

00_00_Environment_setup

August 11, 2019

1 00. Environment setup

1.1 00.1 Install python

I recommend using python3.6 or python3.7 in this tutorial. We will focus ONLY on python on Linux/UNIX. We will not consider other OS like Windows, Mac OS X.

to install python run:

```
sudo apt-get install python3
```

1.2 00.2 Virtual Environment via virtualenv

It is a good python practice to use separate virtual environment for every project. There is a great tutorial about Virtual Environments <https://docs.python-guide.org/dev/virtualenvs/>. I highly recommend to familiarize yourself with it as an introduction. The part we are interested in this tutorial is:

Lower level: virtualenv

.

virtualenvwrapper

1.2.1 00.2.1 Check your version of python

In shell:

```
python
```

```
Python 3.6.8 (default, Jan 14 2019, 11:02:34)
[GCC 8.0.1 20180414 (experimental) [trunk revision 259383]] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

```
exit()
```

In our case we have python3.6 which is OK. If the python version is not satisfying for you then find the desired binary file:

```
which python
```

```
/usr/bin/python2.7
```

```
ls /usr/bin/ | grep python
```

```
python
python2
python2.7
python3
python3.6
python3.6m
python3m
```

if you choose eg. python3.6 the binary file is /usr/bin/python3.6

1.2.2 00.2.2 Install virtualenv

Package: <https://pypi.org/project/virtualenv/>
Documentation: <https://virtualenv.pypa.io/en/latest/>
Run:

```
pip install virtualenv
```

1.2.3 00.2.3 Create new Virtual Environment

Command to create new Virtual Environment:

```
virtualenv -p <path_to_python_binary_file> <path_to_the_directory_to_set_up_virtual_environment>
```

In our case it will be:

```
virtualenv -p /usr/bin/python3.6 ~/venvs/dstip_venv
```

A new Virtual Environment has been created in ~/venvs/dstip_venv. To activate it run:

```
source ~/venvs/dstip_venv/bin/activate
```

In the shell you should see information about Virtual Environment: The default python is now:

```
which python
```

```
/home/lcs123/venvs/dstip_venv/bin/python
```

1.2.4 00.2.4 Clone the repository and install requirements

```
cd some/directory/you/want/to/work...
git clone git@gitlab.com:lukasz_cesarski/dstip.git
cd dstip
ls
```

00_Environment_setup 02_Jupyter_Notebook 04_Numpy 06_Seaborn README.md
01_Ipython 03_Pandas 05_Matplotlib 07_scikit_learn

Now you need to care about having the Virtual Environment activated! Otherwise installed packages will not be working with your Virtual Environment!

```
pip install -r requirements.txt
```

Other required packages Linux (converted jupyter notebook to formats other than HTML, eg. PDF)

```
sudo apt-get install pandoc
sudo apt-get install texlive-xetex
```

Last thing that needs to be done is to slightly modify Virtual Environment activation file with our project root directory. This is VERY convenient. Assuming your current working directory is dstip you need to run:

```
echo "export PYTHONPATH=`pwd`" >> <path_to_the_venv_activation_file>
```

In our case it will be:

```
echo "export PYTHONPATH=`pwd`" >> ~/venvs/dstip_venv/bin/activate
```

1.3 00.3 jupyter notebook server

1.3.1 00.3.1 Run jupyter notebook server

Run:

```
jupyter notebook
```

You should get something like this:

```
[I 15:22:54.211 NotebookApp] Serving notebooks from local directory: /home/lcs123/DS/dstip
[I 15:22:54.211 NotebookApp] The Jupyter Notebook is running at:
[I 15:22:54.212 NotebookApp] http://localhost:8888/?token=05c9bff078455648dbd73882c837e4f44d7f56471047b354
[I 15:22:54.212 NotebookApp] or http://127.0.0.1:8888/?token=05c9bff078455648dbd73882c837e4f44d7f56471047b354
[I 15:22:54.212 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice)
[C 15:22:54.218 NotebookApp]
```

To access the notebook, open this file in a browser:

file:///home/lcs123/.local/share/jupyter/runtime/nbserver-9481-open.html

Or copy and paste one of these URLs:

http://localhost:8888/?token=05c9bff078455648dbd73882c837e4f44d7f56471047b354

or http://127.0.0.1:8888/?token=05c9bff078455648dbd73882c837e4f44d7f56471047b354

Follow the instructions and open jupyter notebook in your web browser Now you can go to the file *00_01_Test_installation.ipynb* and check if everything went well.

1.3.2 00.2.5 Stop jupyter notebook server

Console tab with living jupyter notebook should be safe and sound if you want to use it :)

If you wish to stop jupyter notebook server use CTR+C twice in the tab mentioned above.

If you wish to deactivate virtual environment use in console:

```
deactivate
```

1.4 00.4 Test 3rd party packages installation

Now please **restart Jupyter Notebook and Virtual Environment** and then run the cells below. You should be able to see two big markups that confirm everything went well.

```
[1]: # NOTE: wrapping `import` statement in `try, except` block is not a common
      ↪ practice
try:
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from IPython.display import display, Markdown, Latex
    display(Markdown("# 3rd Party Packages Installation Successful!"))
except ModuleNotFoundError:
    print("Installation of packages failed!")
```

2 3rd Party Packages Installation Successful!

```
[2]: import os
python_path = !echo $PYTHONPATH
if python_path[0] == os.path.dirname(os.getcwd()):
    display(Markdown("# PYTHONPATH modified correctly!"))
else:
    print("PYTHONPATH set improperly: PYTHONPATH='{ }'".format(python_path[0]))
```

3 PYTHONPATH modified correctly!

```
[ ]:
```