

Welcome to NB theme!

Rulebook

Task 0

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1

Task 2

Task 3

Task 4

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog



eYRC 2020-21: Nirikshak Bot (NB)

## Task 0

# Software Installation (for Ubuntu OS)

[ Last Updated on: **5th October 2020, 20:34 Hrs** ]

This document contains instructions to install following software/libraries on **Ubuntu OS**:

- **Miniconda for Python 3.8**
- **OpenCV**
- **NumPy**
- **Matplotlib**
- **Jupyter Notebook, JupyterLab**
- **CoppeliaSim 4.0.0**
- **VS Code** (*optional but highly recommended*)

The installation of all software/libraries has been tested on **Ubuntu 16.04** and **18.04**. We recommend you to use one of these versions of **Ubuntu OS**. These software have to be installed **ONLY ON 64-bit OS**.

- 1. Miniconda for Python 3.8
  - A. Installation Steps
  - B. Verification Steps
- 2. OpenCV, NumPy, Matplotlib, Jupyter Notebook, JupyterLab
  - A. Setting up a Virtual Environment using Miniconda
  - B. Verifying the newly created Virtual Environment
  - C. Using the packages and tools
- 3. CoppeliaSim 4.0.0
  - A. Installation Steps
- 4. Visual Studio Code (optional but highly recommended)
- 5. Testing the Setup
- References

Welcome to NB theme!

Rulebook ›

Task 0 ~

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

## 1. Miniconda for Python 3.8

- In this section, we will be installing Miniconda and use it to create a Python 3.8 virtual environment in next section.
- We will be using **Python 3.8** in this theme. You don't have to install Python separately. It come by default with Miniconda installation.
- Conda is a package management system, created specifically to assist working with Python packages in a cross-platform fashion. Conda also has the concept of a virtual environment, which is an independent, self-contained install of Python and packages.
- There are two options for getting Conda: Anaconda and Miniconda. You can read in detail about these and which one to prefer, over here: [reference1](#) and [reference2](#).

**Note:** If you already have Anaconda or Miniconda pre-installed in your system, kindly skip this section.

### A. Installation Steps

- Download Miniconda for Python 3.8 from [here](#) (file size - 88.7MB).
- We need to make this downloaded shell (.sh) file executable. Right-click on the folder where this file is present (for instance, **Downloads** folder) and select **Open in Terminal** option.
- Run the following commands to start the installation.

```
sudo chmod u+x Miniconda3-latest-Linux-x86_64.sh
./Miniconda3-latest-Linux-x86_64.sh
```



- To continue the installation process, hit **Enter**. Accept the license terms by typing in **yes**. Confirm the default location of installation by press **Enter**. It will install the basic necessary libraries and tools.
- Type in **yes** when asked by installer to initialize **Miniconda3**.
- Once the installation completes, you will be greeted with message: **Thank you for installing Miniconda3!**.
- Now, run the following commands to complete some post-installation steps:

Welcome to NB theme!

Rulebook

Task 0

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1

Task 2

Task 3

Task 4

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

```
source ~/.bashrc
conda config --set auto_activate_base false
source ~/.bashrc
```

```
eyrc@erts: ~/Downloads
File Edit View Search Terminal Help

Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes/no]
[no] >>> yes
./Miniconda3-latest-Linux-x86_64.sh: 486: ./Miniconda3-latest-Linux-x86_64.sh: [: not found
no change      /home/eyrc/miniconda3/condabin/conda
no change      /home/eyrc/miniconda3/bin/conda
no change      /home/eyrc/miniconda3/bin/conda-env
no change      /home/eyrc/miniconda3/bin/activate
no change      /home/eyrc/miniconda3/bin/deactivate
no change      /home/eyrc/miniconda3/etc/profile.d/conda.sh
no change      /home/eyrc/miniconda3/etc/fish/conf.d/conda.fish
no change      /home/eyrc/miniconda3/shell/condabin/Conda.psm1
no change      /home/eyrc/miniconda3/shell/condabin/conda-hook.ps1
no change      /home/eyrc/miniconda3/lib/python3.8/site-packages/xontrib/conda.xsh
no change      /home/eyrc/miniconda3/etc/profile.d/conda.csh
no change      /home/eyrc/.bashrc
No action taken.
If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniconda3!
eyrc@erts:~/Downloads$ source ~/.bashrc
(base) eyrc@erts:~/Downloads$ conda config --set auto_activate_base false
(base) eyrc@erts:~/Downloads$ source ~/.bashrc
eyrc@erts:~/Downloads$
```

Figure 1: Miniconda Installation Complete

- You will see output similar to Figure 1. Once the installation completes, it registers **conda** as new command to the default **bash** script ( **bashrc** file).
- So, what we did in post-installation steps is:
  - First command invoked this default bash script so that we can use **conda** command further or in other words, this command makes the changes in **bashrc** to take effect.
  - Miniconda creates a default Python environment named **base** and the previous step activates this environment by default. So with second command, we set the flag for automatic activation of **base** environment to **false**; else for each new Terminal you open the base environment would be activated.
  - Third command, again invokes the default bash script for changes in it to take effect. And now you can see just after this command, the **(base)** has disappeared.
- Cheers, **Miniconda for Python 3.8** is successfully installed. Let's verify now.

Welcome to NB theme!

Rulebook ›

Task 0 ▾

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

## B. Verification Steps

- Open a new Terminal or continue in the previous one.

- Run the below command:

```
conda info
```



You will get the information on Miniconda as in Figure 2.

```

eyrc@erts: ~
File Edit View Search Terminal Help
eyrc@erts:~$ conda info

      active environment : None
            shell level : 0
        user config file : /home/eyrc/.condarc
populated config files : /home/eyrc/.condarc
         conda version : 4.8.3
    conda-build version : not installed
         python version : 3.8.3.final.0
    virtual packages : __glibc=2.27
      base environment : /home/eyrc/miniconda3 (writable)
        channel URLs : https://repo.anaconda.com/pkgs/main/linux-64
                      https://repo.anaconda.com/pkgs/main/noarch
                      https://repo.anaconda.com/pkgs/r/linux-64
                      https://repo.anaconda.com/pkgs/r/noarch
         package cache : /home/eyrc/miniconda3/pkgs
                       /home/eyrc/.conda/pkgs
        envs directories : /home/eyrc/miniconda3/envs
                       /home/eyrc/.conda/envs
               platform : linux-64
             user-agent : conda/4.8.3 requests/2.23.0 CPython/3.8.3 Linux/5.4.0-47-generic
ubuntu/18.04.5 glibc/2.27
                UID:GID : 1001:1001
                netrc file : None
            offline mode : False

eyrc@erts:~$
```

Figure 2: Output of conda info command

- As mentioned earlier, Miniconda creates a default Python environment named **base** at the installation directory, let's check this out. Type **one of the following three** commands to find all the existing virtual environments on your system with their path location:

```
conda env list
```



OR

```
conda info --envs
```



OR

Welcome to NB theme!

Rulebook ›

Task 0 ~

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

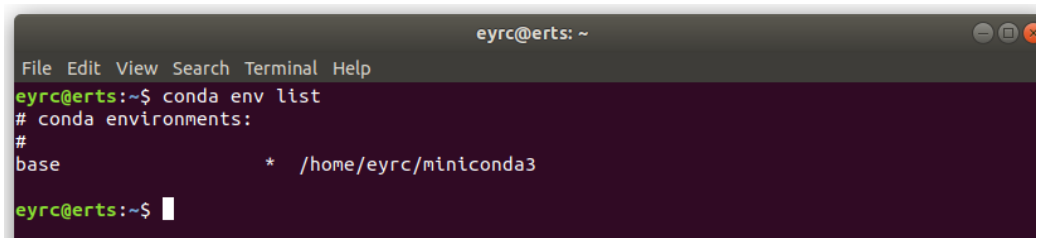
Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

conda info -e

Since for now we just have one default environment **base** existing, there is one entry in Figure 3. The **asterisk** sign " \* ", against the name indicates the current active environment, so it is **base** in this case.



```

eyrc@erts: ~
File Edit View Search Terminal Help
eyrc@erts:~$ conda env list
# conda environments:
#
base                * /home/eyrc/miniconda3
eyrc@erts:~$

```

Figure 3: Output of conda env list command

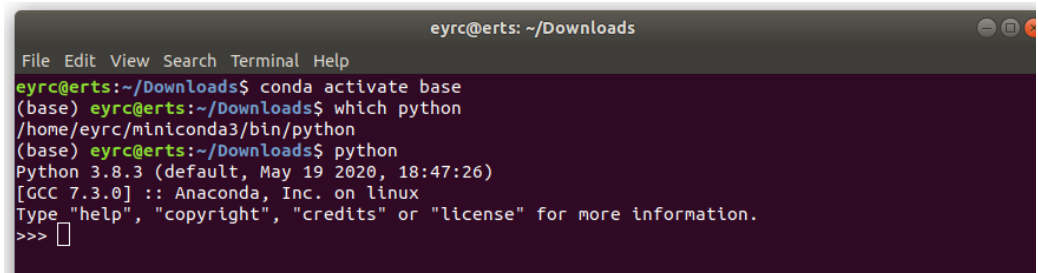
- When any environment is activated, its name appears in round braces (**env\_name**) at the start of line. To activate the default Python environment that comes with Miniconda installation, type in following in sequence:

conda activate base

which python

python

You will see output similar to Figure 4. What we did here is, first we invoked the default environment, then checked whether the prompt is able to detect the path of Python that came installed with Miniconda and at last we called the Python interpreter or console, you can see the very first message has Anaconda word which justifies it.



```

eyrc@erts: ~/Downloads
File Edit View Search Terminal Help
eyrc@erts:~/Downloads$ conda activate base
(base) eyrc@erts:~/Downloads$ which python
/home/eyrc/miniconda3/bin/python
(base) eyrc@erts:~/Downloads$ python
Python 3.8.3 (default, May 19 2020, 18:47:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

Figure 4: Output of where python and python

- The Python version (**3.8.3**) might be different at your end but that's okay as long as it is **3.8.x**. Type **exit()** to come out of Python console.
- Just as we use the command: **conda activate env\_name** to activate the Python environment named **env\_name**; to deactivate any active environment, type: **conda deactivate**. The

Welcome to NB theme!

Rulebook ›

Task 0 ▾

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

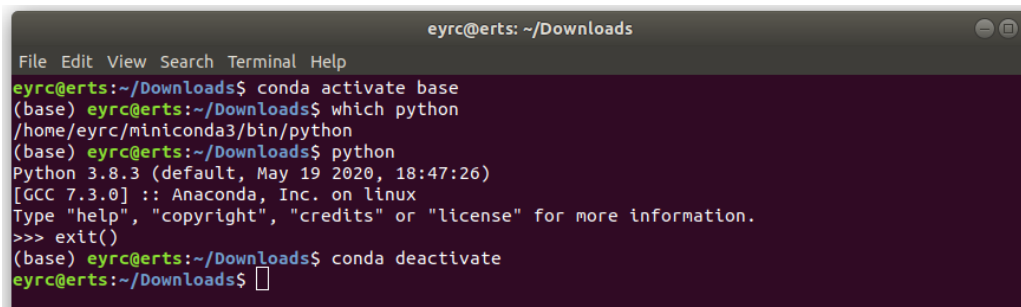
Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

(**env\_name**) will now disappear from the start of line. You will see output similar to Figure 5.

conda deactivate



```

eyrc@erts: ~/Downloads
File Edit View Search Terminal Help
eyrc@erts:~/Downloads$ conda activate base
(base) eyrc@erts:~/Downloads$ which python
/home/eyrc/miniconda3/bin/python
(base) eyrc@erts:~/Downloads$ python
Python 3.8.3 (default, May 19 2020, 18:47:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
(base) eyrc@erts:~/Downloads$ conda deactivate
eyrc@erts:~/Downloads$
  
```

Figure 5: Output of conda deactivate command

## 2. OpenCV, NumPy, Matplotlib, Jupyter Notebook, JupyterLab

- In this section, let's create a virtual environment of **Python 3.8** with all the necessary package
- We will then see how to use these packages or tools in brief.

### A. Setting up a Virtual Environment using Miniconda

- The necessary packages or libraries, tools are encapsulated into the **environment.yml** file which can be downloaded from [here](#) (file size - 174 bytes). Right-click on the hyperlink and select **Save Link As...** option to download.
- In the Terminal, navigate to the folder where the **environment.yml** file is downloaded. Create a new environment of the name **NB\_<Team-ID>** by typing the command:

```
conda env create --name NB_<Team-ID> -f environment.ymlbash
```

Replace **<Team-ID>** with your **Team ID**. **Note:** There should be no space in the environment name.

- For instance, if your Team ID is **9999**, type:

```
conda env create --name NB_9999 -f environment.yml
```

Welcome to NB theme!

Rulebook ›

Task 0 ~

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

You will see the output similar to Figure 6.

**Note:** Make sure you have good Internet connection while running this command, as it downloads and installs all packages listed in the environment.yml file.

```

eyrc@erts: ~/Downloads
File Edit View Search Terminal Help

Please update conda by running

    $ conda update -n base -c defaults conda

Downloading and Extracting Packages
ffmpeg-4.3.1           | 83.3 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: / b'Enabling notebook extension jupyter-js-widgets/extension...\n    - Validating: \x1b[32mOK\x1b[0m\n'
done
#
# To activate this environment, use
#
#     $ conda activate NB_9999
#
# To deactivate an active environment, use
#
#     $ conda deactivate

eyrc@erts:~/Downloads$

```

Figure 6: Output of conda env create

## B. Verifying the newly created Virtual Environment

- Let's verify the previous step. Type: `conda env list` now and you should now expect the environment name `NB_<Team-ID>` listed in the output as in Figure 7.

```

eyrc@erts: ~/Downloads
File Edit View Search Terminal Help

eyrc@erts:~/Downloads$ conda env list
# conda environments:
#
base                  *  /home/eyrc/miniconda3
NB_9999               /home/eyrc/miniconda3/envs/NB_9999

eyrc@erts:~/Downloads$

```

Figure 7: Output of conda env list after environment creation

Welcome to NB theme!

Rulebook ›

Task 0 ˘

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog



- The **environment.yml** file used to setup the virtual environment has following packages, tool encapsulated:
  - **OpenCV, NumPy, Matplotlib** - these packages are useful for Image Processing, Array manipulations and Plotting.
  - **Jupyter Notebook, JupyterLab** - these tools extend the console-based approach to interactive computing, providing a web-based application suitable for developing, documenting and executing code, as well as communicating the results.
- You can find more commands and know how to install a new package inside the environment clone the environment, etc. in the [Conda Cheat Sheet](#).

## C. Using the packages and tools

- **OpenCV, NumPy, Matplotlib**
  - Activate the environment **NB\_9999** in Terminal, and invoke the Python interpreter.
  - Enter the following commands:

```
conda activate NB_9999
python
```



```
import cv2
cv2.__version__
import numpy
numpy.__version__
import matplotlib
matplotlib.__version__
exit()
```



- You will get the version numbers of each package in the output as shown in Figure 8. The version numbers of some packages at your end **maybe** different.
- Once you get the similar results, exit the Python interpreter by typing **exit()**.



Welcome to NB theme!

Rulebook ›

Task 0 ▾

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

```

eyrc@erts: ~/Downloads
File Edit View Search Terminal Help
eyrc@erts:~/Downloads$ conda activate NB_9999
(NB_9999) eyrc@erts:~/Downloads$ python
Python 3.8.5 | packaged by conda-forge | (default, Sep 24 2020, 16:55:52)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.4.0'
>>> import numpy
>>> numpy.__version__
'1.19.1'
>>> import matplotlib
>>> matplotlib.__version__
'3.3.2'
>>> exit()
(NB_9999) eyrc@erts:~/Downloads$

```

Figure 8: Output of version numbers of Python, NumPy, Matplotlib

#### • Jupyter Notebook, JupyterLab

- Make sure the environment **NB\_9999** is activated before you proceed.
- To ensure our Jupyter Notebook has access to the newly created Virtual Environment, we need to execute the following:

```
python -m ipykernel install --user --name=NB_9999
```

- Now let's make sure our new environment is accessible through Jupyter Notebook.
- Launch the **Jupyter Notebook** with the command:

```
jupyter-notebook
```

OR

```
jupyter notebook
```

This will start the Notebook server and open a default web browser to the URL of the web application, which will be: **http://127.0.0.1:8888** OR **http://localhost:8888** as below in Figure 9.

Welcome to NB theme!

Rulebook ›

Task 0 ▾

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

```

eyrc@erts: ~/Downloads
File Edit View Search Terminal Help
(NB_9999) eyrc@erts:~/Downloads$ python -m ipykernel install --user --name=NB_9999
Installed kernelspec NB_9999 in /home/eyrc/.local/share/jupyter/kernels/nb_9999
(NB_9999) eyrc@erts:~/Downloads$ jupyter notebook
[I 16:45:20.007 NotebookApp] JupyterLab extension loaded from /home/eyrc/miniconda3/envs/
/NB_9999/lib/python3.8/site-packages/jupyterlab
[I 16:45:20.008 NotebookApp] JupyterLab application directory is /home/eyrc/miniconda3/e
nvs/NB_9999/share/jupyter/lab
[I 16:45:20.010 NotebookApp] Serving notebooks from local directory: /home/eyrc/Downloa
ds
[I 16:45:20.010 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[I 16:45:20.010 NotebookApp] http://localhost:8888/?token=5ce8c654d1596a45b367397f876d05
07f2a48db1b6d627e8
[I 16:45:20.010 NotebookApp] or http://127.0.0.1:8888/?token=5ce8c654d1596a45b367397f87
6d0507f2a48db1b6d627e8
[I 16:45:20.010 NotebookApp] Use Control-C to stop this server and shut down all kernels
(twice to skip confirmation).
[C 16:45:20.433 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/eyrc/.local/share/jupyter/runtime/nbserver-19348-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=5ce8c654d1596a45b367397f876d0507f2a48db1b6d627e8
or http://127.0.0.1:8888/?token=5ce8c654d1596a45b367397f876d0507f2a48db1b6d627e8

```

Figure 9: Output of jupyter notebook

- The landing page of the Jupyter Notebook, the **dashboard** shows the **IPython Notebooks (.ipynb)** currently available in the directory from where the notebook server was started.
- In the top-right, under the **New** dropdown, you will be able to see your newly created environment as shown in Figure 10.

Welcome to NB theme!

Rulebook

Task 0

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1

Task 2

Task 3

Task 4

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

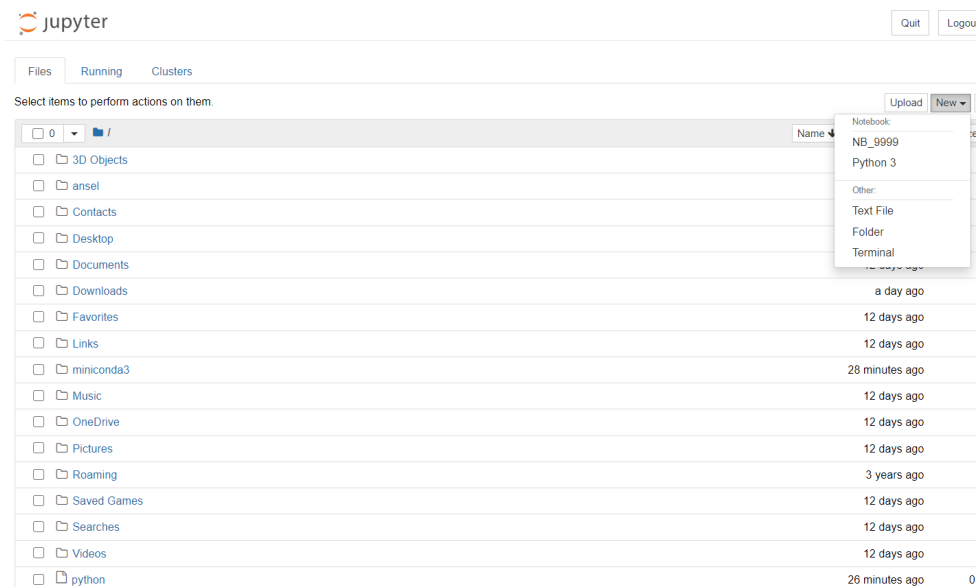


Figure 10: Jupyter Notebook able to detect the new environment

- Launch the **JupyterLab** using the below command with your conda environment activated.

```
jupyter-lab
```



OR

```
jupyter lab
```



- JupyterLab will open automatically in your browser just like the **Jupyter Notebook**. This will start the Notebook server and open a default web browser to the URL of the web application, which will be: **http://127.0.0.1:8888** OR **http://localhost:8888** as below in Figure 11a and 11b.

Welcome to NB theme!

Rulebook >

Task 0 >

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 >

Task 2 >

Task 3 >

Task 4 >

Task 5 >

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

```

eyrc@erts: ~/Downloads
File Edit View Search Terminal Help
(NB_9999) eyrc@erts:~/Downloads$ jupyter lab
[I 16:46:54.747 LabApp] JupyterLab extension loaded from /home/eyrc/miniconda3/envs/NB_9
999/lib/python3.8/site-packages/jupyterlab
[I 16:46:54.747 LabApp] JupyterLab application directory is /home/eyrc/miniconda3/envs/N
B_9999/share/jupyter/lab
[I 16:46:54.749 LabApp] Serving notebooks from local directory: /home/eyrc/Downloads
[I 16:46:54.749 LabApp] Jupyter Notebook 6.1.4 is running at:
[I 16:46:54.749 LabApp] http://localhost:8888/?token=975acf8075462d099fd0946cee7e8c88896
fe7ce33546bff
[I 16:46:54.749 LabApp] or http://127.0.0.1:8888/?token=975acf8075462d099fd0946cee7e8c8
8896fe7ce33546bff
[I 16:46:54.749 LabApp] Use Control-C to stop this server and shut down all kernels (twi
ce to skip confirmation).
[C 16:46:54.796 LabApp]

To access the notebook, open this file in a browser:
file:///home/eyrc/.local/share/jupyter/runtime/nbserver-19541-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=975acf8075462d099fd0946cee7e8c88896fe7ce33546bff
or http://127.0.0.1:8888/?token=975acf8075462d099fd0946cee7e8c88896fe7ce33546bff
[W 16:47:00.078 LabApp] Could not determine jupyterlab build status without nodejs

```

Figure 11a: Output of jupyter lab

The **warning message**: *Could not determine jupyterlab build status without nodejs* can be ignored.

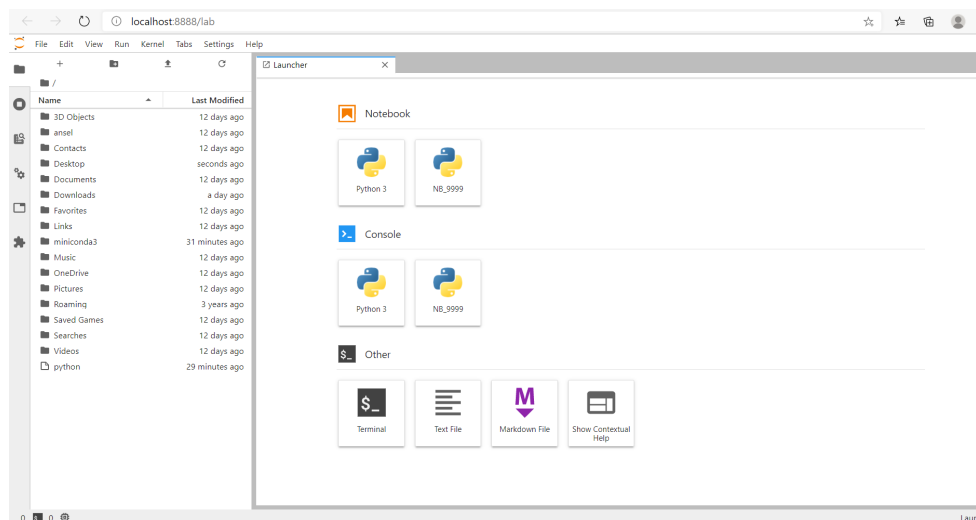


Figure 11b: Jupyter Lab able to detect the new environment

Welcome to NB theme!

Rulebook ›

Task 0 ~

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

**Note:** Get acquainted yourself with the UI of Jupyter Notebook dashboard, how to create markdow and code cells and run them, keyboard shortcuts, etc. from the [official documentation](#).

**Voila!** Your new Virtual Environment is setup successfully without any errors!

### 3. Coppeliasim 4.0.0

- In this section, we will install Coppeliasim Robotics Simulator Platform.
- The robot simulator Coppeliasim is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a ROS or BlueZero node, a remote API client, or a custom solution. This makes Coppeliasim very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave.
- Coppeliasim is used for fast algorithm development, factory automation simulations, fast prototyping and verification, robotics related education, remote monitoring, safety double-checking, as digital twin, and much more. You can find a feature overview [here](#).
- It can be used as a stand-alone application or can easily be embedded into a main client application: its small footprint and elaborate API makes Coppeliasim an ideal candidate to embed into higher-level applications.

#### A. Installation Steps

- Download **Coppeliasim Edu 4.0.0** for **Ubuntu 18.04 64-bit OS** from [here](#) (file size - 152MB). It will download as **.tar.xz** (compressed zip) file.

**Note:** To download Coppeliasim for **Ubuntu 16.04 64-bit OS**, click [here](#) (file size - 144MB).

- Open Terminal and navigate to the directory where this file was downloaded. Run the followir command:

```
tar -xf Coppeliasim_Edu_V4_0_0_Ubuntu18_04.tar.xz
```



This command will decompress and extract the Coppeliasim software to the folder named **Coppeliasim\_Edu\_V4\_0\_0\_Ubuntu18\_04** in the same directory.

- Now type the below commands in sequence to launch Coppeliasim.

Welcome to NB theme!

Rulebook

Task 0

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1

Task 2

Task 3

Task 4

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

```
cd CoppeliaSim_Edu_V4_0_0_Ubuntu18_04
./coppeliaSim.sh
```

- You will see the output as shown in Figure 12a and 12b. CoppeliaSim will open with the default scene loaded.

```
eyrc@erts: ~/Downloads/CoppeliaSim_Edu_V4_0_0_Ubuntu18_04
File Edit View Search Terminal Help
eyrc@erts:~/Downloads$ tar -xvf CoppeliaSim_Edu_V4_0_0_Ubuntu18_04.tar.xz
eyrc@erts:~/Downloads$ cd CoppeliaSim_Edu_V4_0_0_Ubuntu18_04
eyrc@erts:~/Downloads/CoppeliaSim_Edu_V4_0_0_Ubuntu18_04$ ./coppeliaSim.sh
Loading the CoppeliaSim library...
Done!
Launching CoppeliaSim...
lib: 1
lic: 1

CoppeliaSim Edu V4.0.0. (rev. 4)
Using the default Lua library.
```

Figure 12a: Extract CoppeliaSim and launch it

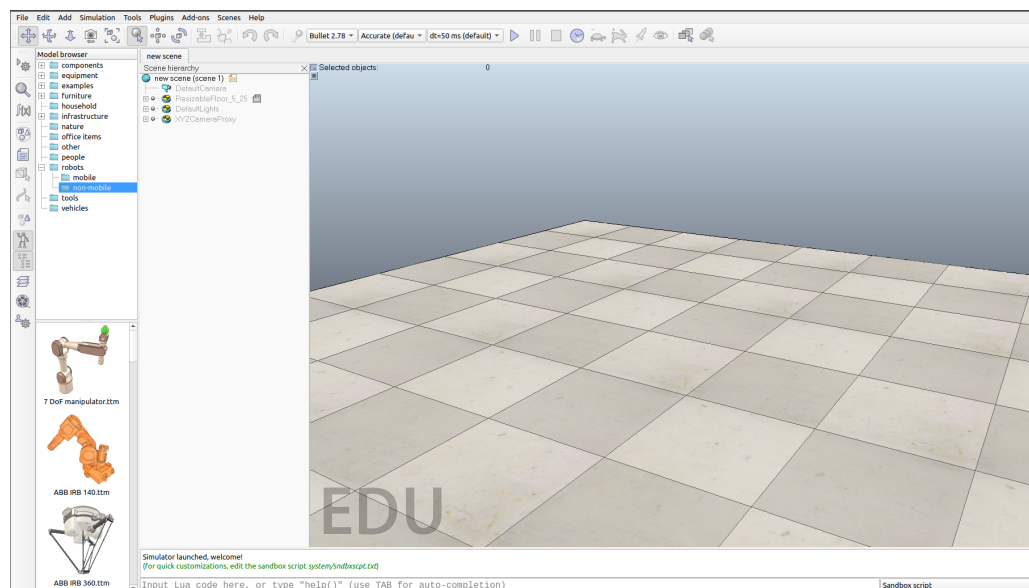


Figure 12b: CoppeliaSim software launch

Welcome to NB theme!

Rulebook ›

Task 0 ~

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

## 4. Visual Studio Code (*optional but highly recommended*)

- The below steps will increase your programming efficiency.
- Programmers spend a lot of time looking at documentation, re-factoring code, debugging issues, unit testing and styling code.
- What if you get a helping hand at this so that your code looks beautiful and you could debug issues faster? Read further to know more. Don't worry!
- Since using tooling may be a new thing to most of you it may look difficult but none of this is difficult. It's actually rather very very easy and you will become a better programmer the earlier you master these tools.
  - Download a Text Editor. We recommend **Visual Studio Code** (*file size - 60.9MB*).
  - If you write a code in Jupyter Notebook and wish to reuse it written, you will have to move the code in a **.py** file and then just use the file in **VS Code**.
  - When writing a Python program with IDLE you miss out on a lot of things, like auto-complete suggestions, auto-indentation, inline documentation for module/function, aesthetics, etc.
  - For efficacy reasons, we will use a good text editor like **VS Code**.
  - Set up **VS Code** for **Python**.
    - Install the Python Extension in VS Code: [link](#). This will setup all the tools necessary for easy editing, running and testing Python programs.
    - Select a Python Interpreter: [link](#). This will tell the Python IntelliSense which interpreter to use while you write, run and test your Python programs. It will also detect the virtual environments created by Miniconda and let you choose and switch between them. So, you will select the environment **NB\_<Team-ID>** in this case.
  - Other than auto-completion, inline documentation, etc. **VS Code** also [lints your code](#), helps you [debug](#) your code very easily (so you can avoid print statements).

## 5. Testing the Setup

In order to verify and test the successful installation of above software and libraries, follow the steps provided in [Test\\_Setup](#) document.

## References

Welcome to NB theme!

Rulebook ›

Task 0 ⌵

(1a) Software Installation (Ubuntu)

(1b) Software Installation (Windows)

(2) Test Setup

(3) Learning Resources / Tutorials

(4) Solving Assignment

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ›

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

1. [How, What and Why you should use Anaconda Miniconda?](#)

2. [Anaconda vs Miniconda - StackOverflow answer](#)

3. [Conda Cheat Sheet](#)

4. [JupyterLab Documentation](#)

5. [Jupyter Notebook Documentation](#)

6. [Coppelia Robotics](#)

7. [CoppeliaSim User Manual](#)

8. [Getting Started with Python in VS Code](#)

---