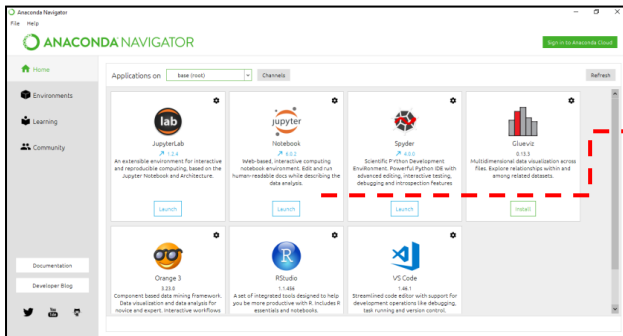


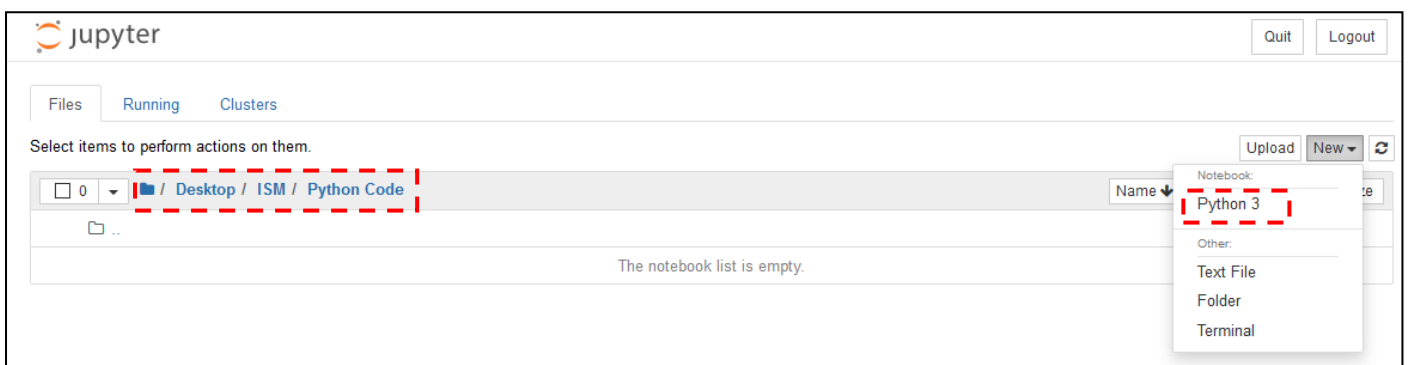
Objective: To provide a quick start with the basic Python in Jupyter Notebook in Anaconda. Anaconda is an environment and distributor for Python. This worksheet also introduces basic data types like strings, integers, floating point numbers and boolean. Students are expected to practice all the steps.

Prerequisite: Students must have installed [Anaconda](#) and able to launch web based Jupyter Notebook through it.

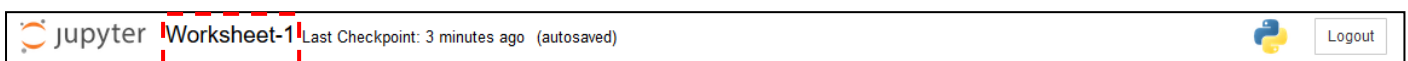
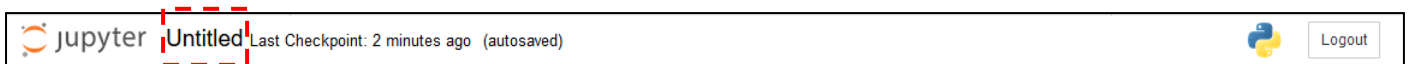


Steps: After the launch, the Jupyter Notebook opens up in your web browser.

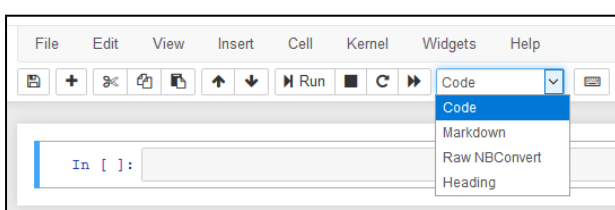
- You can select the directory where you want to save your Python files. For example `/Desktop/ISM/Python Code` is selected in this worksheet.
- Open a new notebook using the `New → Python 3` option as shown below.

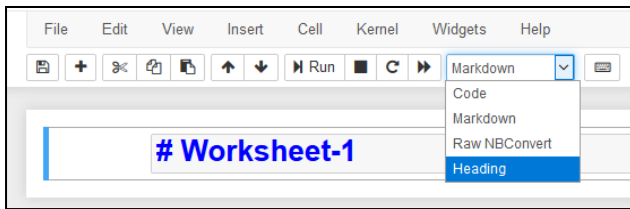


- On top of the notebook, initially notebook name is `Untitled`, re-name it as per your choice clicking it. In this worksheet, it is renamed as `Worksheet-1`.

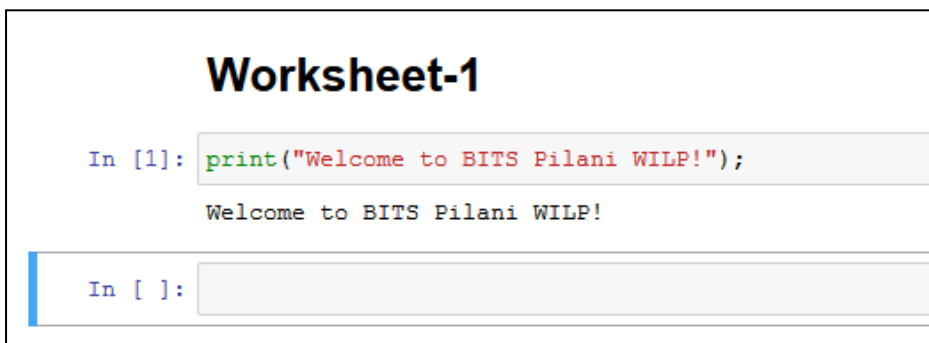


- Cells are shown in the work area of the notebook **In []:** A cell is a multiline text input field, where the user will type Headings, Python statements, expressions etc. A user can organize the notebook in headings and subheadings and organize the Python code under them as shown below:





- First level heading precedes with #, second level with ## and so on.
- When you start writing code, select **Code**.
- When the cursor is in the cell, press **SHIFT+ENTER** (or **CTRL+ENTER**) to accept the changes in the cell and move to the next cell.
- To delete a complete cell, select the cell, press **ESC** and press letter **D twice (DD)**. More keyboard shortcuts can be seen from **HELP→ Keyboard Shortcuts**.
- To write a simple print Python statement you can do the following and press **SHIFT+ENTER** to run and move to the next cell. Output will be displayed just below it. Python statements are not to be terminated with semicolon (;).



- A cell can have multiple statements. Press just **ENTER** to move to the next line in the same cell.



- As you progress with the new cells or re-run them, you would notice they are auto-numbered sequentially, **In [1], In [2] ...**
- Now we pass the welcome message to the print statement using + operator. In this case, the content that is passed to print is called an **expression**. Python first evaluates the expression and then pass it to print.
- Numerical values can also be provided in expressions. Like 1234+3456.
- Any user comments can be entered using with a preceding # as shown in the example below. Python interpreter ignores such comments from execution. Notice a comment **#Arithmetic Calculations** below.

Worksheet-1

```
In [1]: print("Welcome" + " to" + " BITS" + " Pilani" + " WILP!");
```

```
Welcome to BITS Pilani WILP!
```

```
In [2]: # Arithmetic Calculations
```

```
print (2234+4322)
```

```
6556
```

```
In [ ]:
```

- In the above example "Welcome" , "to" etc. are all strings, 2234, 4322 are integers and + is an operator in the expression.
- We can also use floating point numbers which represent real numbers (numbers with decimals), as shown in the example below:

```
In [3]: #Floating point numbers
```

```
print(2344/4322)
```

```
0.5423415085608515
```

```
In [4]: #Floating point numbers
```

```
x = 22
```

```
y = 7
```

```
print(x/y)
```

```
3.142857142857143
```

- Python also supports boolean variables that can take only one of two values **True** or **False**. Logical operators like **and**, **or** can be applied to them.

```
In [5]: #Boolean Data Types
print(True and False)
```

```
False
```

```
In [6]: #Boolean Data Types
```

```
x = True
```

```
y = False
```

```
print(x or y)
```

```
True
```