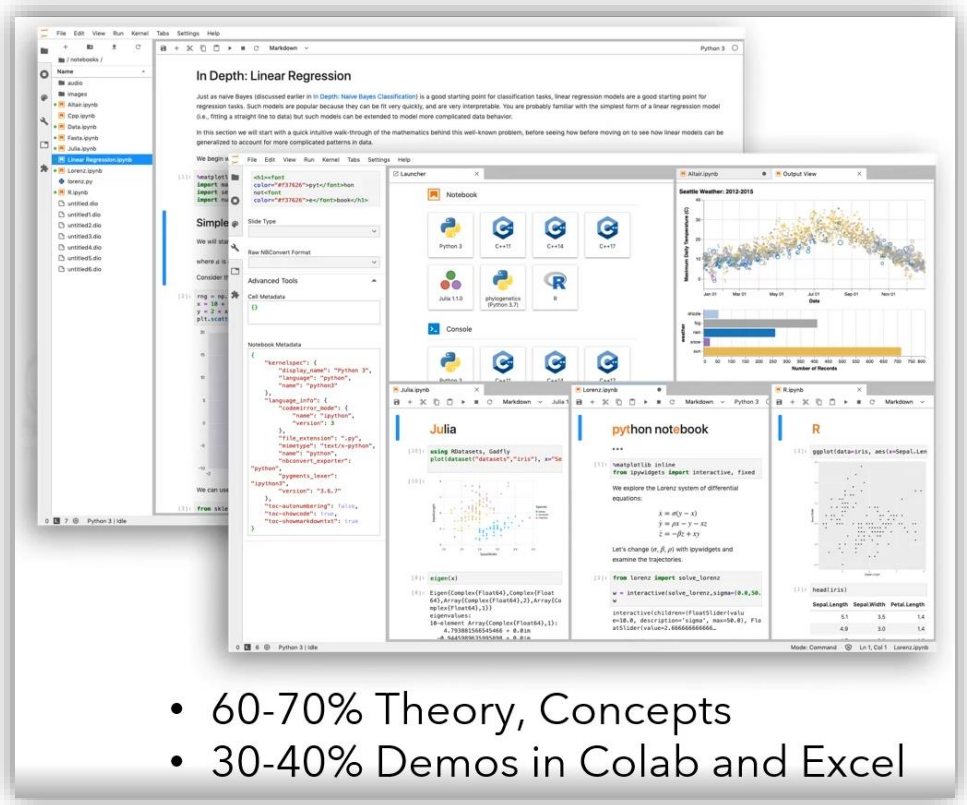
Week 03

- **3.1 Learning Methodologies**

Comments about Course Contents -

- Not reading a textbook and spitting it out
- Have lived through the contents that I am teaching – convinced myself that they are extremely useful.
- Not teaching everything under the sun – but will be a good beginning
- Hold on tight!

Learn by doing –



"You can't connect the dots looking forward. You can only connect them looking backwards" **– Steve Jobs.**

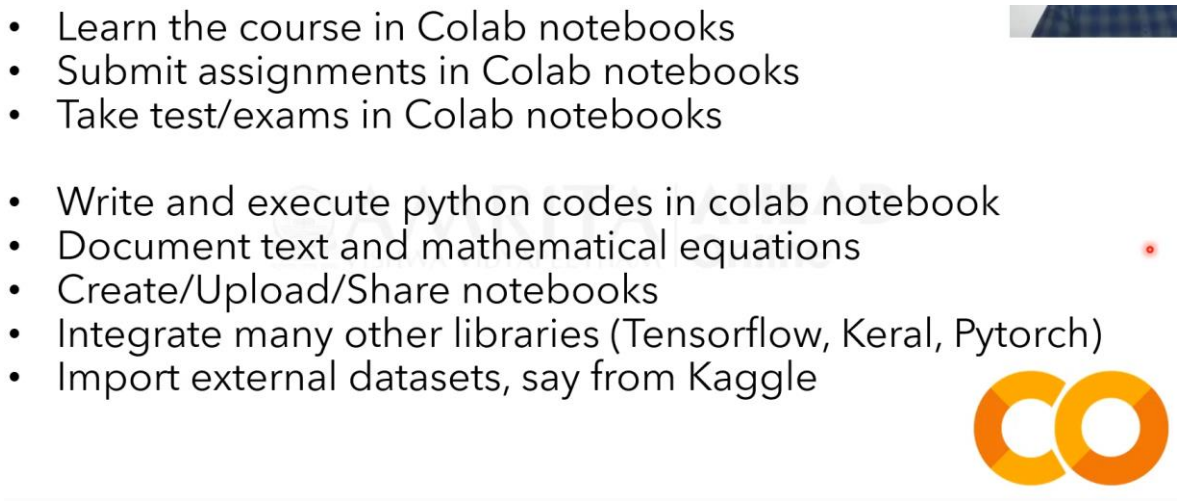
- **3.2 Learning Computational Tools**

Objective: Learn Python/Excel to conceptually learn mathematics.

Python, Jupyter and Colab

- **Python –**
  - Easy to learn
  - Very powerful
  - Favorite language for Data Science and Machine learning
  - Great support System.
- **Jupyter –**
  - Is a notebook
  - Notebook are documents that can have both codes and rich text
  - Can execute python commands
  - Makes it easier to learn mathematics
  - Share notebooks
- **Colab –**
  - Colaboratory – product from Google
  - Based on Jupyter, can execute python
  - Makes it even easier to share notebooks
  - No hassle of installation or computing resources
  - Free service

- **Colab Notebooks –**



- ## 3.3 Learning Computational Tools

Objective of this Colab Tutorial

- Many Colab tutorials on the internet
- This tutorial is designed in a different way
- Overall Objective
    1. Understand what Colab is ?
    2. Understand features of Colab
    3. How to user Colab ?
- Learn by doing
- Imagine that we have to submit an assignment in Colab
- Bring out the necessary features and methodologies in Colab.

Traditional Way – Handwritten

Typeset assignment – Printed

Microsoft word

Google Docs

The Latex Project

Learning math and typing up assignments in Colab

- Sections, sub-sections, or sub-sub-sections are used to organize the document
- They help the reader see how the material develops in the document.

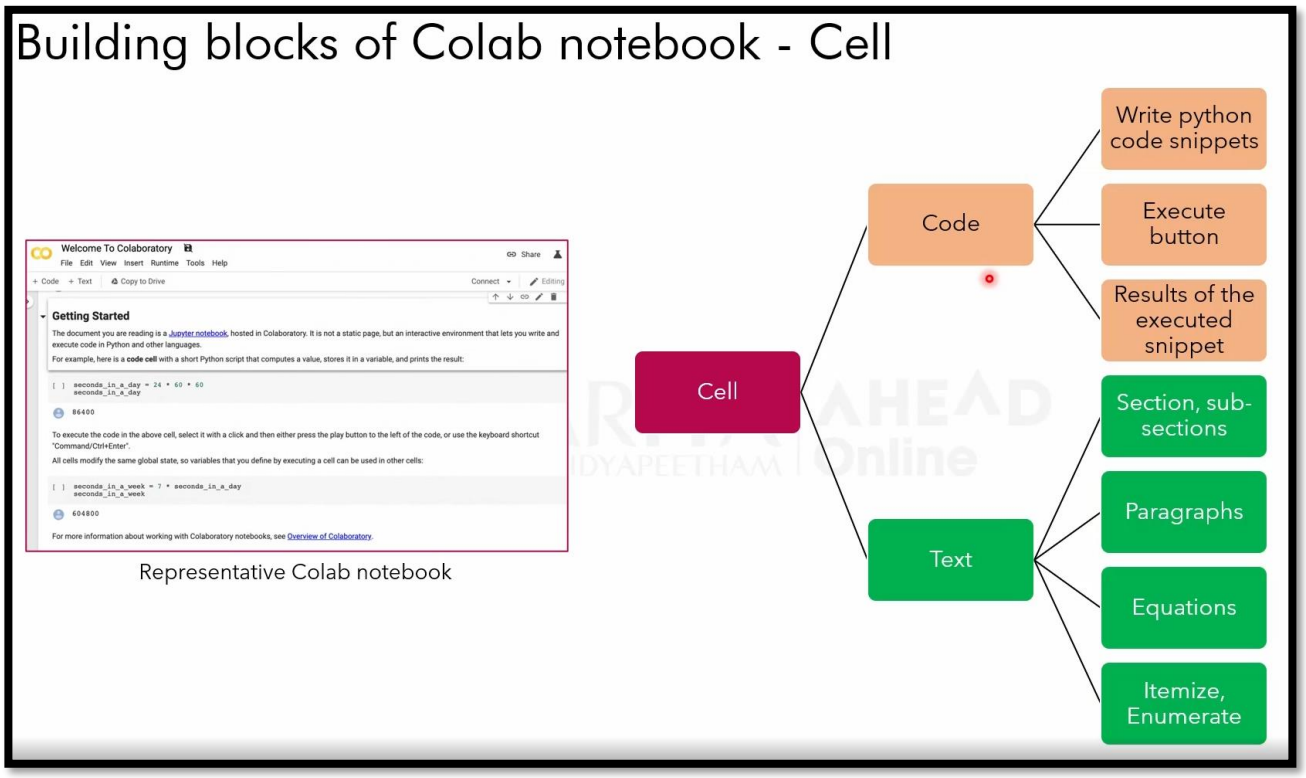
- Paragraphs
- Bulleted list – itemize
- Ordered list – enumerate

- We can perform simple calculations using python inside the Colab notebook
- We can also code up complex programs inside Colab and execute it right inside

- We can write equations beautifully in Colab
- We will be also using a python package called – Sympy that will make the process of derivation completely different and easy.

- **3.4 Learning Computational Tools**

- Extension of Colab notebook files: **ipynb**
- Stands for interactive python notebook



- Cells are classified into code cell and text cell.

- **3.5 Learning Computational Tools**

- **Markdown**

- As discussed, Colab has two types of cells: text and code
- Text cells are formatted using a simple language called Markdown.
- Use the markdown with symbol # to signal section
- Absence of any markdown defaults to paragraph
- Markdown with symbol ## (two hash symbols) to signal sub-section
- To generate bulleted list, markdown with *
- Markdown with 1.
- Note that we don’t have to be keep track of numbering of the list. Colab knows how to keep track.

- ## 3.6 Evaluating and documenting mathematical expressions

- Enclose the mathematical expression within **“$”** (before and after).
- Rendering of such mathematical expression looks elegant and differential from text.
- Mathematical expression or Python code snippets that have to executed should be written in code cell.

- In text cell for documentation, 3^2 is written as 3^2
- In code cell for evaluation, 3^2 is written as 3**2

- We will have a separate dedicated session to learn how to write (or document) mathematical expression in detail

- ## 3.7 Embedding python code in colab

- We learned earlier on enclosing the mathematical expression withing $ (before and after). Another way is to

  \begin{equation}
  Write your equation here
  \end{equation}

- We have completed making up the mockup assignment document in colab