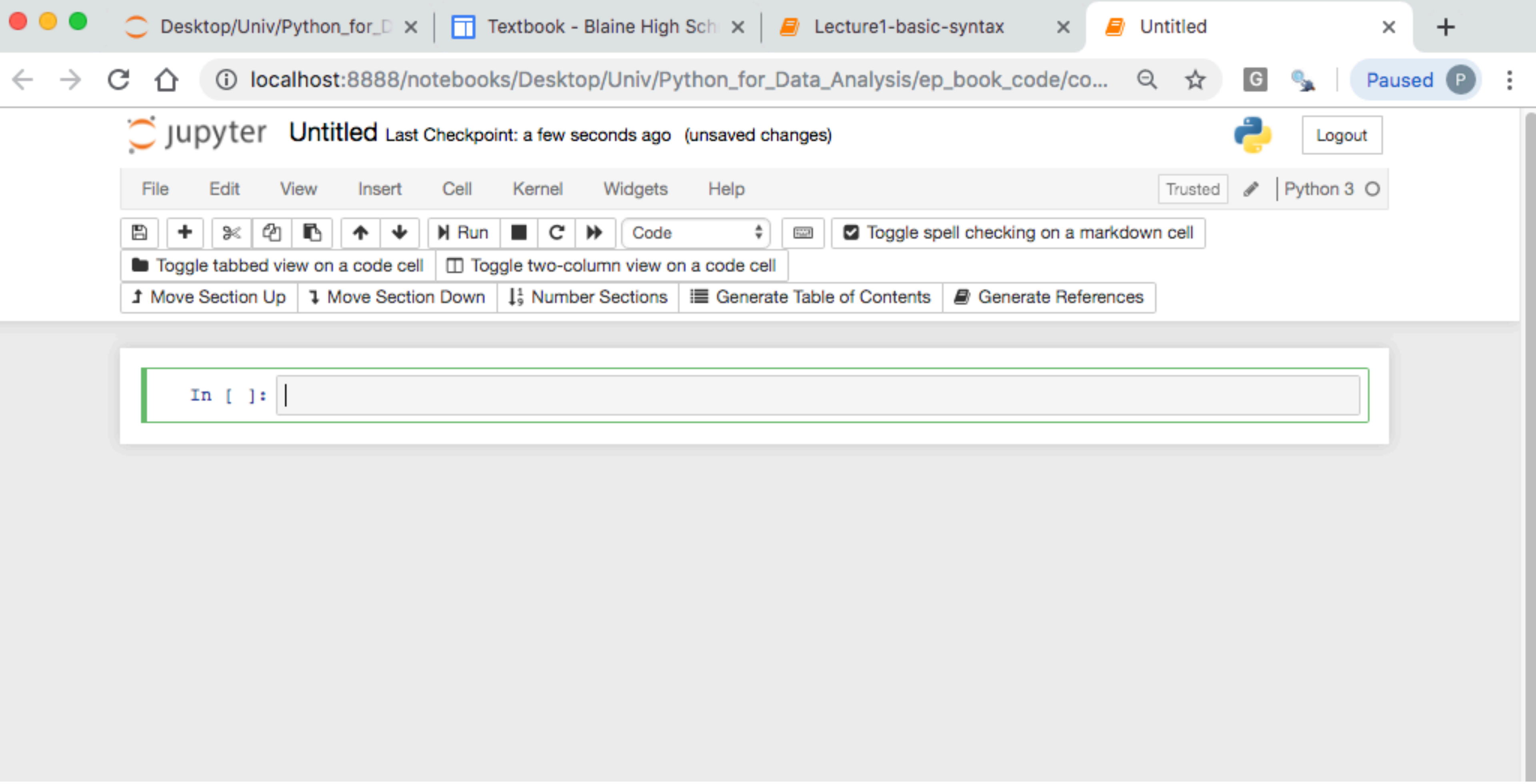
Select items to perform actions on them.

/ Desktop / Univ / Python_for_Data_Analysis / ep_book_code / code

Name	Last Modified	File size
..	seconds ago	
New D	3 months ago	
02-basic-syntax.ipynb	a day ago	18.6 kB
03-basic-data-type.ipynb	3 months ago	7.17 kB
04-basic-operator.ipynb	3 months ago	11.1 kB
05-basic-statement.ipynb	3 months ago	13.5 kB
06-formatted-printing.ipynb	3 months ago	5.09 kB
07-strings.ipynb	3 months ago	9.71 kB
08-data-structures.ipynb	3 months ago	44.7 kB
09-functions.ipynb	3 months ago	26.5 kB
10-object-oriented-programming .ipynb	10 months ago	14.7 kB
11-magic-methods.ipynb	10 months ago	24.1 kB
12-inheritance.ipynb	10 months ago	20 kB
13-design-patterns.ipynb	10 months ago	6.28 kB
14-advanced-concepts-in-oops.ipynb	10 months ago	14.2 kB
15-exceptions.ipynb	3 months ago	12.1 kB
16-modules-and-packages.ipynb	3 months ago	15.1 kB
17-file-input-output.ipynb	3 months ago	12.6 kB
18-regular-expression.ipynb	10 months ago	16.5 kB
19-collections.ipynb	3 months ago	13.9 kB
20-generators-iterators.ipynb	3 months ago	3.7 kB

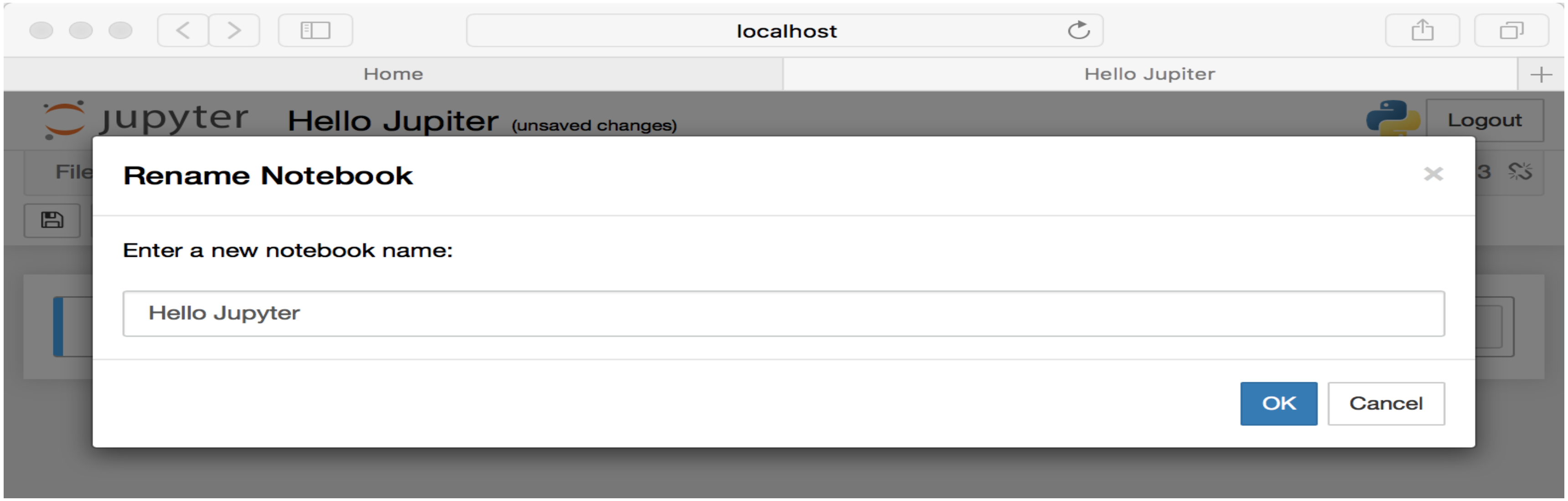
Running the Jupyter Notebook

To create a new notebook, click the New button and select the “Python 3” option. You should see something like


Desktop/Univ/Python_for_D
Textbook - Blaine High Sch
Lecture1-basic-syntax
Untitled
localhost:8888/notebooks/Desktop/Univ/Python_for_Data_Analysis/ep_book_code/co...
Paused
jupyter
Untitled Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Toggle spell checking on a markdown cell
Toggle tabbed view on a code cell
Toggle two-column view on a code cell
Move Section Up
Move Section Down
Number Sections
Generate Table of Contents
Generate References
In [ ]:

Naming

- You will notice that at the top of the page is the word *Untitled*. This is the title for the page and the name of your Notebook. Since that isn't a very descriptive name, let's change it!
- Just move your mouse over the word *Untitled* and click on the text. You should now see an in-browser dialog titled *Rename Notebook*. Let's rename this one to *Hello Jupyter*:



Rename Notebook

Enter a new notebook name:

Hello Jupyter

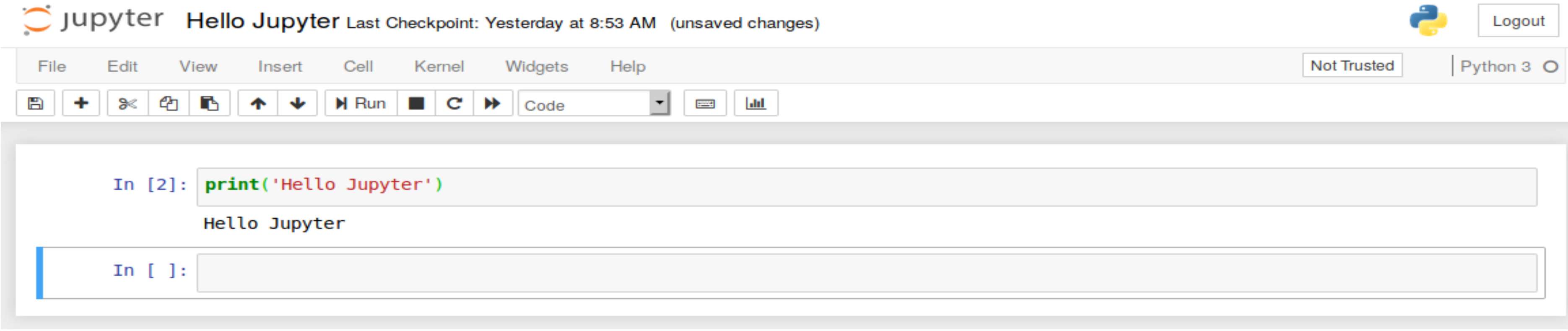
Running Cells

Running a cell means that you will execute the cell's contents. To execute a cell, you can just select the cell and click the *Run* button that is in the row of buttons along the top. It's towards the middle. If you prefer using your keyboard, you can just

Press Shift+Enter.

Example

print("Hello Jupyter")



```
In [2]: print('Hello Jupyter')

        Hello Jupyter

In [ ]:
```

If you have multiple cells in your Notebook, and you run the cells in order, you can share your variables and imports across cells. This makes it easy to separate out your code into logical chunks without needing to reimport libraries or recreate variables or functions in every cell.

When you run a cell, you will notice that there are some square braces next to the word *In* to the left of the cell. The square braces will auto fill with a number that indicates the order that you ran the cells. For example, if you open a fresh Notebook and run the first cell at the top of the Notebook, the square braces will fill with the number *1*.

The Menus

The Jupyter Notebook has several menus that you can use to interact with your Notebook. The menu runs along the top of the Notebook just like menus do in other applications. Here is a list of the current menus:

- *File*
- *Edit*
- *View*
- *Insert*
- *Cell*
- *Kernel*
- *Widgets*
- *Help*

If you have multiple cells in your Notebook, and you run the cells in order, you can share your variables and imports across cells. This makes it easy to separate out your code into logical chunks without needing to reimport libraries or recreate variables or functions in every cell.

When you run a cell, you will notice that there are some square braces next to the word *In* to the left of the cell. The square braces will auto fill with a number that indicates the order that you ran the cells. For example, if you open a fresh Notebook and run the first cell at the top of the Notebook, the square braces will fill with the number *1*.

Cell Types

There are technically four cell types: Code, Markdown, Raw NBConvert, and Heading.

The Heading cell type is no longer supported and will display a dialog that says as much. Instead, you are supposed to use Markdown for your Headings.

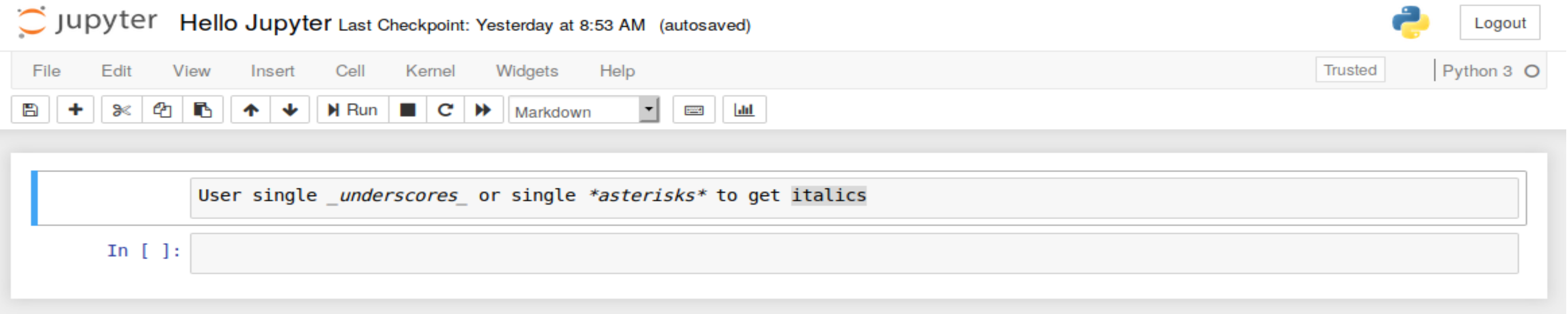
The Raw NBConvert cell type is only intended for special use cases when using the nbconvert command line tool. Basically it allows you to control the formatting in a very specific way when converting from a Notebook to another format.

The primary cell types that you will use are the Code and Markdown cell types. You have already learned how code cells work, so let’s learn how to style your text with Markdown.

Styling Your Text

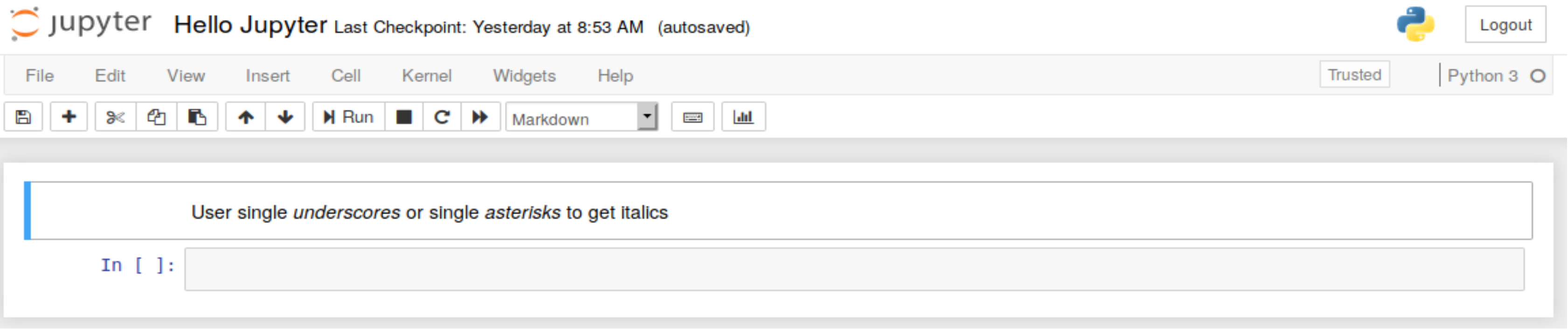
Jupyter Notebook supports Markdown, which is a markup language that is a superset of HTML. This tutorial will cover some of the basics of what you can do with Markdown.

Set a new cell to Markdown and then add the following text to the cell:



```
User single _underscores_ or single *asterisks* to get italics
```

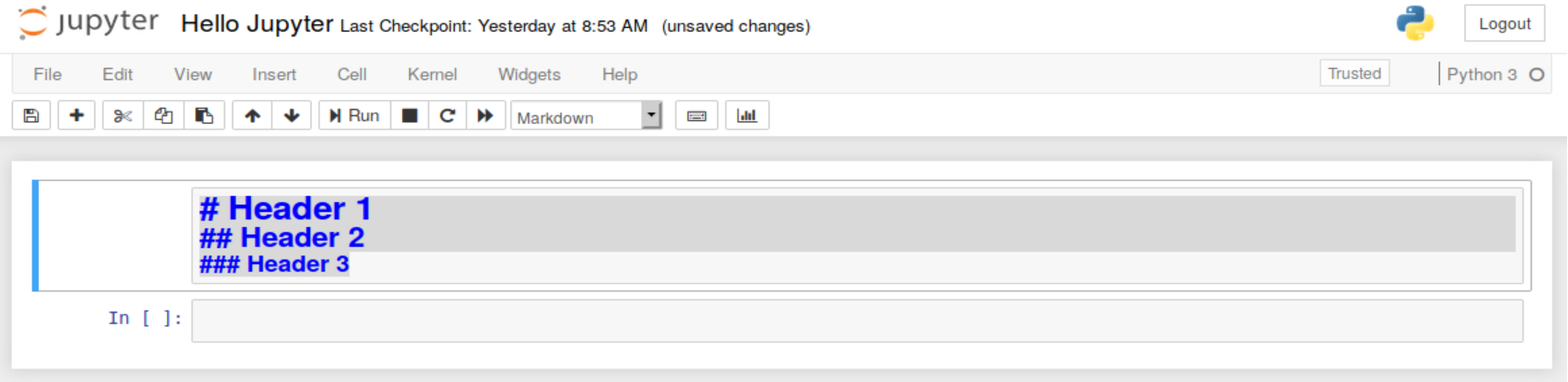
When you run the cell, the output should look like this:



User single *underscores* or single *asterisks* to get italics

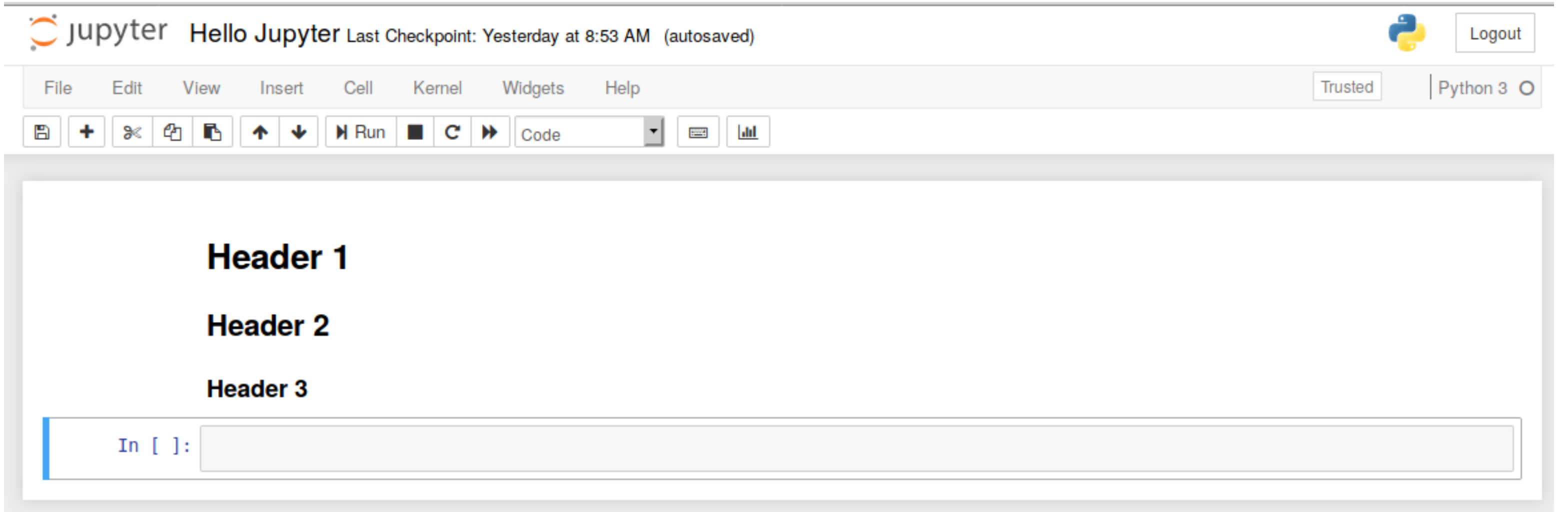
Headers

Creating headers in Markdown is also quite simple. You just have to use the humble pound sign. The more pound signs you use, the smaller the header. Jupyter Notebook even kind of previews it for you:



```
# Header 1
## Header 2
### Header 3
```

When you run the cell, you will end up with a nicely formatted header:



Header 1

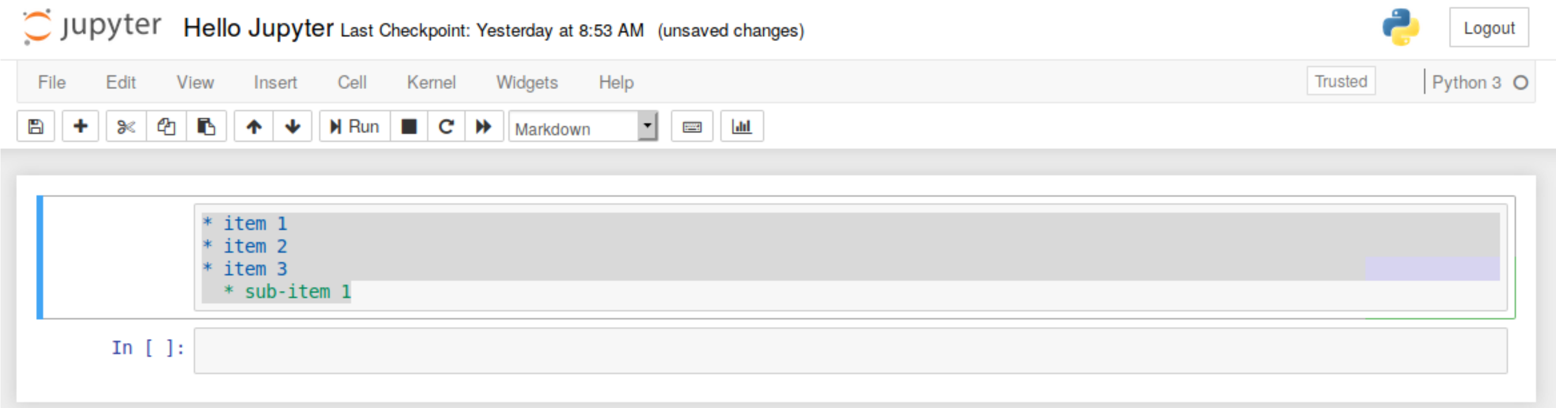
Header 2

Header 3

```
In [ ]:
```

Creating Lists

You can create a list (bullet points) by using dashes, plus signs, or asterisks. Here is an example:



```
* item 1
* item 2
* item 3
  * sub-item 1
```

```
In [ ]:
```

Exporting Notebooks

When you are working with Jupyter Notebooks, you will find that you need to share your results with non-technical people. When that happens, you can use the nbconvert tool which comes with Jupyter Notebook to convert or export your Notebook into one of the following formats:

- HTML
- LaTeX
- PDF
- RevealJS
- Markdown
- ReStructured Text
- Executable script

Sharing Your Notebooks

When people talk about sharing their notebooks, there are generally two paradigms they may be considering.

Most often, individuals share the end-result of their work, much like this article itself, which means sharing non-interactive, pre-rendered versions of their notebooks. However, it is also possible to collaborate on notebooks with the aid of version control systems such as Git or online platforms like Google Colab.

Before You Share

A shared notebook will appear exactly in the state it was in when you export or save it, including the output of any code cells. Therefore, to ensure that your notebook is share-ready, so to speak, there are a few steps you should take before sharing:

- Click “Cell > All Output > Clear”
- Click “Kernel > Restart & Run All”
- Wait for your code cells to finish executing and check ran as expected

This will ensure your notebooks don’t contain intermediary output, have a stale state, and execute in order at the time of sharing.

- https://www.dataquest.io/blog/jupyter-notebook-tutorial/

- https://realpython.com/jupyter-notebook-introduction/

- https://www.dataquest.io/blog/advanced-jupyter-notebooks-tutorial/

- https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks