Running Code

First and foremost, the Jupyter Notebook is an interactive environment for writing and running code. The notebook is capable of running code in a wide range of languages. However, each notebook is associated with a single kernel. This notebook is associated with the IPython kernel, therefor runs Python code.

Code cells allow you to enter and run code

Run a code cell using `Shift-Enter` or pressing the ▶| button in the toolbar above:

```
In [6]:  ▶|  a = 19
```

```
In [7]:  ▶|  print(a)
```

```
19
```

There are two other keyboard shortcuts for running code:

- `Alt-Enter` runs the current cell and inserts a new one below.
- `Ctrl-Enter` run the current cell and enters command mode.

Cells

This is a markdown cell

See the menu "Cell --> Cell Type" to change any cell to a markdown

Double click on this cell to see some of the markdown format

Title

Heading 1

Heading 2

- Bullet 1
- Bullet 2

Run this cell to see the formatting again.

Raw NBConvert

```
Raw NBConvert acts like a text only cell.  Also good for a notes,
interpretations, etc..
```

Cell menu

The "Cell" menu has a number of menu items for running code in different ways. These includes:

- Run and Select Below
- Run and Insert Below
- Run All
- Run All Above
- Run All Below

Restarting the kernels

The kernel maintains the state of a notebook's computations. You can reset this state by restarting the kernel. This is done by clicking on the ↻ in the toolbar above.

Make a markdown

```
In the cell below..
Make you name as Title "#"
Date as Heading 2 "##"
example:
# My Name
## December 1, 2020

Then run the cell
```

Allan Herrera

January 12, 2022

Output is asynchronous

All output is displayed asynchronously as it is generated in the Kernel. If you execute the next cell, you will see the output one piece at a time, not all at the end.

In [8]:

```
import time, sys
for i in range(9):
    print(i)
    time.sleep(0.5)
```

```
0
1
2
3
4
5
6
7
8
```

Large outputs

To better handle large outputs, the output area can be collapsed. Run the following cell and then single- or double- click on the active area to the left of the output:

In [9]:

```
for i in range(50):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

Beyond a certain point, output will scroll automatically:

```
In [10]:  for i in range(500):
              print(2**i - 1)
```

```
0
1
3
7
15
31
63
127
255
511
1023
2047
4095
8191
16383
32767
65535
131071
262143
524287
```

END