Conda commands cheat sheet 🡪 <https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf>

conda update conda 🡪 Udate conda

tells you whether conda is installed properly and also supplies you with some commands 🡪 conda

list all the conda envs 🡪 conda env list

list packages in conda env 🡪 conda list -n <my\_env>

Creating new virtual envs 🡪 conda create -n name\_ENV python=version\_req (-n stands for new)

-y 🡪 Is used to avoid yes 🡪y which comes in between

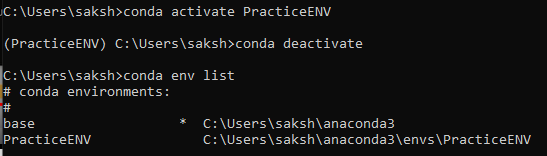


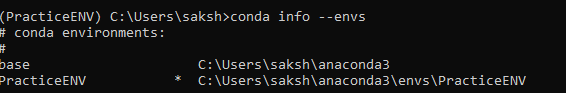
Installing multiple libraries at the sametime



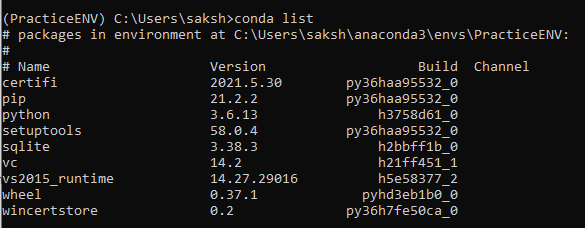
Activating ENVS 🡪 conda activate PracticeENV

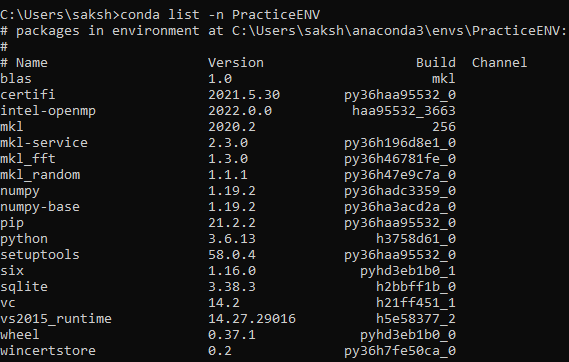
Deactivating ENVS 🡪 conda deactivate



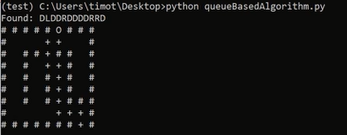
Getting info regarding the envs 🡪 

Getting list of packages along with their versions in your Env

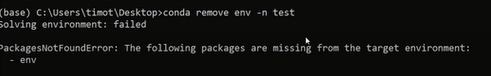


Getting list of packages along with their versions in outside your Env 

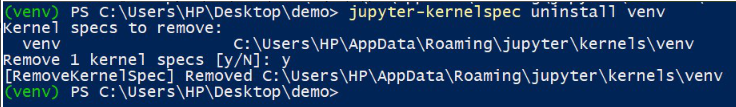
Running a particular programme via using a particular virtual env on the cmd



Deleting conda env



Making virtual env and using it with jupyter notebook

* Make the virtual env
* Activate that
* Most important thing ipykernel package should already be installed in that virtual env
* Activate that virtual environment
* 
* Running the following command will create a kernel that can be used to run jupyter notebook commands inside the virtual environment.
* python -m ipykernel install --user --name other-env --display-name "Python (other-env)"
* The --name value is used by Jupyter internally. These commands will overwrite any existing kernel with the same name. --display-name is what you see in the notebook menus
* Un installing kernel doesn’t means that you uninstall the virtual env
* 

Exporting virtual environment 🡪https://towardsdatascience.com/how-to-export-and-load-anaconda-environments-for-data-science-projects-77dc3b781369



FOR Creating a copy/Creating a similar env

1. You must change the env name at the very first line of .yaml file
2. 
3. The name of the environment will be the same as that of the one being changed by you in the .yaml file

You can use Anaconda to export a Python virtual environment to a YAML file you can then reuse in different projects.

You can also share the YAML file with your team, so everyone’s on the same track.