Session – 14 : Introduction to python using jupyter notebook interface
12-09-2019, 09:00 – 10:00 Hrs, RJN 302

[1] Installation:

You have multiple options while installing python and jupyter notebook to get going on your laptop. The options are as follows:

1. Install python and jupyter-notebook using “`sudo apt-get install python`” etc.,
2. Install anaconda. It comes with all things python for you to use.
3. Install sagemath. It comes with python and jupyter included.

Watch out if you are installing these using a virtual machine. The bundle of sagemath will take almost 8 GB as it has a whole bunch of things inside it. If you don’t have enough space in the virtual disk, consider using a shared folder feature on the Oracle VirtualBox to install. You can create a folder in one of your drives in windows called, say, “LinuxShare” and ask VirtualBox to share it as a permanent, auto-mount share to the virtual box with a specific mount path such as “/opt/LinuxShare”. When you restart the VirtualBox, your share will be available from within the linux OS and you can use it to store the folders of sage and such heavy duty software.

Alternatively, you can create another VDI disc on your host computer, make it available to the virtual machine, mount it from within the virtual machine at a mount path and use it. This route is faster but limited again by the size of the VDI disc you chose to create in the first place.

[2] Launching

If you have installed jupyter-notebook natively, then you can lauch it from your home folder simply by running “`jupyter-notebook`”. If you get an error abour socket or IP address, try running it as “`jupyter-notebook --ip=127.0.0.1`”.

If you have installed sage, then launch it from your home folder using the command where the “`/path/to/sage`” is where you unzipped the sage binary after downloading it. It is a good idea to keep it in a path like “`/opt/sage-8.8/`”.

```
/opt/sage-8.8/sage --notebook=jupyter --ip=127.0.0.1
```

Once the jupyter notebook opens in the browser, you can create a new notebook, choose the kernel to be python 3 and you are ready to use python from within this notebook.

[3] Introduction to jupyter notebook

The two major types of cells in the jupyter notebook are called “code” and “markdown”. You can click on a cell and use the menu options to convert cells of one type to other. Use the cells of type “code” to type in your python code. Use the cells of type “markdown” to type in your documentation. The markdown syntax is illustrated in the Sandbox file uploaded on the moodle page.

Cells of type “code” will get numbered chronologically as you keep entering new ones. You can delete cells and move them up and down as needed.

Whenever you open a notebook to continue working from your previous sitting, make sure you “run all the cells” using the menu option under “cells” to ensure all the variables etc., are available in the memory. You don’t have to worry about this as long as you continue working on the notebook without closing it.

Jupyter will save your notebook once a while. It is a good idea to click on the save button and name the file meaningfully. The files will be with an ending “.ipynb” which are pure text files. You can open them using vi editor in a terminal and edit if you need to. Don’t do it unless you are sure of the syntax. Look at the last few lines of this file, you will notice metadata that says what language of script is being stored in the cells of that file etc.,

You can share the “ipynb” files to share your notebooks with others. The format is fairly universal and will work on most machines without any change. Sometimes if the corresponding language is not installed, the notebook file will still show the content but may not be able to let you modify it.

[4] Introduction to python using jupyter notebook.

Copy the `Session-14.ipynb` file from the moodle page to see what is covered in the class. Some description is already there in the notebook itself. You can edit this notebook from within the jupyter notebook of your machine to explore more.

Homework

[1] Edit the jupyter notebook to explore a mapping of all data types of C language with the corresponding ones of python.

[2] Figure out what data types are available in python that are not readily available from C.