

Instructions for using Jupyter Notebooks and Assignments

For this course, all your assignments will be provided to you in the form of Jupyter Notebooks, and they must be submitted as Notebooks as well (unless mentioned otherwise). You can utilise either Google Colab for working on them with your teammate or install Jupyter Notebook on your local machine. Below, we provide you with some instructions to help you get started.

1 Google Colab

It is possible to create, upload, download, store, and share notebooks using Colab. You can also mount your Google Drive to use the data stored there. These notebooks can also be downloaded as .py files and PDFs.

1.1 Getting started

1. To use Colab, you need to have a Google account. Go to colab.research.google.com and sign in with your Google account. A folder called 'Colab Notebooks' will be created in your Drive.
2. You can create a new Notebook by clicking **File** → **New Notebook** on the Colab website, or from your Drive by selecting **New** → **More** → **Google Colaboratory**.
3. You can rename your Notebook in the top left corner:

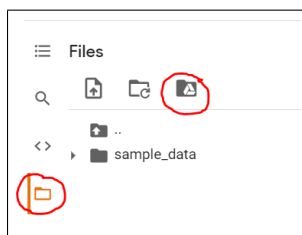


1.2 Other functionality

1. You can change the Runtime by clicking on **Runtime** → **Change runtime type** and select the appropriate runtime and hardware accelerator. (Not required at the moment.)
2. **Mounting the Drive:** You can use the data stored on the Drive by mounting it in your Colab environment. To do this, either run the following command in your Notebook cell

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Or, you can simply select the folder icon as shown below and mount the drive:



3. Most system based commands can be directly run from the Notebook by prepending them with an `!` symbol. This allows on-the-fly installation and cloning functionalities.

```
!pip install numpy
```

4. For further details and usage, refer to analyticsvidhya.com/blog/2020/03/google-colab-machine-learning-deep-learning/

2 Local Machine

You can also use Jupyter Notebooks on your local machine. To do this, follow the given steps:

2.1 Set up Anaconda

Most installations recommend using the conda package manager which makes it easy to set up environments and install all necessary packages. For this, you need to download and install Anaconda.¹ Note that you need to use the Anaconda prompt if you're using Windows, or make sure you add the Anaconda file path to your system environment variables.

Launch Jupyter Notebook in the current file path by typing:

```
$ jupyter notebook
```

This will open a new Jupyter Notebook instance in your browser. All the files in your current directory will be visible in this instance.

2.2 Set up conda environment

Since you might require specific packages and versions for your assignments, we recommend that you create a separate environment for this course. Specify the base packages while creating your environment:

```
$ conda create -n snlp python=3.7 numpy matplotlib
```

Or, you can install packages separately in your environment using the command

```
$ conda install <package-name>
```

You can activate / deactivate your environment with the commands

```
$ source activate snlp
```

```
$ source deactivate
```

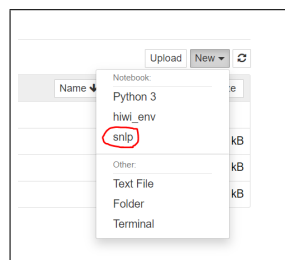
To add your environment to your Jupyter Notebook, you need to install ipykernel using

```
$ conda install -c anaconda ipykernel
```

After installing, just run

```
$ python -m ipykernel install --user --name=snlp
```

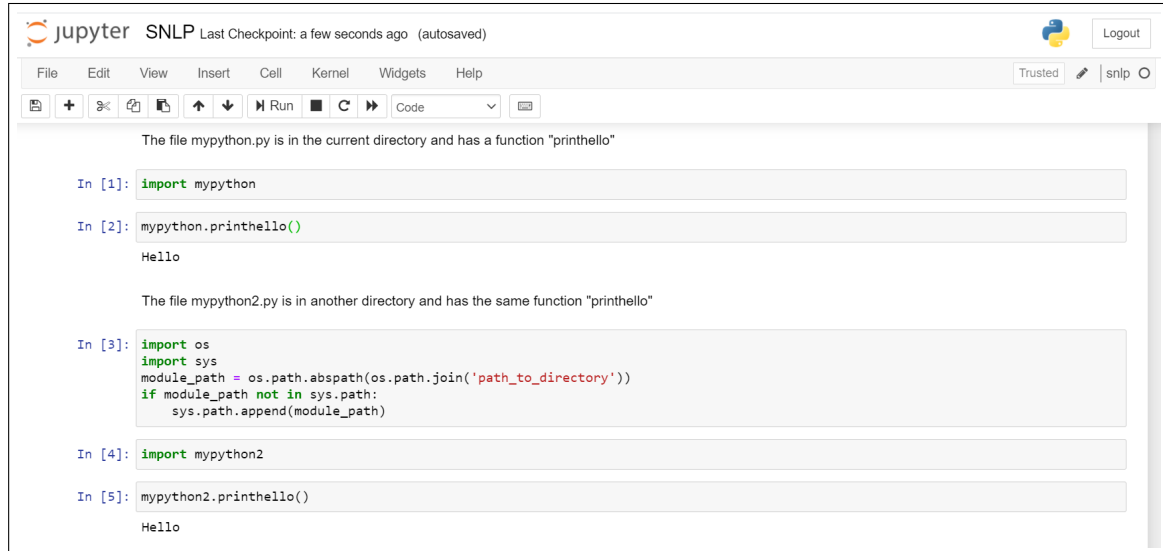
You can now create a new Notebook in that environment:



¹anaconda.com/products/individual

3 Importing Python code in Notebook

We mostly require you to answer only textually in the notebook and to store your Python code in pure .py files. For that, you have to import the file and call the functions from the Notebook. If your Python file is saved in a directory other than your current directory, you can import it to your Notebook by adding it to your system path:



However, once your Python file has been imported into the Notebook, if you make changes to this code, the Notebook does not reimport these changes. One way to overcome this is to restart the kernel and rerun the Notebook cells. Another way is to utilise Python's importlib module in the Notebook as follows:

```
from importlib import reload
import mypython2
mypython2 = reload(mypython2)
```

You can also connect Google Colab to your local runtime, which allows you to execute code on your local hardware and have access to your local file system. For this, you need to install Jupyter to your local machine, as indicated in Section 2.1.

1. You also need to install and enable the `jupyter_http_over_ws` Jupyter extension.

```
$ pip install jupyter_http_over_ws
$ jupyter serverextension enable --py jupyter_http_over_ws
```

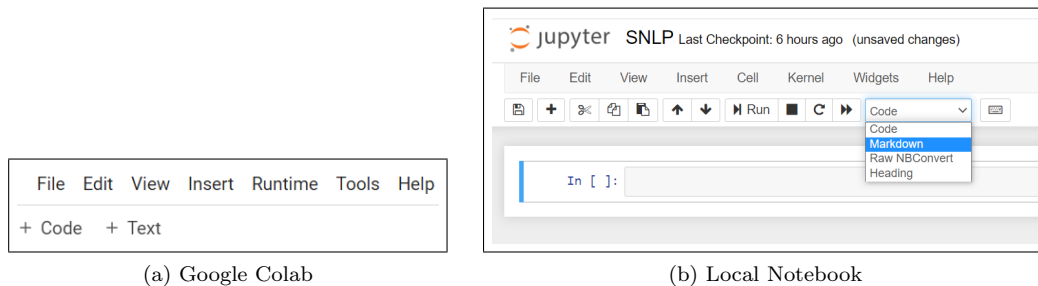
2. To start the server, you need to set a flag to explicitly trust WebSocket connections from the Collaboratory frontend.

```
$ jupyter notebook \
--NotebookApp.allow_origin='https://colab.research.google.com' \
--port=8888 \
--NotebookApp.port_retries=0
```

4 Other functionalities of Notebooks

For most assignments, you will be required to write a text-based answer or analysis along with the coding solution. To avoid utilising multiple file formats, you are expected to write your analyses in the same Notebook. For this, we ask you to get acquainted with certain other features provided by Jupyter Notebooks, such as using Markdown for text, LaTeX for mathematical equations etc.

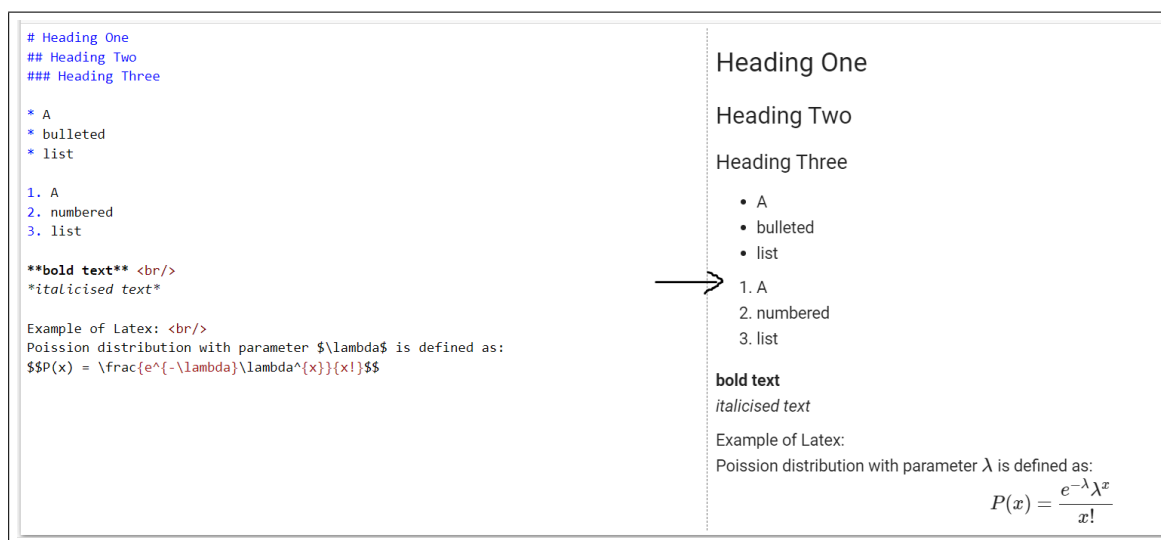
This can be done in Google Colab by simply creating the cell as a **Text** cell instead of a **Code** cell. In your local runtime, you can toggle the cell type in the dropdown menu.



(a) Google Colab

(b) Local Notebook

An example of markdown and LaTeX renderings in the code cell:



Please refer to the following link for more examples:
towardsdatascience.com/write-markdown-latex-in-the-jupyter-notebook-10985edb91fd

5 Submission

As you will be working in teams of two, please name your Notebook submission in the format:

Name1_id1_Name2_id2.ipynb

Wherever you have to submit more than one file, you should place them all in one folder and submit the zipped contents. The submitted folder should also follow the same naming template. **If you do not follow the formal requirements, the tutors can decide not to grade your submission.**

You will be explicitly informed separately in each assignment what files to submit, and what format they should adhere to. Please make sure you follow those instructions consistently across all the assignment submissions. Also, when you submit your assignments in Teams, you must remember to upload the submission **AND** click on the “Turn-in” button at the top right hand corner. Otherwise your assignment will not be considered as submitted.

Assignments will be released at 23:59 on the day of the lecture (i.e. every Friday). This is also the deadline for the previous week’s assignment.

If you have any problems with your system setup, as well as clarifications about the assignments, first check Piazza² for similar posts, and only then create a new question, which we will do our best to answer as soon as possible. You are free (and encouraged) to answer other people’s questions if you know the answer.

²piazza.com/class?nid=kniwgq42qek5oe