# 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 [1]:

```
a = 10
```

In [2]:

```
print(a)
```

```
10
```

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

# Richa Patel

## 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:
# Richa Patel
## January 15, 2021

Then run the cell
```

Type *Markdown* and LaTeX: $\alpha^2$

# 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 [3]:

```
import time, sys
for i in range(8):
    print(i)
    time.sleep(0.5)
```

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

# 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 [4]:

```
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
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

Beyond a certain point, output will scroll automatically:

```
In [5]: for i in range(500):
            print(2**i - 1)
```

```
309485009821345068724781055
618970019642690137449562111
1237940039285380274899124223
2475880078570760549798248447
4951760157141521099596496895
9903520314283042199192993791
19807040628566084398385987583
39614081257132168796771975167
79228162514264337593543950335
158456325028528675187087900671
316912650057057350374175801343
633825300114114700748351602687
1267650600228229401496703205375
2535301200456458802993406410751
5070602400912917605986812821503
10141204801825835211973625643007
20282409603651670423947251286015
40564819207303340847894502572031
81129638414606681695789005144063
162259276829213363391578010288127
```

# END