DAY 1

- Setting up your system.
- What is Python and Why Python.
- Working with Anaconda.
- Basic Intro to Python
- Feedback - suggestions regarding what you want to see in these sessions.

Right now I believe your systems would have been setup with Python (either anaconda or without).
If not please visit this link

I have also added instructions to dual boot ubuntu with your Windows. NOTE : It's not necessary that you need to have Ubuntu/Linux installed but it's something you must definitely try out.

IN CASE ANYONE HAS ANY ISSUES TILL NOW PLEASE POST IT ON THE ISSUES TAB OF THE GITHUB REPO WITH A CLEAR DESCRIPTION OF YOUR ISSUE.

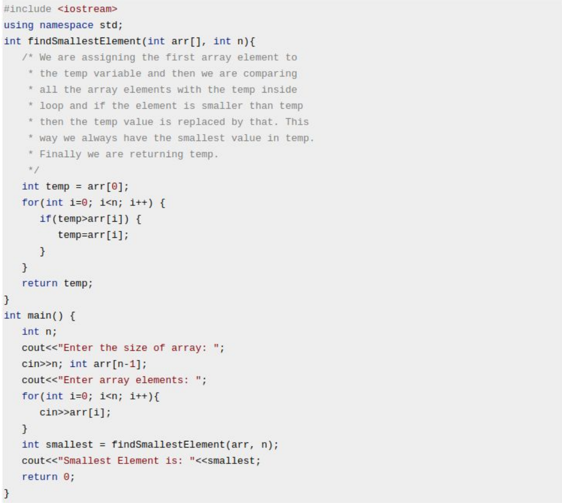
NOW LETS UNDERSTAND WHATS PYTHON AND WHAT'S THIS HYPE AROUND PYTHON FOR.

WHAT IS PYTHON:

Python is an easy to learn, powerful Object Oriented Programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming.

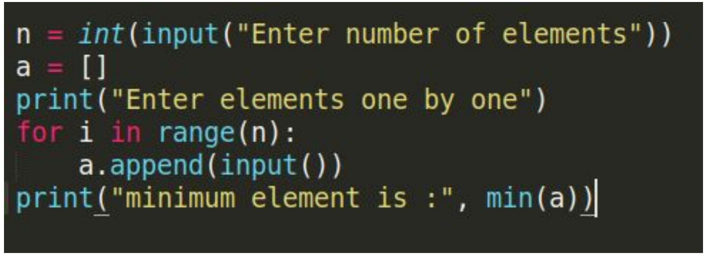
Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. This makes the code shorter, simpler and more readable.

For example to choose the minimum element of an array of n elements the c++ code is:



```cpp
#include <iostream>
using namespace std;
int findSmallestElement(int arr[], int n){
   /* We are assigning the first array element to
    * the temp variable and then we are comparing
    * all the array elements with the temp inside
    * loop and if the element is smaller than temp
    * then the temp value is replaced by that. This
    * way we always have the smallest value in temp.
    * Finally we are returning temp.
    */
   int temp = arr[0];
   for(int i=0; i<n; i++) {
      if(temp>arr[i]) {
         temp=arr[i];
      }
   }
   return temp;
}
int main() {
   int n;
   cout<<"Enter the size of array: ";
   cin>>n; int arr[n-1];
   cout<<"Enter array elements: ";
   for(int i=0; i<n; i++){
      cin>>arr[i];
   }
   int smallest = findSmallestElement(arr, n);
   cout<<"Smallest Element is: "<<smallest;
   return 0;
}
```

Whereas the same in python is just 6 lines.



```python
n = int(input("Enter number of elements"))
a = []
print("Enter elements one by one")
for i in range(n):
    a.append(input())
print("minimum element is :", min(a))
```

But…..

Python is an **interpreter** based language meaning every line in Python is executed one after the other.
This makes Python comparatively slower than C++ but it can be very useful in many ways.

Helps us execute a line of code instantly.

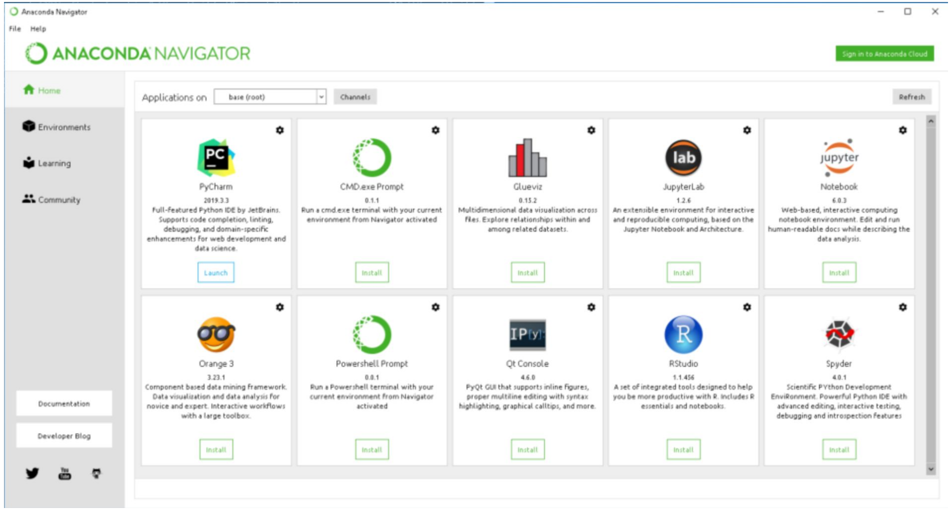
Can help in debugging as the execution of the code can be stopped at any point. This is a feature in IDE's like PyCharm and VS Code and the user can hover over the variables to view what each one contains.

Another major difference between Python and C++ is that one need not define a variable in Python since its an interpreter based language. While reading every line the Python interpreter assigns a variable a type based on the assignment. This once again helps reduce lines of code.

Another major advantage in Python is that there are a lot of libraries in Python which make our lives easier and helps reduce amount of redundant code. Eg: Numpy - for handling matrices and matrix operations, Matplotlib - for plotting, etc.

We will also be looking at these libraries as we progress through the sessions.

So let's explore the Anaconda navigator. First open Anaconda Navigator. You should see something like this:



PyCharm
2019.3.3
Full-featured Python IDE by JetBrains. Supports code completion, linting, debugging, and domain-specific enhancements for web development and data science.

CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
1.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

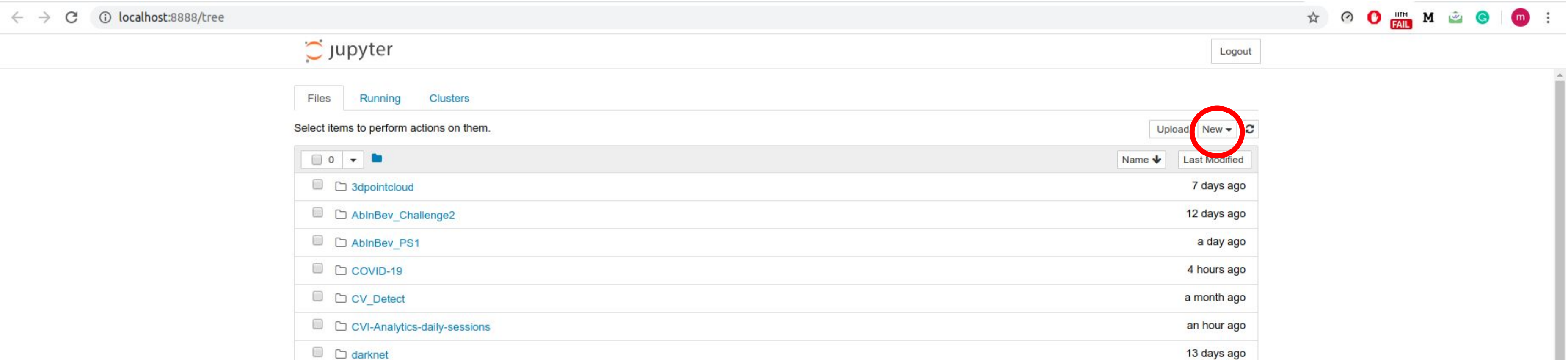
Spyder
4.0.1
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

On the main page you see the different IDE’s installed on your system. You can launch an IDE instance easily by clicking on one of them. We will be majorly using Jupyter notebooks in these sessions.

The Jupyter Notebook App is a server-client 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 (as described in this document) or can be installed on a remote server and accessed through the internet.

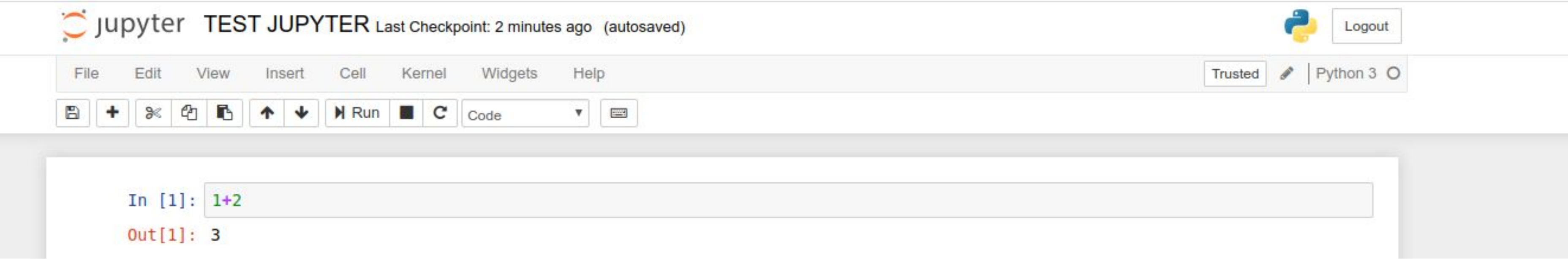
In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

Try launching a Jupyter Notebook instance and the browser will open up with a main page containing all your local files. Navigate to a specific directory and create a new Python notebook by clicking on the button on the top right



Select items to perform actions on them.

Name	Last Modified
3dpointcloud	7 days ago
AbInBev_Challenge2	12 days ago
AbInBev_PS1	a day ago
COVID-19	4 hours ago
CV_Detect	a month ago
CVI-Analytics-daily-sessions	an hour ago
darknet	13 days ago



```
In [1]: 1+2
Out[1]: 3
```

This is the jupyter notebook layout. Each Jupyter Notebook has a cell. One needs to code in these cells and execute them one by one.

Some basic shortcuts to run a cell.

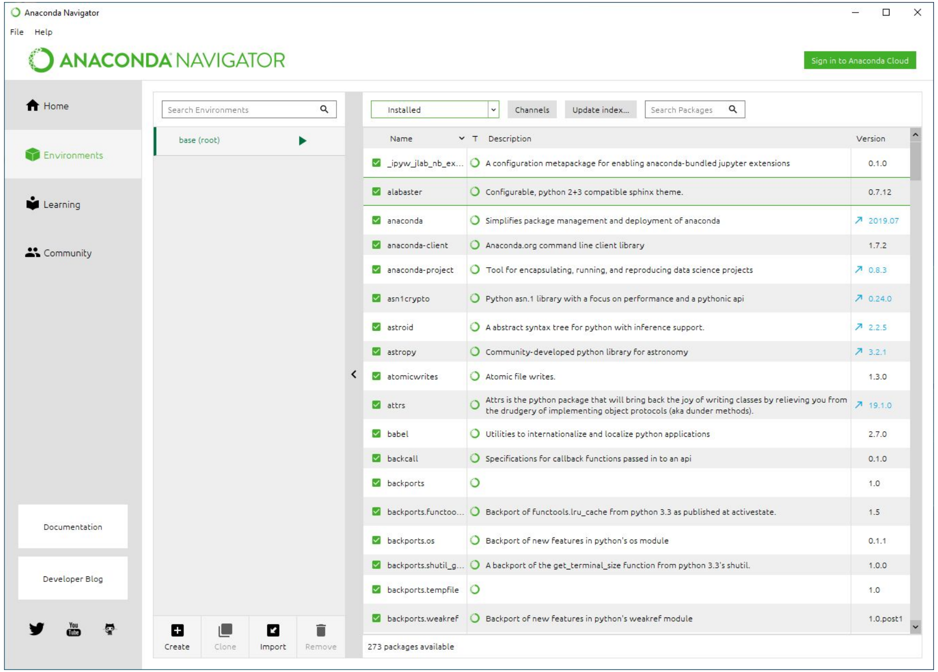
- Shift + Enter run the current cell, select below.
- Ctrl + Enter run selected cells.
- Alt + Enter run the current cell, insert below.

Please try downloading this notebook link and navigating to it from the dashboard and try executing some commands.

We will look into this in detail as we progress in these sessions.

Under the kernels tab there are a few options, try to understand what each one means.

Now let's go back to the Anaconda Navigator:
Click on the environments tab on the left and you can see the pre-installed packages Anaconda comes with.



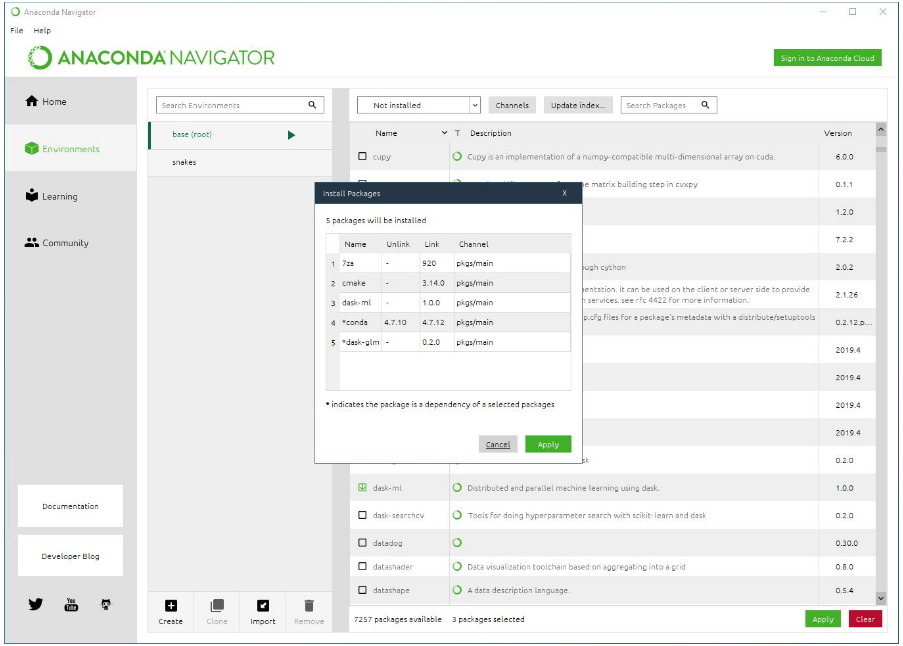
	Name	T	Description	Version
☑	_ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
☑	anaconda	○	Simplifies package management and deployment of anaconda	↗ 2019.07
☑	anaconda-client	○	Anaconda.org command line client library	1.7.2
☑	anaconda-project	○	Tool for encapsulating, running, and reproducing data science projects	↗ 0.8.3
☑	asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	↗ 0.24.0
☑	astroid	○	A abstract syntax tree for python with inference support.	↗ 2.2.5
☑	astropy	○	Community-developed python library for astronomy	↗ 3.2.1
☑	atomicwrites	○	Atomic file writes.	1.3.0
☑	attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	↗ 19.1.0
☑	babel	○	Utilities to internationalize and localize python applications	2.7.0
☑	backcall	○	Specifications for callback functions passed in to an api	0.1.0
☑	backports	○		1.0
☑	backports.functoo...	○	Backport of functools.lru_cache from python 3.3 as published at activestate.	1.5
☑	backports.os	○	Backport of new features in python's os module	0.1.1
☑	backports.shutil_g...	○	A backport of the get_terminal_size function from python 3.3's shutil.	1.0.0
☑	backports.tempfile	○		1.0
☑	backports.weakref	○	Backport of new features in python's weakref module	1.0.post1

273 packages available

So what is an environment :

A conda environment is a directory that contains a specific collection of conda packages that you have installed. So let's say you have NumPy 1.7 installed in one environment and you want to test out an older version, then one creates a new environment and installs the necessary version. Every environment is independent of one another and this becomes very handy for legacy testing, etc. One can easily switch between environments, delete them, etc. Currently you are in your base environment with all the pre-installed packages.

To install a new package go to the search bar on the right and enter and a package name (lets say keras for example) and choose the not installed category and you can see the package. Now select this package and click on apply.



Install Packages

5 packages will be installed

	Name	Unlink	Link	Channel
1	7za	-	920	pkgs/main
2	cmake	-	3.14.0	pkgs/main
3	dask-ml	-	1.0.0	pkgs/main
4	*conda	4.7.10	4.7.12	pkgs/main
5	*dask-glm	-	0.2.0	pkgs/main

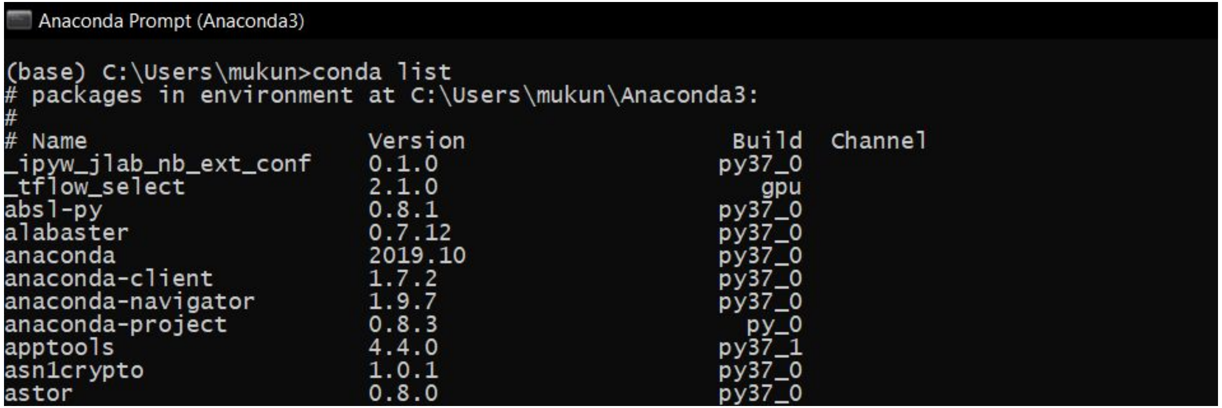
* indicates the package is a dependency of a selected packages

Cancel Apply

Name	T	Description	Version
□ cupy	○	Cupy is an implementation of a numpy-compatible multi-dimensional array on cuda.	6.0.0
☑ dask-ml	○	Distributed and parallel machine learning using dask.	1.0.0
□ dask-searchcv	○	Tools for doing hyperparameter search with scikit-learn and dask	0.2.0
□ datadog	○		0.30.0
□ datashader	○	Data visualization toolchain based on aggregating into a grid	0.8.0
□ datashape	○	A data description language.	0.5.4

7257 packages available 3 packages selected

One can also do this via the anaconda prompt. So let's open the anaconda navigator command prompt for windows and for linux users the terminal itself.



```
(base) C:\Users\mukun>conda list
# packages in environment at C:\Users\mukun\Anaconda3:
#
# Name                    Version                   Build  Channel
_ipyw_jlab_nb_ext_conf    0.1.0                    py37_0
_tflow_select             2.1.0                       gpu
absl-py                   0.8.1                    py37_0
alabaster                 0.7.12                   py37_0
anaconda                  2019.10                  py37_0
anaconda-client           1.7.2                    py37_0
anaconda-navigator        1.9.7                    py37_0
anaconda-project          0.8.3                      py_0
apptools                  4.4.0                    py37_1
asn1crypto                1.0.1                    py37_0
astor                     0.8.0                    py37_0
```

This is the anaconda command prompt. You can see that we are in the base environment. Typing **conda list** prints out the entire list of packages installed in the current environment. One can also install a package from here by typing **conda install <package-name>.**

There are various other commands which you can check out.

That's it for today I guess we will be exploring different python libraries as we progress through these sessions. We will be updating the repo daily so please make sure you check them out regularly and make best use of this vacation period.

Please do fill this feedback form for specific content you would like to see during these sessions:
https://forms.gle/8G9j9C6BbFW5FFiZA