



# Lecture # 2.2

## Anaconda and Jupyter Notebook

**DATA  
SCIENCE**



TensorFlow



pandas



REGRESSION



CLASSIFICATION

matplotlib



RECOMMENDATION SYSTEM



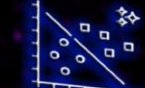
DIMENSIONALITY REDUCTION



FEATURE EXTRACTION



OPTIMIZATION



CLUSTERING



# Today's Agenda

---

- Intro to IPython
- Intro to Anaconda Navigator and its installation
- Creating Environment and managing packages in Anaconda
- Overview of Jupyter Notebook App
- Architecture of a Jupyter Notebook
- **Demonstration**
- Converting Jupyter Notebooks formats
- Sharing Jupyter Notebooks





# Introduction to

# IP[y]:

IPython



# IPython

---

- We have already written Python programs (files with the .py extension) and executed them in non-interactive mode of Python.
- We have executed various Python code snippets in Interactive mode using the Python terminal.

IP[y]:

IPython

- **IPython** is an interactive command-line terminal for Python, developed by Fernando Perez in 2001.
- **IPython** is a powerful interface to the Python programming language, in which you generally write one command at a time and you get the results instantly in REPL (read-eval-print loop)



# IPython (cont...)

- There are multiple ways to install IPython. One is to use Python package manager **pip/pip3**, which comes pre-installed when you install Python.

```
$ pip3 --version
```

```
$ pip3 list
```

```
$ pip3 install ipython
```

```
(base) Arifs-MacBook-Pro:Section-1 arif$ ipython
Python 3.8.10 (v3.8.10:3d8993a744, May 3 2021, 08:55:58)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.26.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: print ("Learning is fun with Arif")
Learning is fun with Arif
```

```
In [2]: 5 + 6
Out[2]: 11
```

```
In [3]: !pwd
/Users/arif/Desktop/DS-522/Demo-Files/Section-1
```

```
In [4]: █
```



# IPython (cont...)

In 2011, IPython introduced a new tool named the Notebook. Notebook offers a modern and powerful web interface to Python. Compared to the original IPython terminal, the Notebook offers a more convenient text editor, the possibility to write rich text, and improved graphical capabilities.

## Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#) using windowing, to reveal the frequency content of a sound signal.

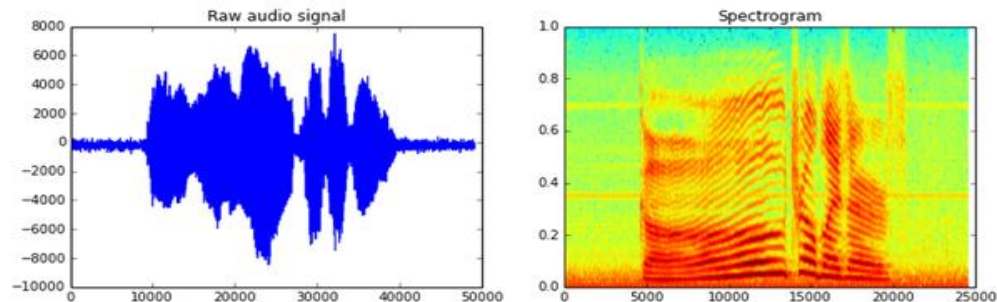
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi}{N} kn} \quad k = 0, \dots, N-1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin specgram routine:

```
In [2]: %matplotlib inline
from matplotlib import pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```





# Introduction to Jupyter Notebook



# Introduction to Jupyter Notebook

---

**Acronym for:**

Julia, Python and R

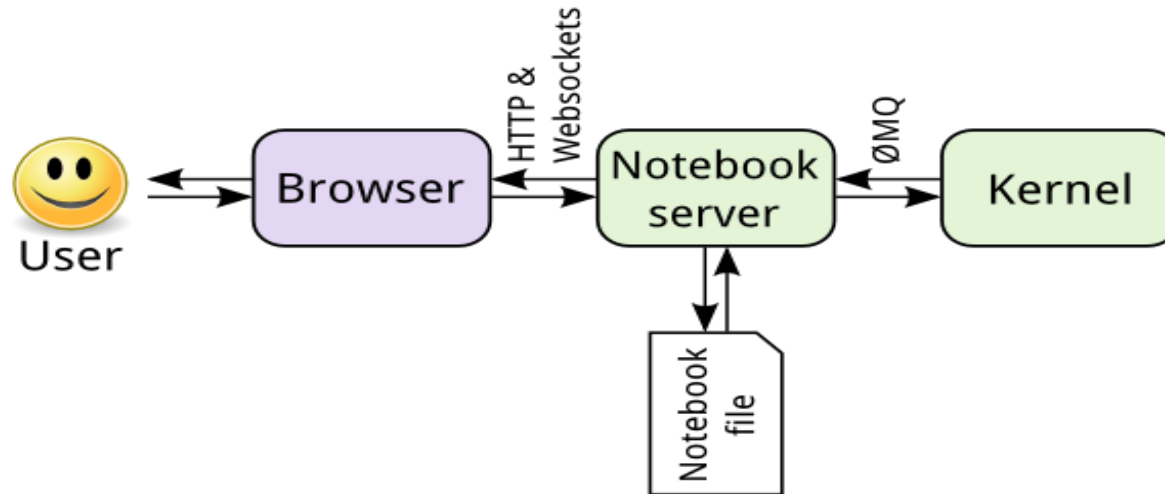


- The Jupyter Notebook App is a client-server application that allows editing and running notebook documents via a web browser.
- The Jupyter Notebook App can be executed on a local desktop requiring no Internet access or can be installed on a remote server and accessed through the Internet.
- Notebook is a single document where you can run code, display the output, and also add explanations, formulas, charts, and make your work more transparent, understandable, repeatable, and shareable.
- The easiest way for a beginner to get started with Jupyter Notebooks is by installing Anaconda.





# Notebook Architecture



- The notebook interface, which shows you a document with code and visualizations in it, called “**a Jupyter notebook**”
- The computational engine behind every notebook, that executes the code contained in the notebook is called the “**kernel**”
- When you open a Notebook document, the associated kernel is automatically launched.
- The “**notebook server**”, not the kernel, is responsible for saving and loading notebooks, so you can edit notebooks even if you don’t have the kernel for that language—you just won’t be able to run code.



# Jupyter in the Cloud



# Jupyter in the Cloud

---

- The easiest way to start using Jupyter is to just point your browser at a website that offers you access to Jupyter in the cloud.
  - The notebook interface runs in your browser on your computer
  - The kernel runs in the cloud on a server provided by someone else
- Some options are:
  - <https://deepnote.com>
  - <https://colab.research.google.com>
  - <https://cocalc.com>



Deepnote



COCALC



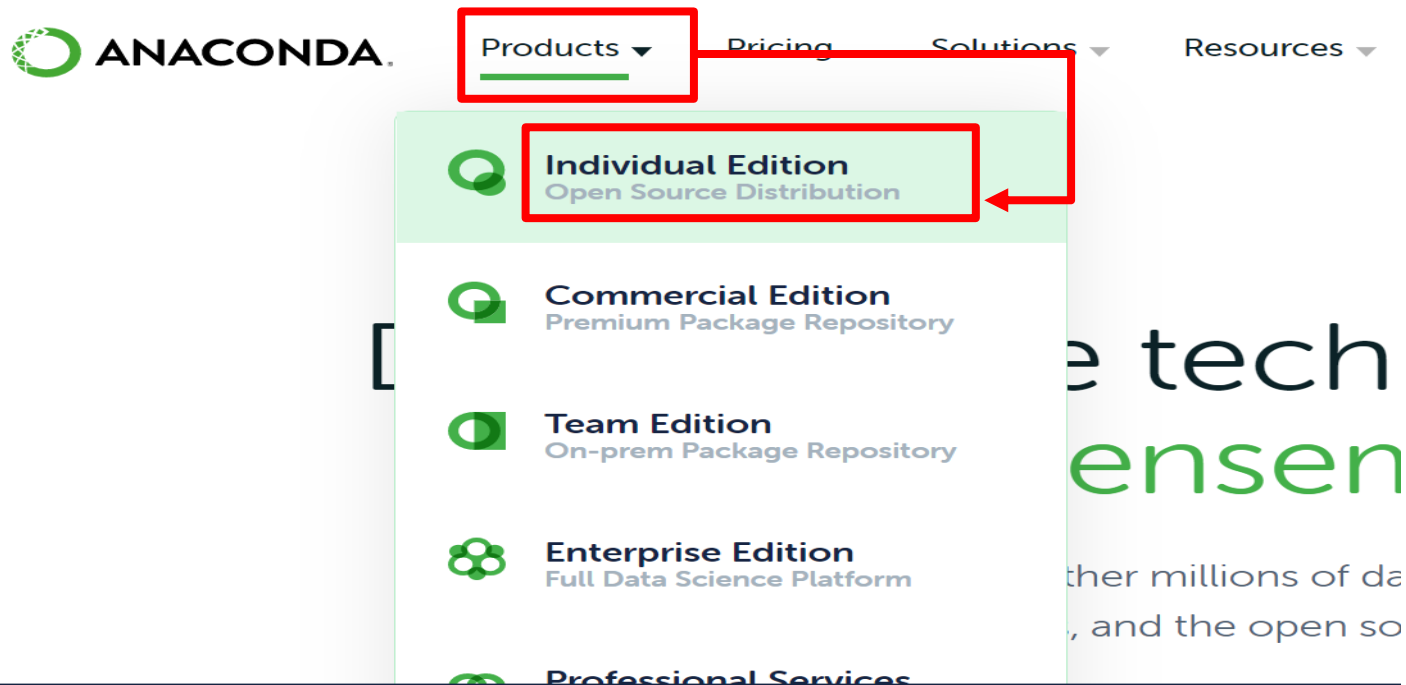


# Installing Anaconda



# Anaconda

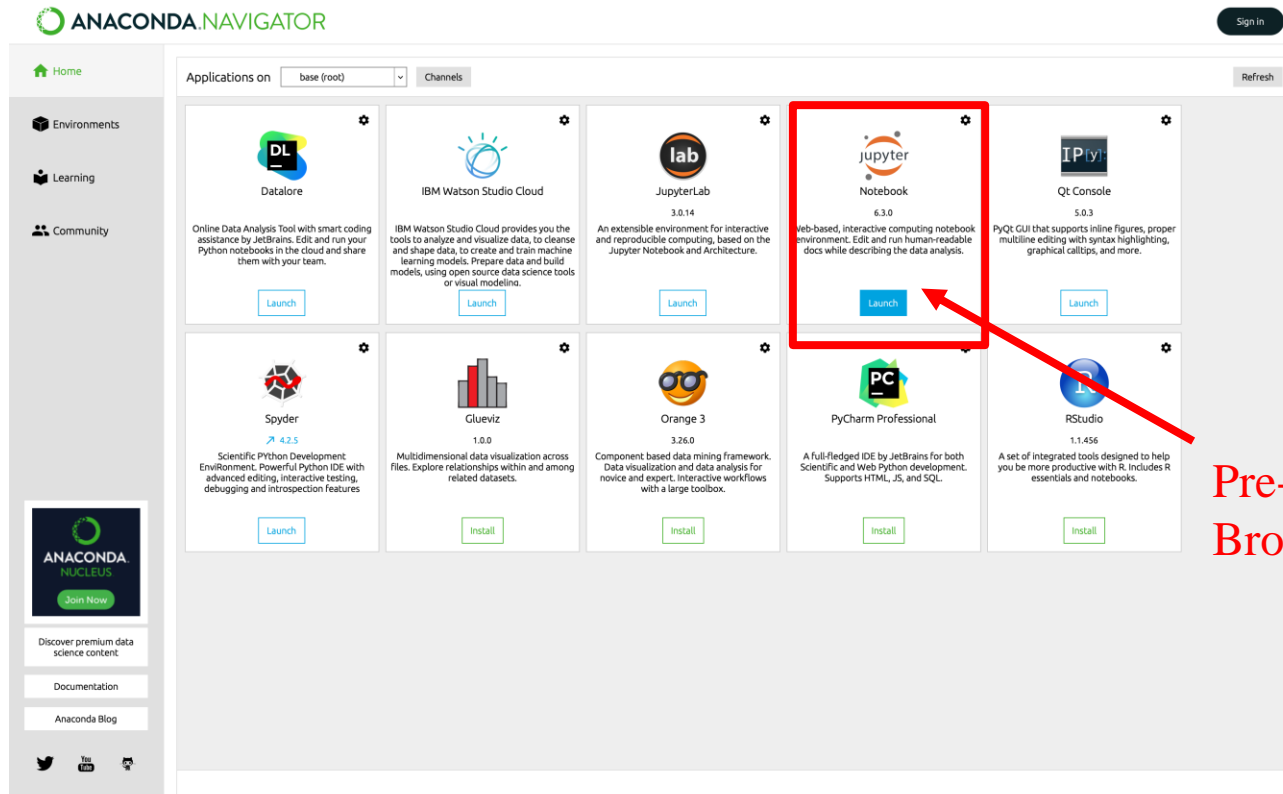
- Anaconda is the most widely used Python distribution for data science and comes pre-loaded with all the most popular libraries and tools.
- Download and install the latest version for your operating system ([www.anaconda.com](http://www.anaconda.com))





# Anaconda Navigator

- After installation, when you run Anaconda Navigator, you can see different pre-installed applications.
- Click the Jupyter Notebook App, it will start a server on the local machine at port 8888 and open a new tab in your default web browser with the Notebook dashboard (shown on next slide).

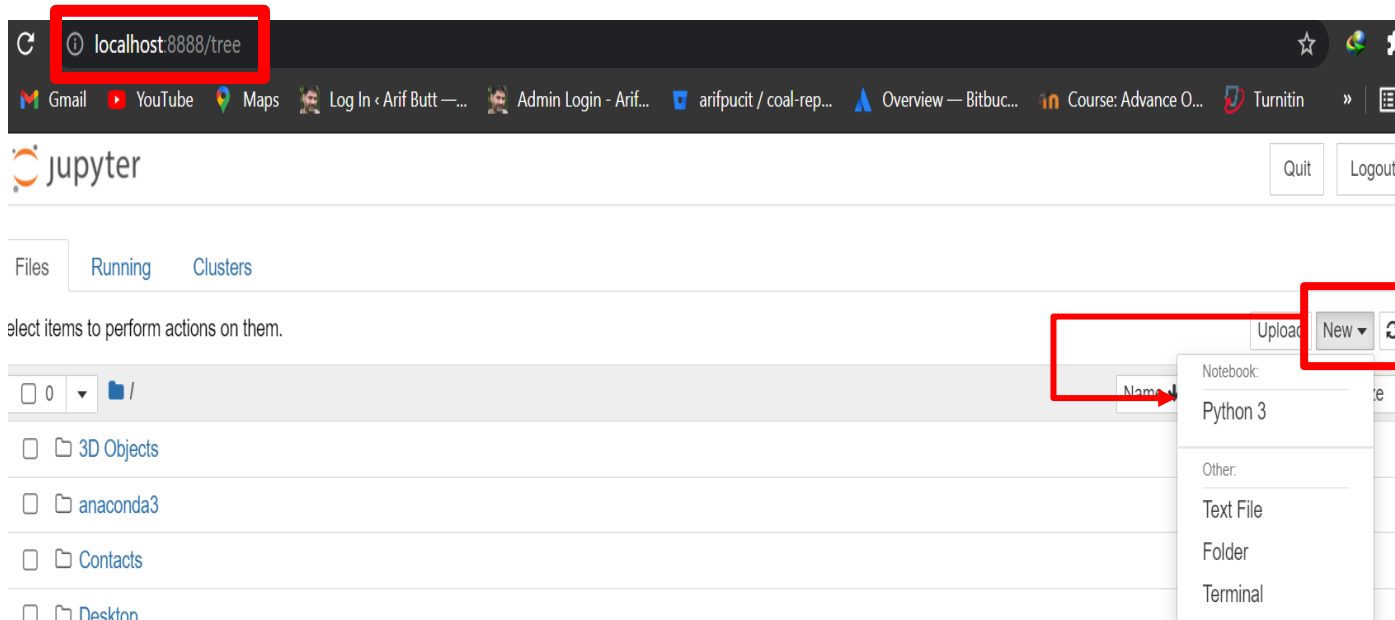


Pre-installed  
Browser-based IDE



# Jupyter Notebook Dashboard

- The Notebook Dashboard is the component which is shown first when you launch Jupyter Notebook App and is mainly used to open notebook documents and to manage the kernels.
- The dashboard serves as a home page for the notebooks. Its main purpose is to display the portion of the filesystem accessible by the user, and to provide an overview of the running kernels, terminals, and parallel clusters.
- To create a new notebook, click the New button and select the desired language kernel (in our case Python3).





# Jupyter Notebook Documents (notebooks)

- When a notebook is opened, a new browser tab will be created which presents the notebook user interface (UI). This UI allows for interactively editing and running the notebook document.
- **Header:** At the top of the notebook document is a header which contains the notebook title, a menu bar, and toolbar to control notebook navigation and document structure.
- **Body:** The body of a notebook is composed of cells. Each cell contains either markdown, code input, code output, or raw text.

The screenshot shows the Jupyter Notebook interface. At the top, the header includes the Jupyter logo, the notebook title 'first-notebook', and the status 'Last Checkpoint: 6 minutes ago (unsaved changes)'. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. To the right of the menu bar are 'Trusted' and 'Python 3 (ipykernel)' indicators. Below the menu bar is a toolbar with icons for saving, adding, undo, redo, home, back, forward, run, stop, refresh, and a dropdown menu currently set to 'Code'. The main area contains three cells:

- Markdown cell:** A cell containing the text 'This is a markdown cell'. A red arrow points from the label 'Markdown cell' to this cell.
- Code cell (input cell + output cell):** A cell containing the code `In [1]: #This is a code cell` and `print('Hello Jupyter Notebook')`. Below the code is the output 'Hello Jupyter Notebook'. A red arrow points from the label 'Code cell (input cell + output cell)' to this cell.
- Raw cell:** A cell containing the text 'this is a raw cell'. A red arrow points from the label 'Raw cell' to this cell.



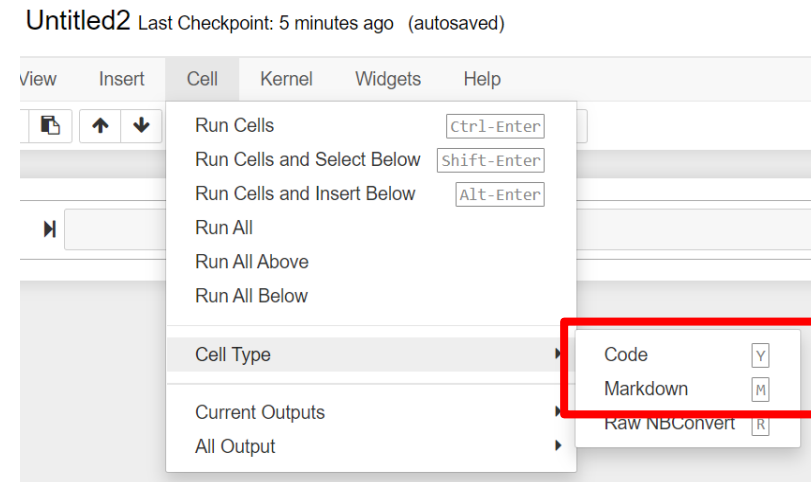
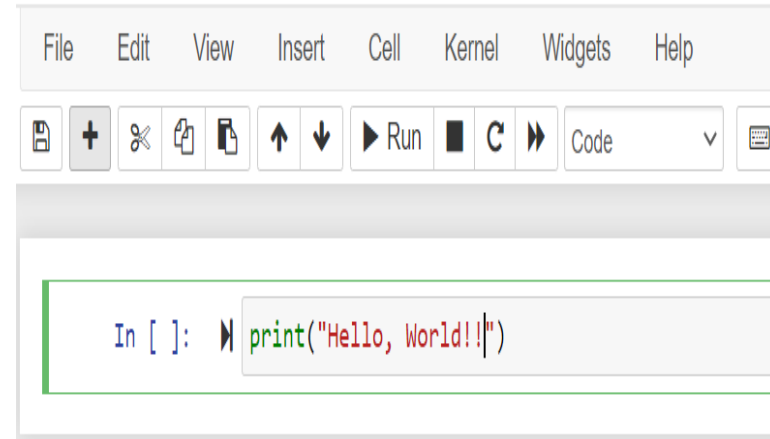


# Jupyter Notebook Documents (notebooks)

- Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc).
- Each time you create a new notebook, a new .ipynb file will be created (saved on disk in JASON format).

<https://nbformat.readthedocs.io/en/latest/>

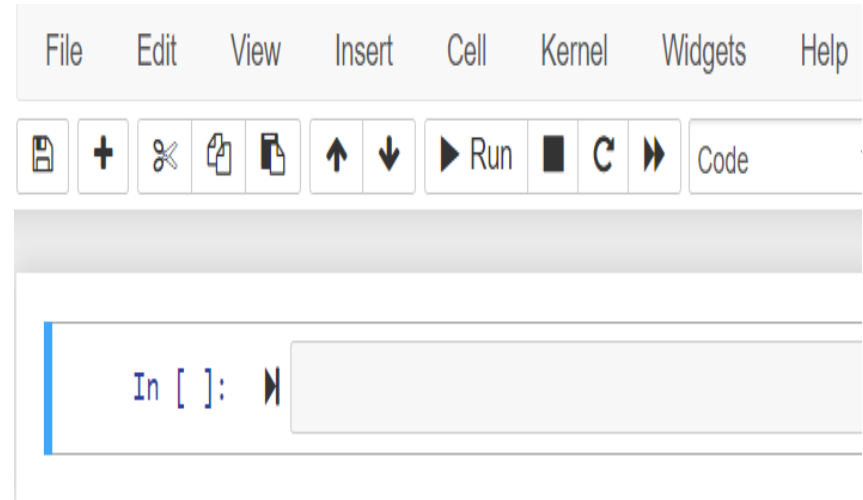
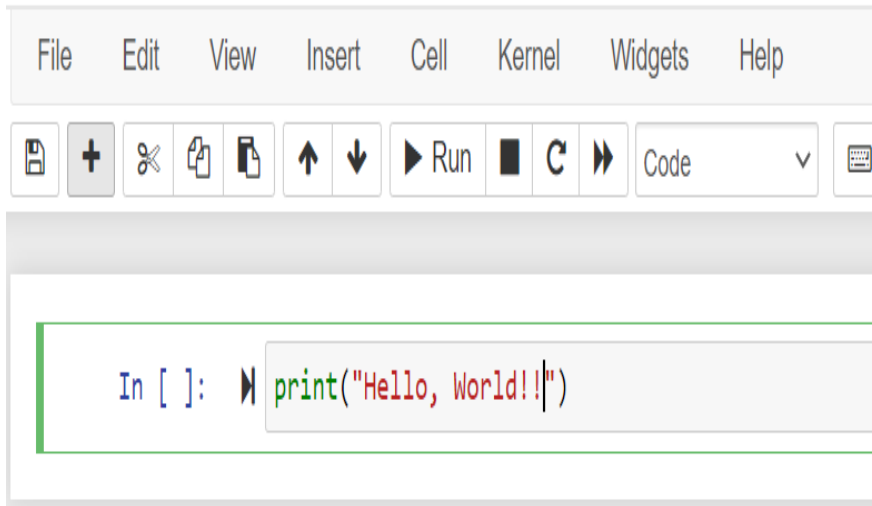
- A notebook cell is a container for text to be displayed in the notebook or code to be executed by the notebook's kernel.
- There are two main cell types:
  - Code cell
  - Markdown cell





# Edit and Command Mode of Cell

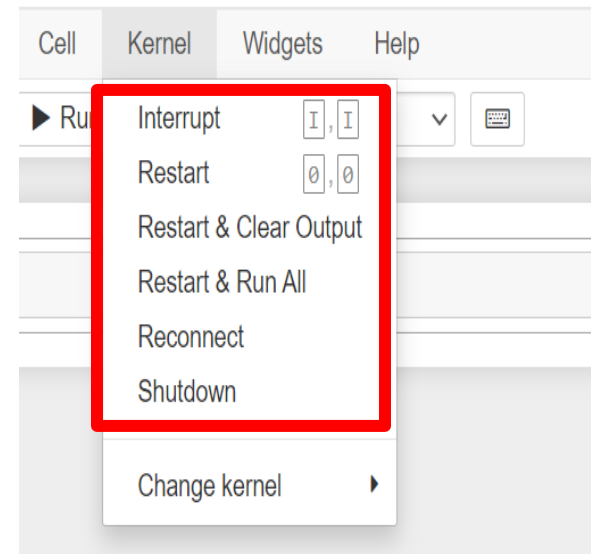
- When running your cells you may notice that their border turns blue, whereas it was green while you were editing. In a Jupyter Notebook, there is always one “active” cell highlighted with a border whose color denotes its current mode:
  - Green outline — cell is in "edit mode“
  - Blue outline — cell is in "command mode"





# Notebook Kernel

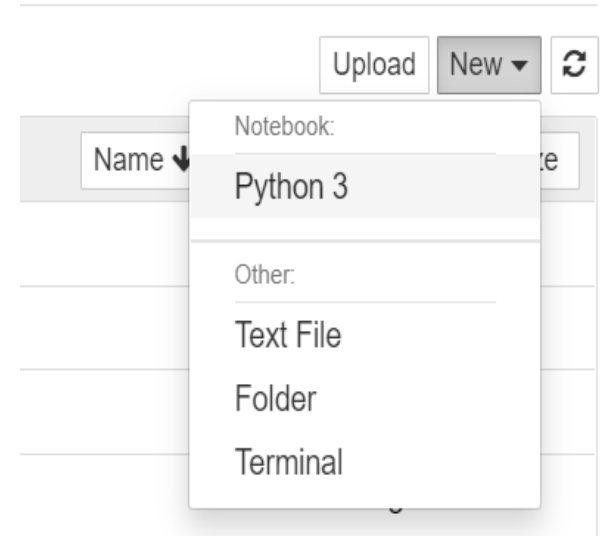
- For a specific notebook, following are the commands that you can give to the kernel
- **Restart:** restarts the kernel, thus clearing all the variables etc., that were defined.
- **Restart & Clear Output:** same as above but will also wipe the output displayed below your code cells.
- **Restart & Run All:** same as above but will also run all your cells in order from first to last.
- **Interrupt:** If your kernel is ever stuck on a computation and you wish to stop it, you can choose the Interrupt option.





# Notebook Kernel

- When you created a new notebook from the dashboard by selecting a Python version, you were actually choosing which kernel to use.
- There are kernels for different versions of Python, and also for over 100 languages including R, Julia, Java, C, and even Fortran.





# How to Shutdown Jupyter

---

- When you launch the Jupyter Notebook or Jupyter Lab, you launch both the user interface (which you see in your browser) and the kernel (which you don't see).
- If you want to close a specific notebook:
  - Close the tab of the browser in which notebook is being displayed and then in the running tab of Jupyter Notebook Dashboard, shutdown the associated kernel. OR
  - Click the File menu of the notebook you want to close and click Close and Halt
- Moreover, remember just closing the Jupyter Notebook Dashboard tab in your browser does not shutdown the service running at localhost on port 8888.



# Hello World in Jupyter Notebook

---

<https://github.com/arifpucit/data-science.git>



[\*View/Download from Jupyter nbviewer\*](#)

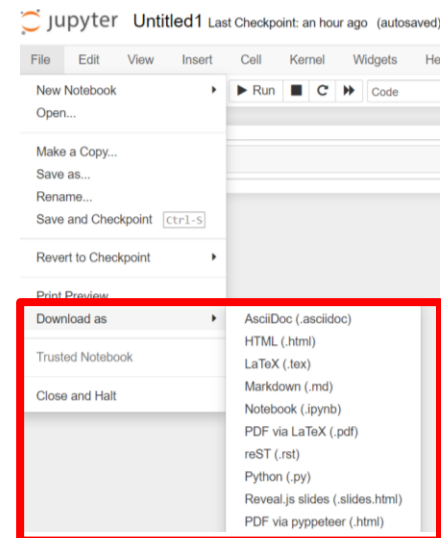


# Sharing Notebooks



# Sharing Notebooks

- Jupyter has built-in support for exporting ipynb files to HTML, PDF, LaTeX, and several other formats. If you wish to share your notebooks with a small private group, you can export your notebook file to an appropriate file format, and share your results with your peers and fellow researchers.
- Nbconvert is a tool that allows you to convert a Jupyter .ipynb notebook document file into another static format including HTML, LaTeX, PDF, Markdown, reStructuredText, and more.
- If used as a Python library, implement the “Download as” feature within the Jupyter Notebook web application



jupyter/**nbconvert**

Jupyter Notebook Conversion







# Sharing Notebooks (Cloud-based IDEs)

- Cloud-based IDEs allows you to share your notebook files with other people. The benefits of cloud IDEs include accessibility from anywhere in the world, on any device – from Chromebooks to Android and iOS devices.
- It normally do not require you to download, install and run any set up on your system and provide ease of collaboration among geographically dispersed developers.
- Google Colab is a cloud service based on jupyter notebook. If you are already signed-in to your Gmail account, you may directly jump to the Colab website and enjoy 😊

[www.colab.research.google.com](http://www.colab.research.google.com)



**trinket**



pythonanywhere

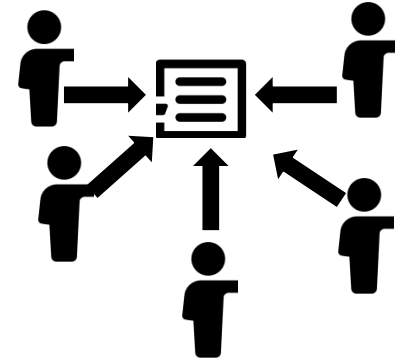


# Sharing Notebooks (GitHub:gist)

- In real life, while working on huge projects along with many team members, source control allows you to make sure that multiple people can easily work on same file from different locations.
- For this task **git** source control is one of the most popular options. Some popular Git source control options include GitHub, Bitbucket, and GitLab.
- The count of public notebooks on GitHub is continuously increasing, being the most popular independent platform for sharing Jupyter projects with the world.

<https://github.com/jupyter/jupyter/wiki>

- GitHub has integrated support for rendering .ipynb files directly both in repositories and gists on its website.



GitHub

ATLASSIAN

Bitbucket



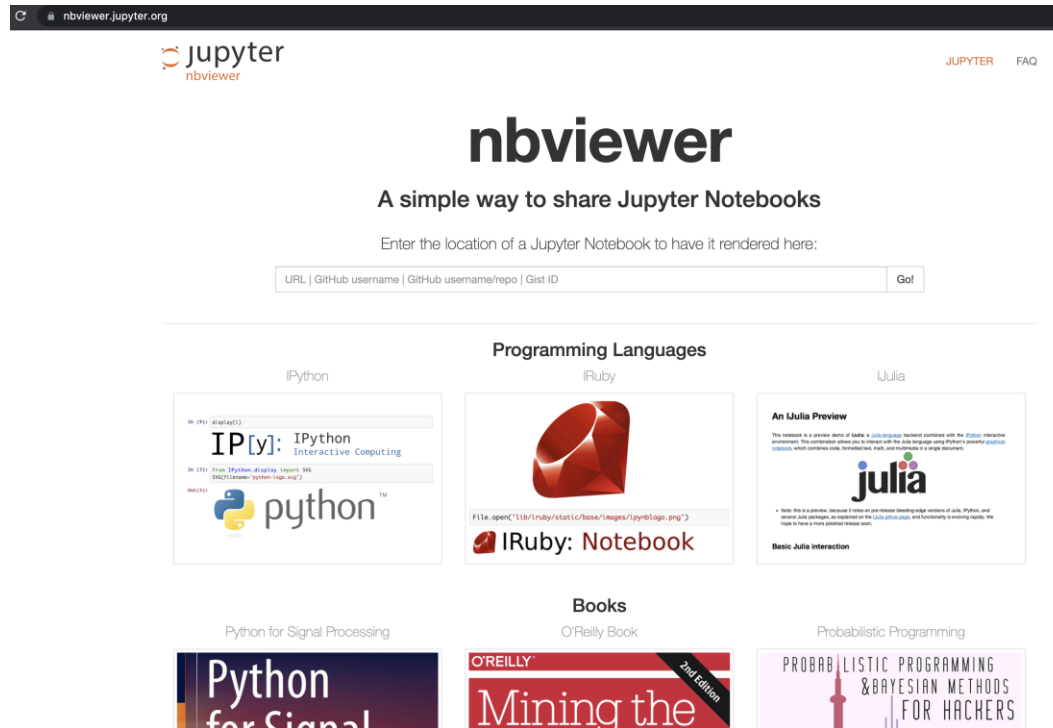
GitLab

github:gist



# Sharing Notebooks: Nbviewer

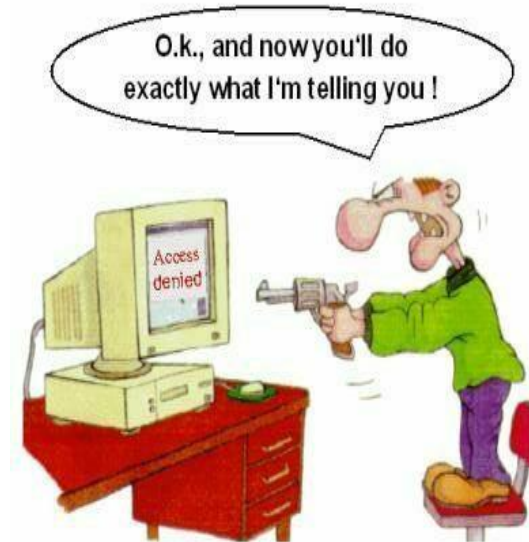
- Jupyter nbviewer is a web service that lets you enter the URL, Gist ID, GitHub username/repo/file of a Jupyter Notebook file.
- Nbviewer renders that notebook as a static HTML web page, and gives you a stable link to that page which you can share with others.





# Things To Do

- Download and install Anaconda Navigator on your system.
- Play around with the Jupyter Notebook Web Application by creating multiple notebook files and understand the concepts of kernel, code cells and markdown cells.
- Practice writing basic Python codes using Google Colab.
- Try converting the format of your created ipynb files and share them with your friends using techniques discussed in class.
- Use Jupyter Nbviewer service to view some publically available Jupyter notebooks



**Coming to office hours does NOT mean you are academically weak!**