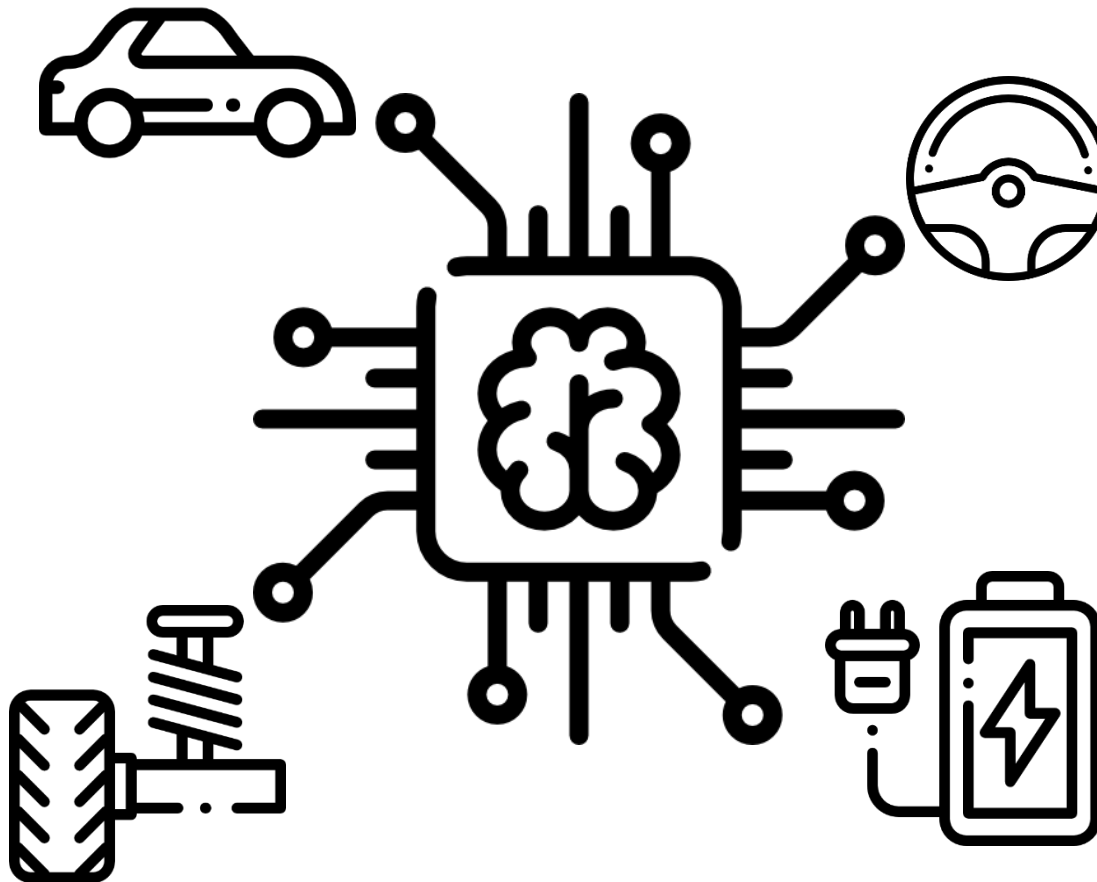


Artificial Intelligence in Automotive Technology

Prof. Dr.-Ing. Markus Lienkamp

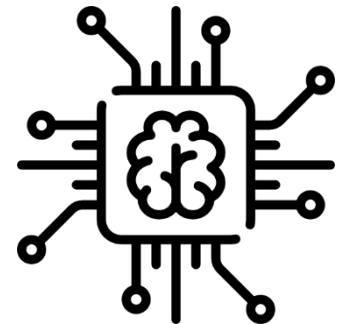
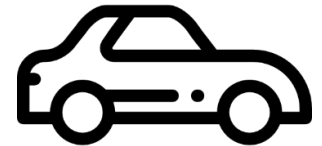


Artificial Intelligence in Automotive Technology

Technical Introduction
(Maximilian Geißlinger, M. Sc.)

Agenda

1. Part: Python
2. Part: Python Packages
3. Part: Introduction to Jupyter Notebook
4. Part: Setting up your system
5. Part: Using Jupyter Notebook
6. Part: Demo/Walkthrough



Python

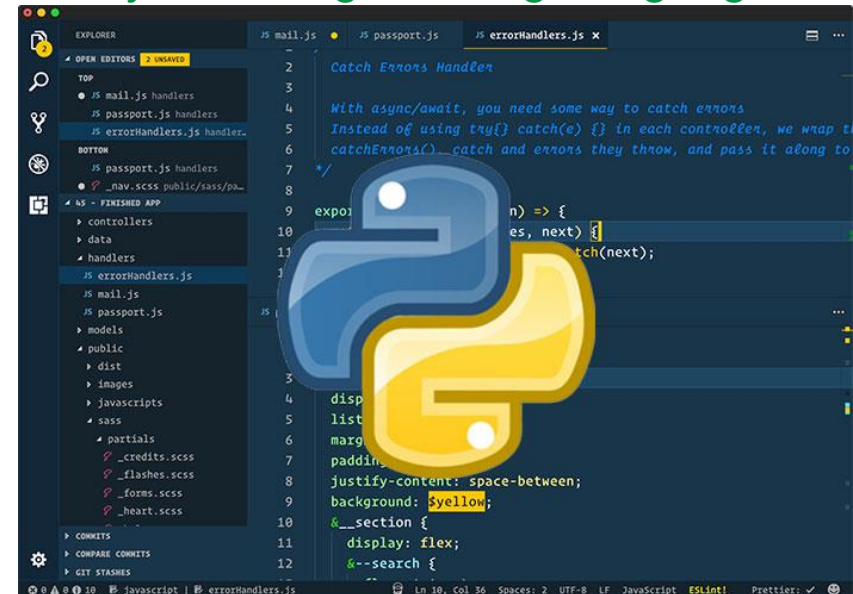
- Python is an interpreted, high-level, general-purpose programming language

NAAAA not this kind of python!



Source: <https://www.pcworld.com/article/3287981/best-python-courses.html>

Python Programming Language



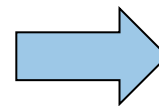
Source: <https://blog.vtutor.com/programming/python-programming-can-help-your-career/>

Python

- To get started working with Python, you will need to have access to a **Python interpreter**
- We will use **Python=3.7.6**

```
if config.mode != 'ev':  
    # split neg data into train and test data  
    x_train, x_test, y_train, y_test = train_test_split(  
        x, y, test_size=config.test_size,  
        shuffle=True, random_state=config.random)
```

Code



Interpreter

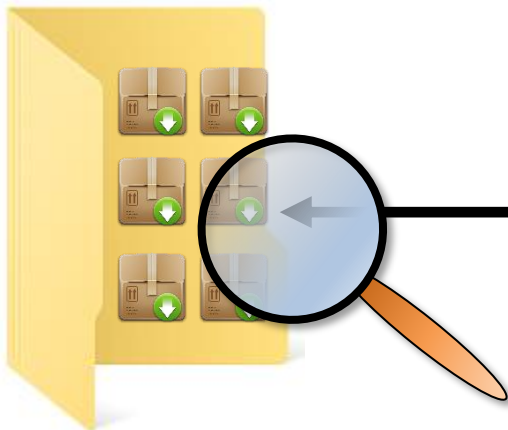
Python Packages

- Python script usually start with **import** commands → You do not program everything yourself, it is an **open source community!**



```
# import packages
import numpy as np
from joblib import load, dump
from sklearn.model_selection import train_test_split, ParameterSampler
from tensorflow.keras.models import load_model
```

Code with imports



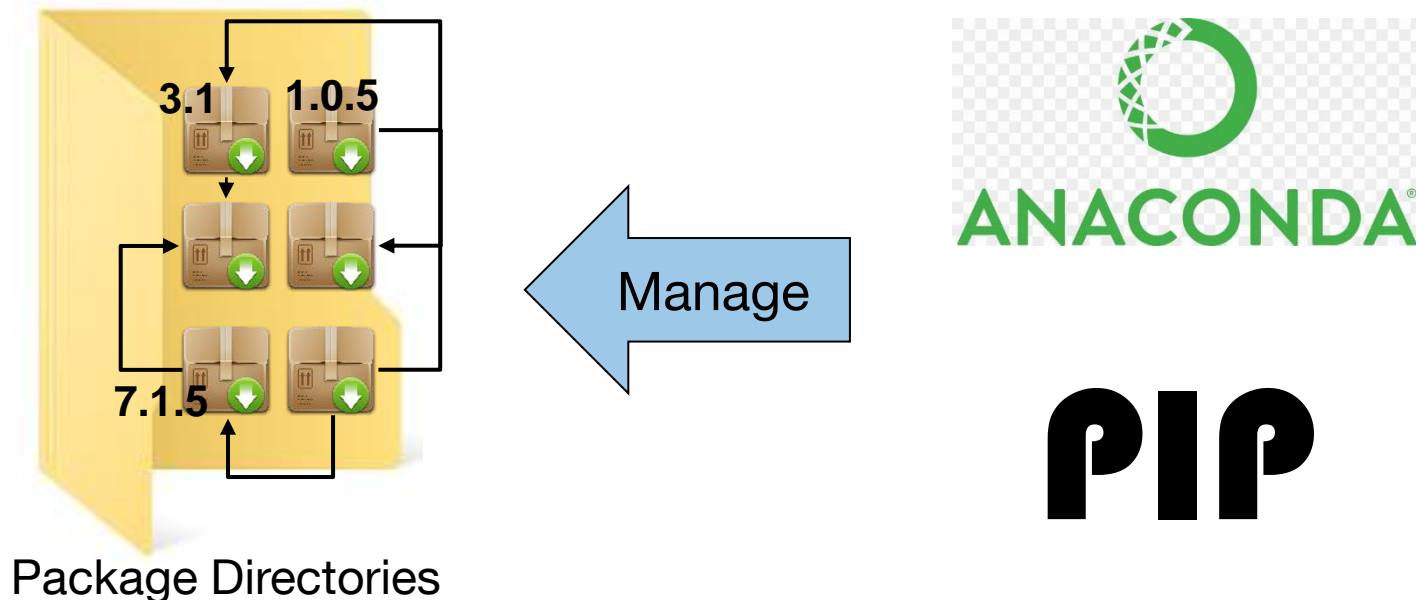
Package Directories



Interpreter

Python Packages

- Python packages experience constant change (multiple versions)
- Python packages may have interdependencies
- **Package managers** (e.g. pip, anaconda) handle installation, versioning, updates and mutual dependencies



Introduction to Jupyter Notebook

- Lightweight tool to develop and execute python code
- Creates “Notebook”-files (.ipynb)
 - Code
 - Documentation
 - Outputs (results, plots)
- Small to medium sized projects
- Documentation: <https://jupyter-notebook.readthedocs.io/en/stable/>
- **You will use Jupyter throughout this course**



Setting up your system

- Install a python interpreter, pip and jupyter notebook on your system:
 - Python can be obtained from the **Python Software Foundation** website at python.org.
 - Some operating systems, notably Linux, provide a **package manager** that can be run to install Python.
 - On macOS, the best way to install Python 3 involves installing a package manager called **Homebrew**.
 - Python installation Tutorial: <https://realpython.com/installing-python/#windows>
 - This is how most developers do it. Nevertheless, you have take care of package installation yourself by using pip from the command line
- Use the anaconda navigator and our anaconda environment from moodle

Setting up your system – Anaconda Navigator

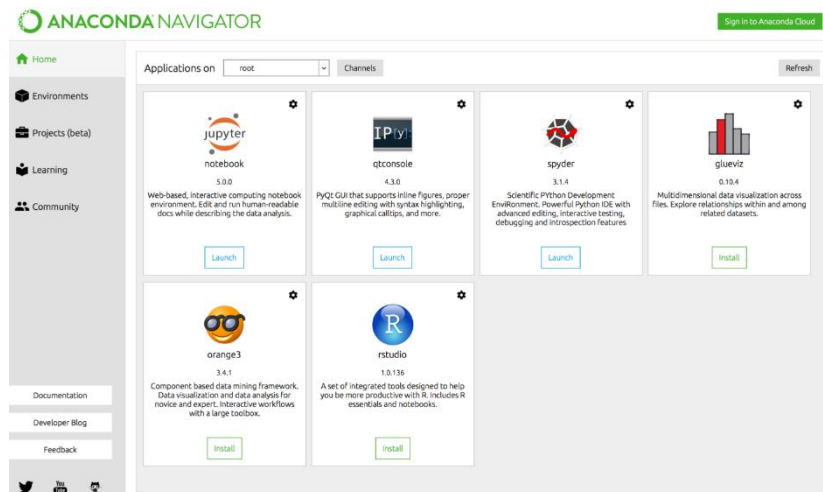
- **Anaconda Navigator:** Combines a python3 installation with a **GUI** to setup project environments and to install packages

NAAA not this kind of anaconda!



Source: <https://en.wikipedia.org/wiki/Anaconda>

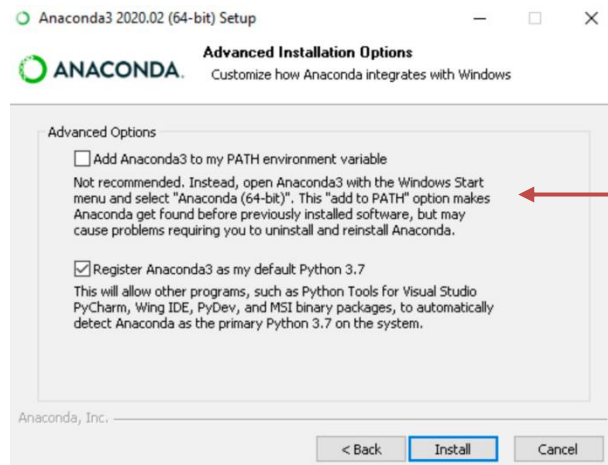
Package Manager Anaconda



Source: <https://medium.com/@kumarankita764/new-features-of-anaconda-5-3-5bdf9b4240>

Setting up your system – Anaconda Navigator

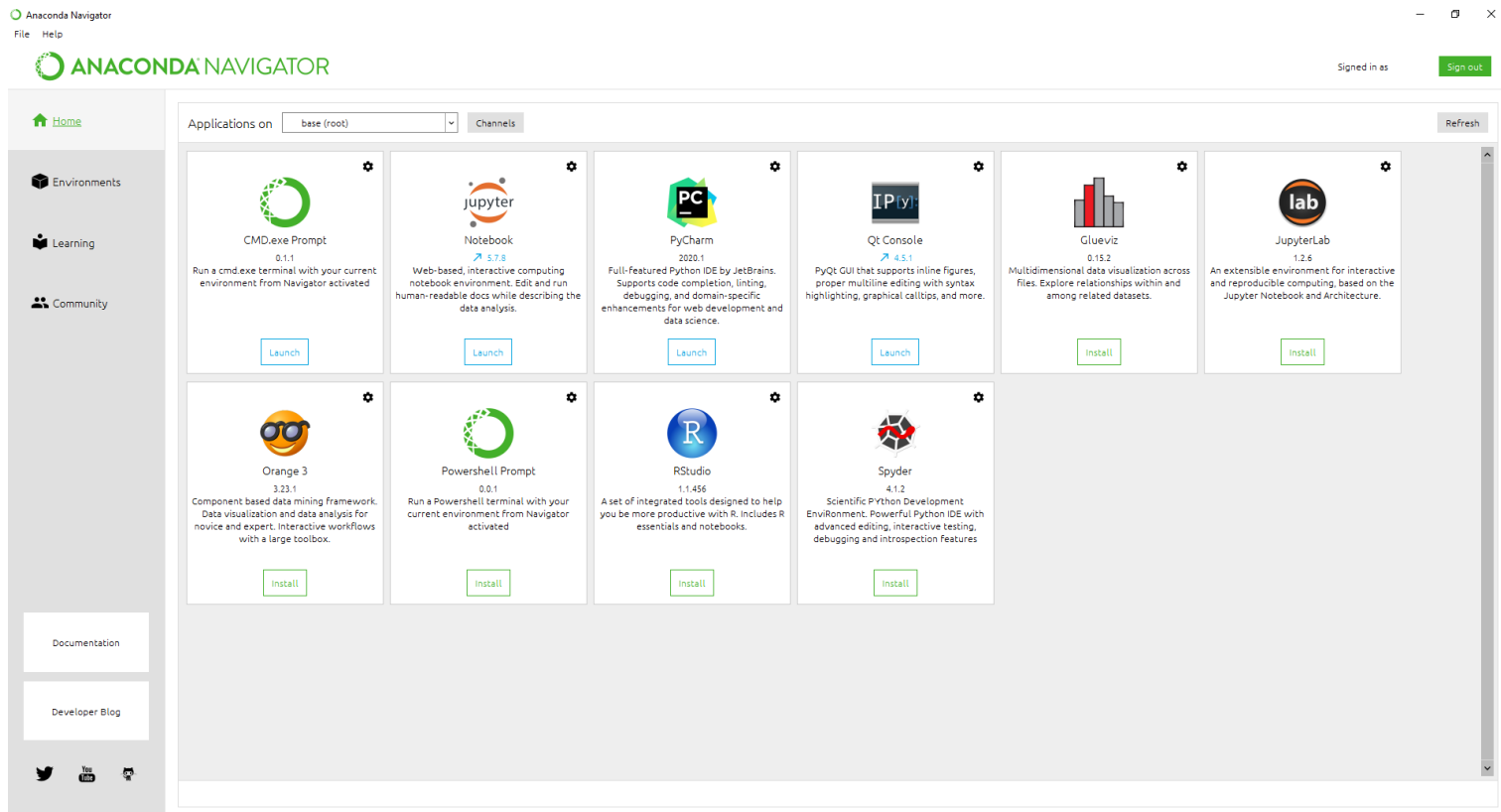
- Download the most recent Anaconda version available on the Anaconda homepage:
 - <https://www.anaconda.com/distribution/#download-section>
 - Select the Python 3.7 version
- Start the installation process
- Detailed information regarding the installation process can be found on <https://docs.anaconda.com/anaconda/install/>



Warning: When installing on Windows, do **not** add Anaconda to the PATH environment.

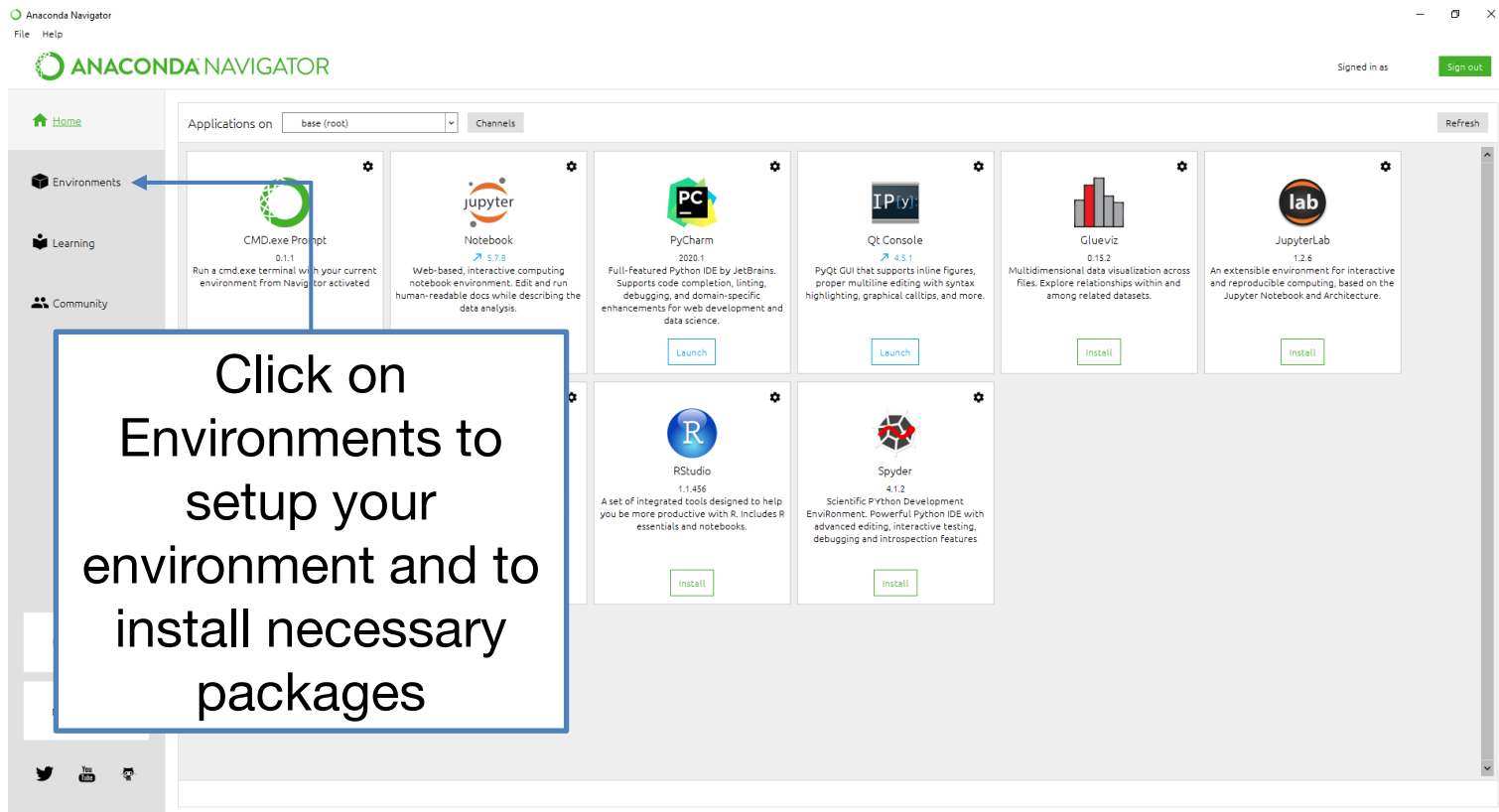
Setting up your system – Anaconda Navigator

■ Launch Anaconda Navigator



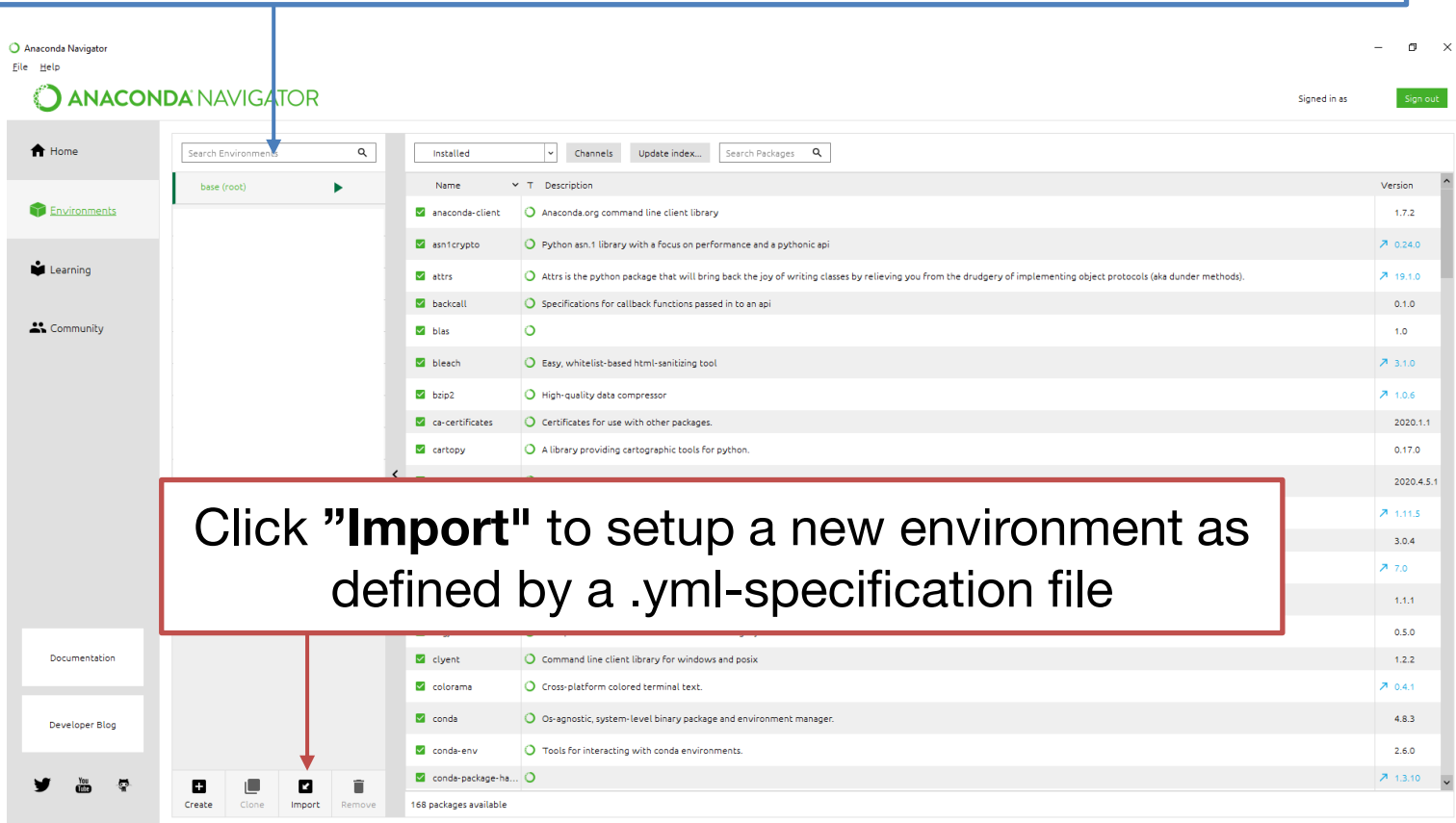
Setting up your system – Anaconda Navigator

- Navigate to the environments section



Setting up your system – Anaconda Navigator

The "base" environment is available after the installation of Anaconda containing commonly used python packages



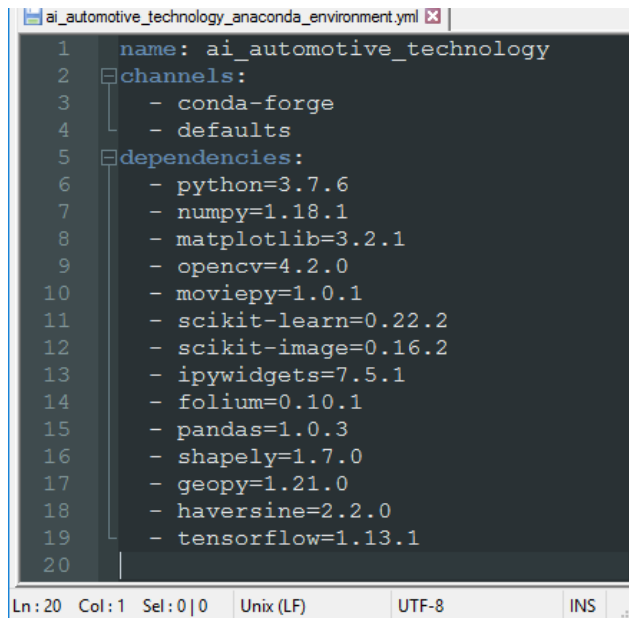
Click **"Import"** to setup a new environment as defined by a .yaml-specification file

Name	Description	Version
anaconda-client	Anaconda.org command line client library	1.7.2
asn1crypto	Python asn.1 library with a focus on performance and a pythonic api	0.24.0
attrs	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	19.1.0
backcall	Specifications for callback functions passed in to an api	0.1.0
blas		1.0
bleach	Easy, whitelist-based html-sanitizing tool	3.1.0
bzip2	High-quality data compressor	1.0.6
ca-certificates	Certificates for use with other packages.	2020.1.1
cartopy	A library providing cartographic tools for python.	0.17.0
		2020.4.5.1
		1.11.5
		3.0.4
		7.0
		1.1.1
		0.5.0
client	Command line client library for windows and posix	1.2.2
colorama	Cross-platform colored terminal text.	0.4.1
conda	Os-agnostic, system-level binary package and environment manager.	4.8.3
conda-env	Tools for interacting with conda environments.	2.6.0
conda-package-ha...		1.3.10

168 packages available

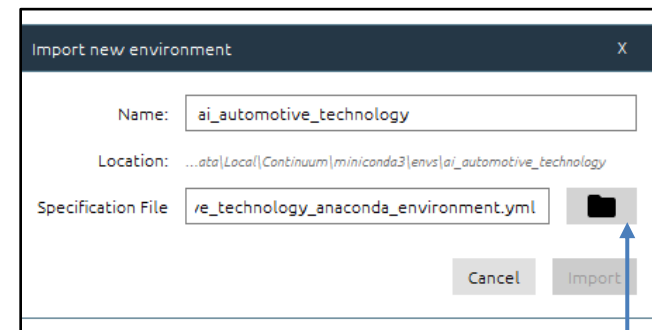
Setting up your system – Anaconda Navigator

- The necessary specification file can be downloaded from our moodle course and looks as follows:



```
1 name: ai_automotive_technology
2 channels:
3   - conda-forge
4   - defaults
5 dependencies:
6   - python=3.7.6
7   - numpy=1.18.1
8   - matplotlib=3.2.1
9   - opencv=4.2.0
10  - moviepy=1.0.1
11  - scikit-learn=0.22.2
12  - scikit-image=0.16.2
13  - ipywidgets=7.5.1
14  - folium=0.10.1
15  - pandas=1.0.3
16  - shapely=1.7.0
17  - geopy=1.21.0
18  - haversine=2.2.0
19  - tensorflow=1.13.1
20
```

Required dependencies are listed
with their package versions

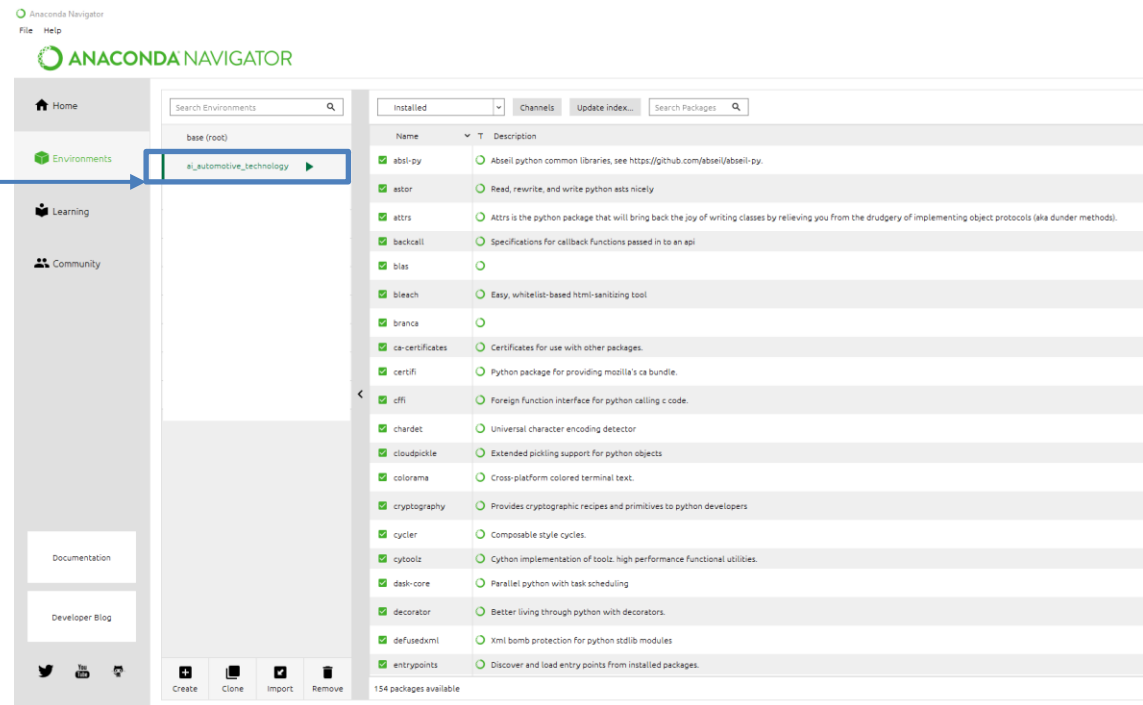


Select the downloaded specification
file named
ai_automotive_technology_anaconda_environment.yml

Setting up your system – Anaconda Navigator

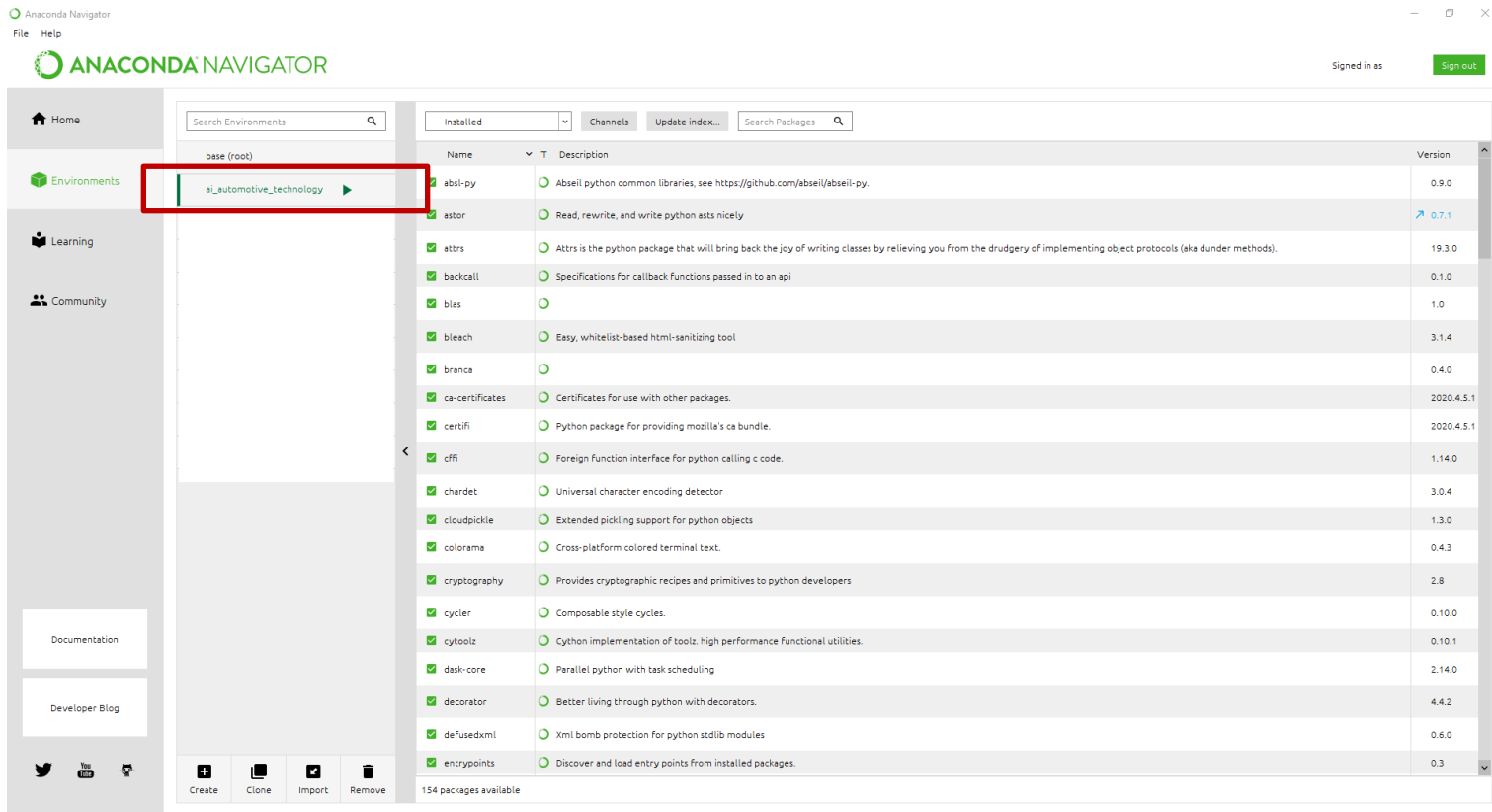
- The installation process **takes some time** because several packages need to be downloaded (ensure a quick internet connection) and installed

There is a new button available after the setup of a new environment



Setting up your system – Anaconda Navigator

- Click on the button “**ai_automotive_technology**” (name of the environment) to **activate** this course’s environment



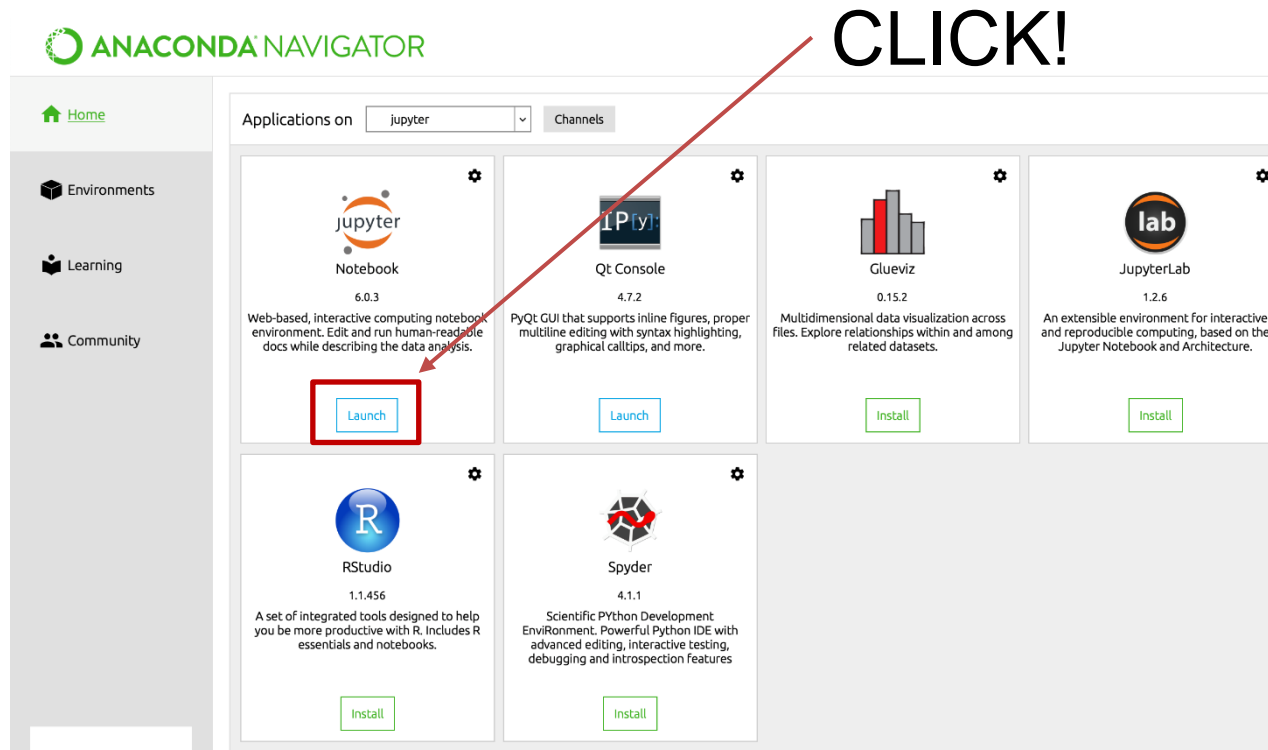
The screenshot shows the Anaconda Navigator application window. On the left sidebar, the 'Environments' tab is selected. In the 'Environments' list, the environment named 'ai_automotive_technology' is highlighted with a red rectangular box. To the right of this environment name is a green play button icon. The main panel displays a list of installed packages for this environment. The table has columns for 'Name', 'Description', and 'Version'.

Name	Description	Version
absl-py	Abseil python common libraries, see https://github.com/abseil/abseil-py .	0.9.0
astor	Read, rewrite, and write python asts nicely	0.7.1
attrs	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	19.3.0
backcall	Specifications for callback functions passed in to an api	0.1.0
bias		1.0
bleach	Easy, whitelist-based html-sanitizing tool	3.1.4
branca		0.4.0
ca-certificates	Certificates for use with other packages.	2020.4.5.1
certifi	Python package for providing mozilla's ca bundle.	2020.4.5.1
cffi	Foreign function interface for python calling c code.	1.14.0
chardet	Universal character encoding detector	3.0.4
cloudpickle	Extended pickling support for python objects	1.3.0
colorama	Cross-platform colored terminal text.	0.4.3
cryptography	Provides cryptographic recipes and primitives to python developers	2.8
cycler	Composable style cycles.	0.10.0
cytoolz	Cython implementation of toolz: high performance functional utilities.	0.10.1
dask-core	Parallel python with task scheduling	2.14.0
decorator	Better living through python with decorators.	4.4.2
defusedxml	Xml bomb protection for python stdlib modules	0.6.0
entrypoints	Discover and load entry points from installed packages.	0.3

At the bottom of the package list, it says '154 packages available'. The bottom of the interface has a toolbar with buttons: 'Create', 'Clone', 'Import', and 'Remove'.

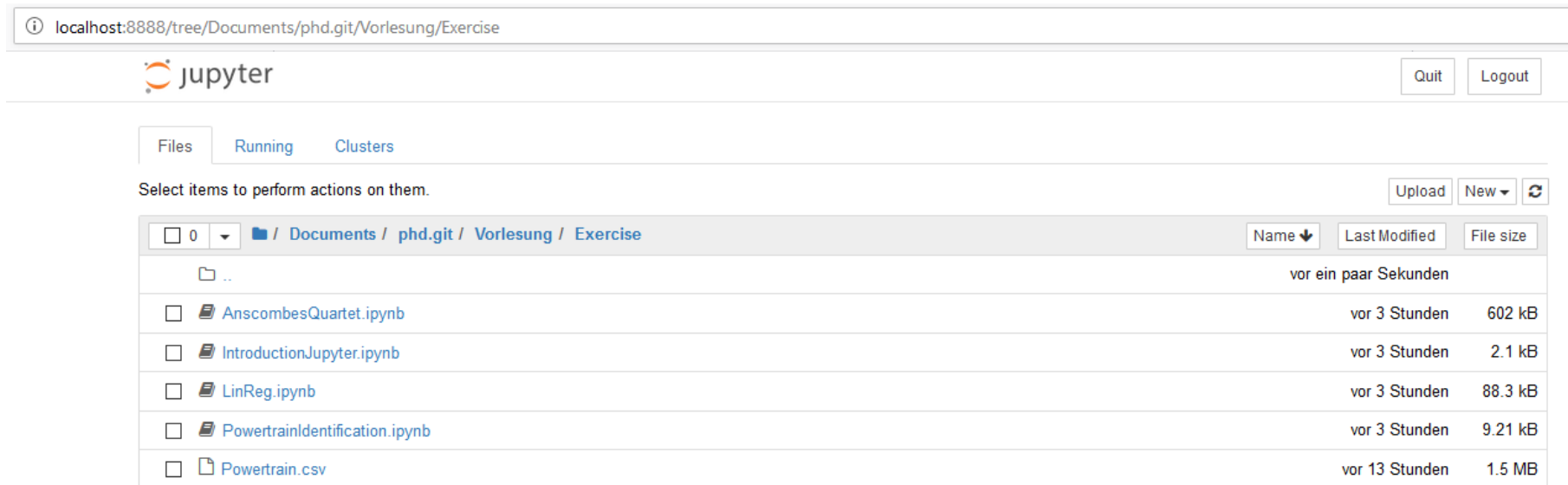
Using Jupyter Notebook – from Anaconda

- Launch the notebook by clicking on the **launch button** of the jupyter notebook field



Using Jupyter Notebook

- A jupyter notebook should launch in your web browser now and you are ready to navigate to your desired file from the practice sessions of the lecture (***.ipynb**)

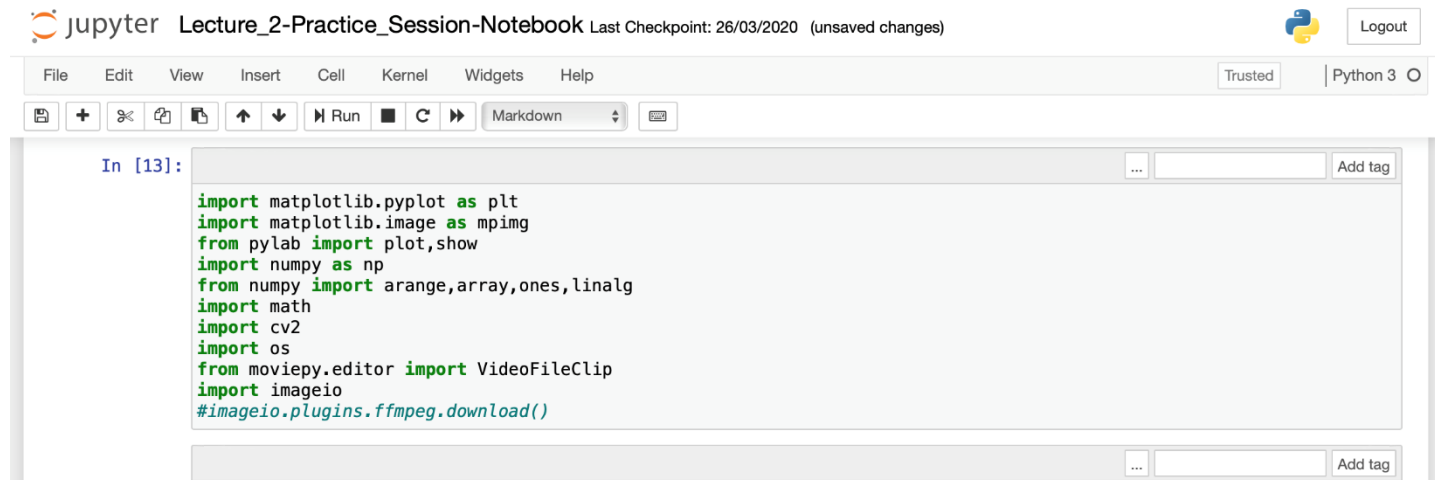


The screenshot shows the Jupyter Notebook web interface in a browser. The address bar displays the URL: `localhost:8888/tree/Documents/phd.git/Vorlesung/Exercise`. The Jupyter logo is visible in the top left, and 'Quit' and 'Logout' buttons are in the top right. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser view. The breadcrumb path is `/ Documents / phd.git / Vorlesung / Exercise`. A table lists the files in the directory:

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	..	vor ein paar Sekunden	
<input type="checkbox"/>	AnscombesQuartet.ipynb	vor 3 Stunden	602 kB
<input type="checkbox"/>	IntroductionJupyter.ipynb	vor 3 Stunden	2.1 kB
<input type="checkbox"/>	LinReg.ipynb	vor 3 Stunden	88.3 kB
<input type="checkbox"/>	PowertrainIdentification.ipynb	vor 3 Stunden	9.21 kB
<input type="checkbox"/>	Powertrain.csv	vor 13 Stunden	1.5 MB

Using Jupyter Notebook

- You can now start coding!
- You will find the jupyter notebooks for this course on moodle



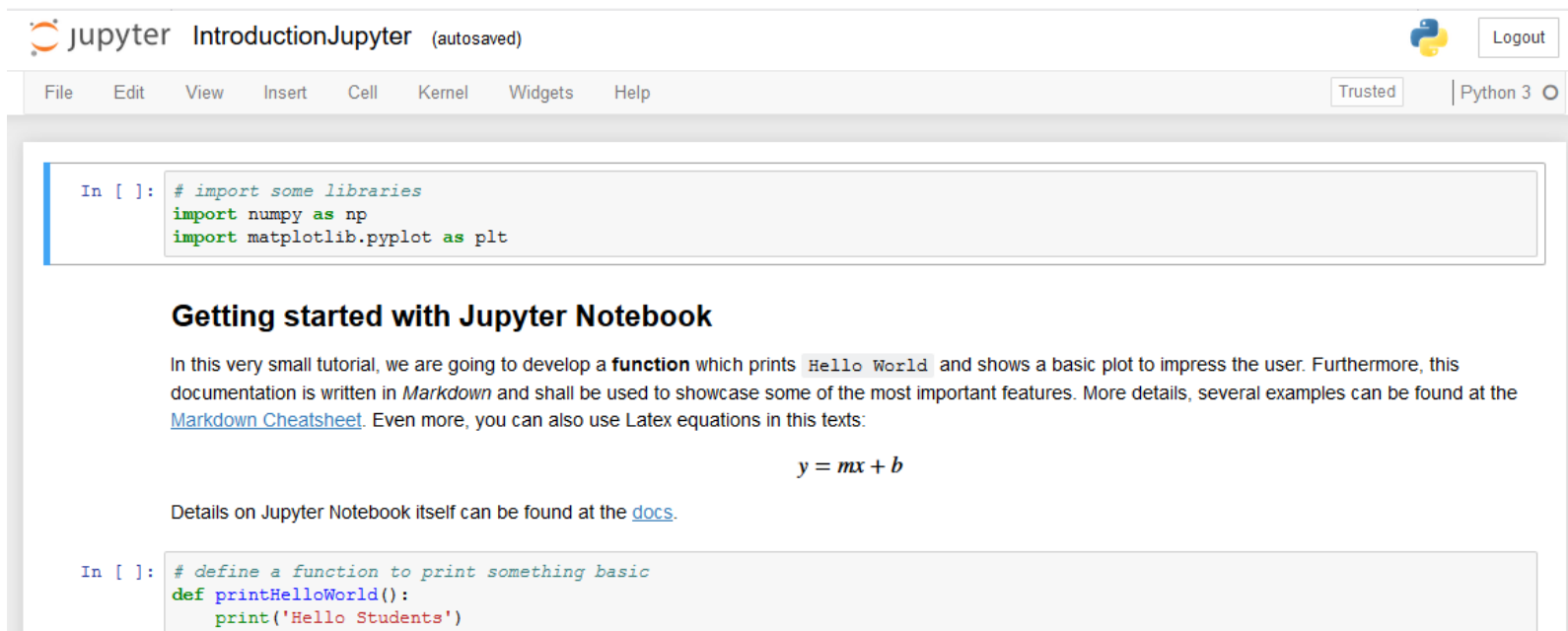
The screenshot displays the Jupyter Notebook web interface. At the top, the title bar reads "jupyter Lecture_2-Practice_Session-Notebook" followed by "Last Checkpoint: 26/03/2020 (unsaved changes)". On the right of the title bar are a Python logo and a "Logout" button. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Trusted" and "Python 3" buttons. Below the menu bar is a toolbar with icons for saving, adding a new cell, undo, redo, copy, paste, and running the cell. A dropdown menu is set to "Markdown". The main area shows a code cell labeled "In [13]:" containing the following Python code:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from pylab import plot, show
import numpy as np
from numpy import arange, array, ones, linalg
import math
import cv2
import os
from moviepy.editor import VideoFileClip
import imageio
#imageio.plugins.ffmpeg.download()
```

At the bottom right of the code cell, there are buttons for "Add tag" and a search bar.

Using Jupyter Notebook

- Open one of the notebooks by clicking on the name
- Run each cell separately: click it and press *Strg+Enter* (execution order matters!)
- Or run the whole notebook: *Kernel* → Restart Kernel and Run all Cells



The screenshot shows a Jupyter Notebook interface. At the top, there's a header bar with the Jupyter logo, the text "jupyter IntroductionJupyter (autosaved)", a Python logo, and a "Logout" button. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". On the right of the menu bar, there's a "Trusted" status indicator and a "Python 3" kernel selector. The main content area contains two cells. The first cell is a code cell with the following Python code:

```
In [ ]: # import some libraries
import numpy as np
import matplotlib.pyplot as plt
```

The second cell is a markdown cell with the following content:

Getting started with Jupyter Notebook

In this very small tutorial, we are going to develop a **function** which prints `Hello World` and shows a basic plot to impress the user. Furthermore, this documentation is written in *Markdown* and shall be used to showcase some of the most important features. More details, several examples can be found at the [Markdown Cheatsheet](#). Even more, you can also use Latex equations in this texts:

$$y = mx + b$$

Details on Jupyter Notebook itself can be found at the [docs](#).

The third cell is a code cell with the following Python code:

```
In [ ]: # define a function to print something basic
def printHelloWorld():
    print('Hello Students')
```

DEMO / WALKTRHOUGH