



13 July 2021

Jupyter Notebook



Carlo Corradini § Massimiliano Fronza



Jupyter Notebook

Project Jupyter



What ?

Project Jupyter is a **broad collaboration that develops open-source tools for interactive and exploratory computing.**

The tools include over **100 computer languages** (with a **focus on Python**) and an **ecosystem of extensions** contributed by a large community.

*"Project Jupyter exists to develop **open-source** software, **open-standards**, and **services** for **interactive computing across dozens of programming languages**."*



Project Jupyter is a *non-profit, open-source* project, born out of the **IPython Project** in 2014 as it evolved to **support interactive data science and scientific computing** across all programming languages.

Jupyter is developed in the open on *Github*, through the consensus of the **Jupyter community**.



Who's using Jupyter ?

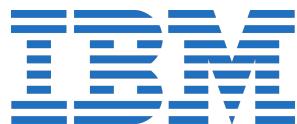


Microsoft



Bloomberg

 **The Data Incubator**
A Pragmatic Institute Company



SOUNDCLOUD

Symphony
AYASDI

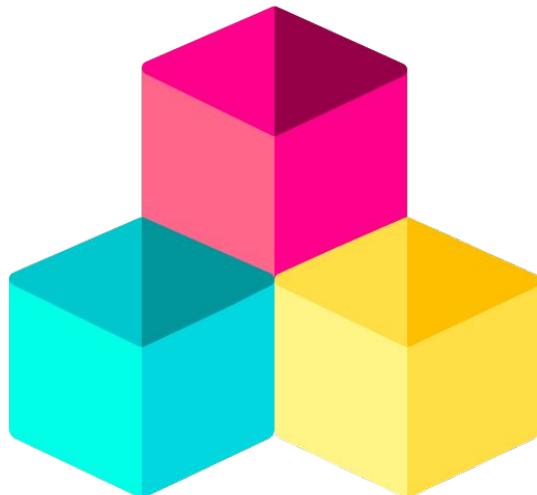
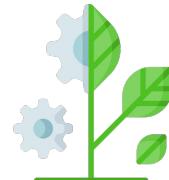
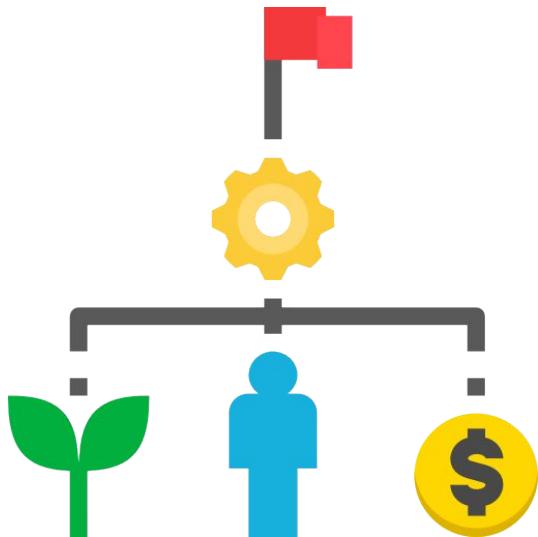


QuantStack
Scientific Computing

O'REILLY®

 **UNIVERSITY OF TRENTO**

Jupyter Ecosystem . . .



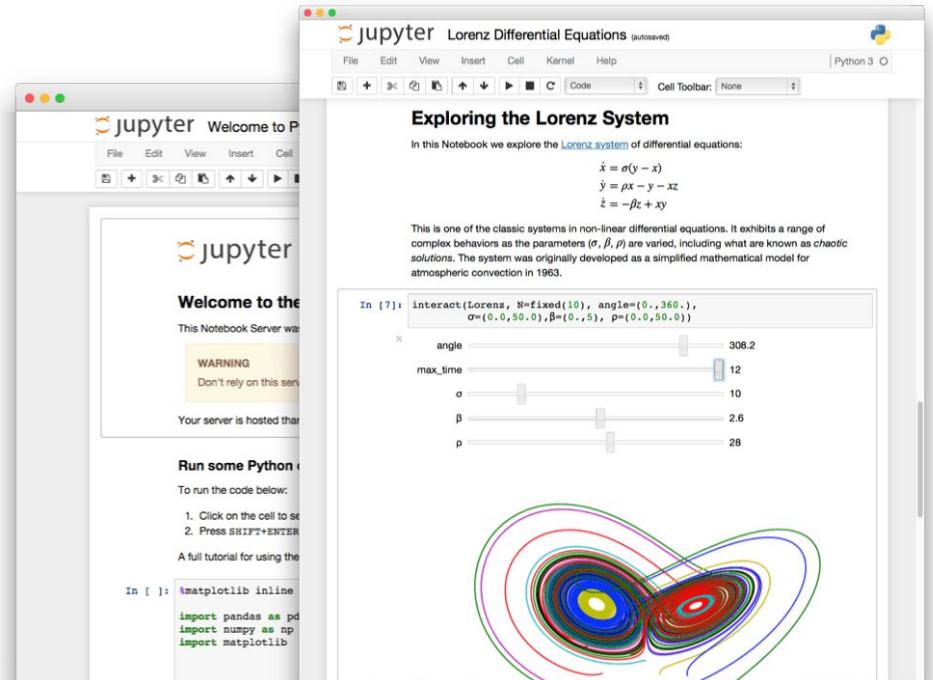


Project Jupyter



Products

1. Jupyter Notebook
2. JupyterLab
3. JupyterHub
4. Jupyter Voilà



UNIVERSITY
OF TRENTO



Project Jupyter



Products

1. **Jupyter Notebook**
2. **JupyterLab**
3. **JupyterHub**
4. **Jupyter Voilà**



Jupyter Notebook is an **open-source web application** that allows to **create and share documents that contain live code, equations, visualizations and narrative text.**

Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.



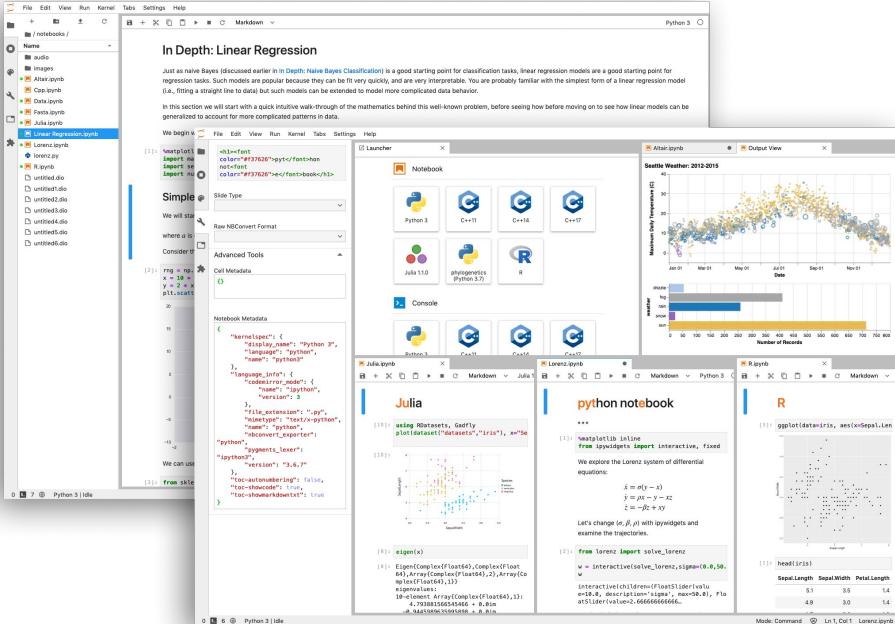


Project Jupyter



Products

1. Jupyter Notebook
2. JupyterLab
3. JupyterHub
4. Jupyter Voilà





Products

1. Jupyter Notebook
2. **JupyterLab**
3. JupyterHub
4. Jupyter Voilà



Jupyter's Next-Generation Notebook Interface.

JupyterLab is a **web-based interactive development environment for Jupyter notebooks, code, and data**. *JupyterLab* is **flexible**: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. *JupyterLab* is **extensible** and **modular**: write **plugins** that add new components and integrate with existing ones.



Products

1. Jupyter Notebook
2. JupyterLab
- 3. JupyterHub**
4. Jupyter Voilà





Products

1. Jupyter Notebook
2. JupyterLab
3. JupyterHub
4. Jupyter Voilà

A **multi-user version of the notebook** designed for companies, classrooms and research labs.



Pluggable Authentication

Manage users and authentication with PAM, OAuth or integrate with custom directory service system.



Centralized Deployment

Deploy the Jupyter Notebook to thousands of users in an organization on centralized infrastructure on- or off-site.



Container Friendly

Use Docker and Kubernetes to scale the deployment, isolate user processes, and simplify software installation.



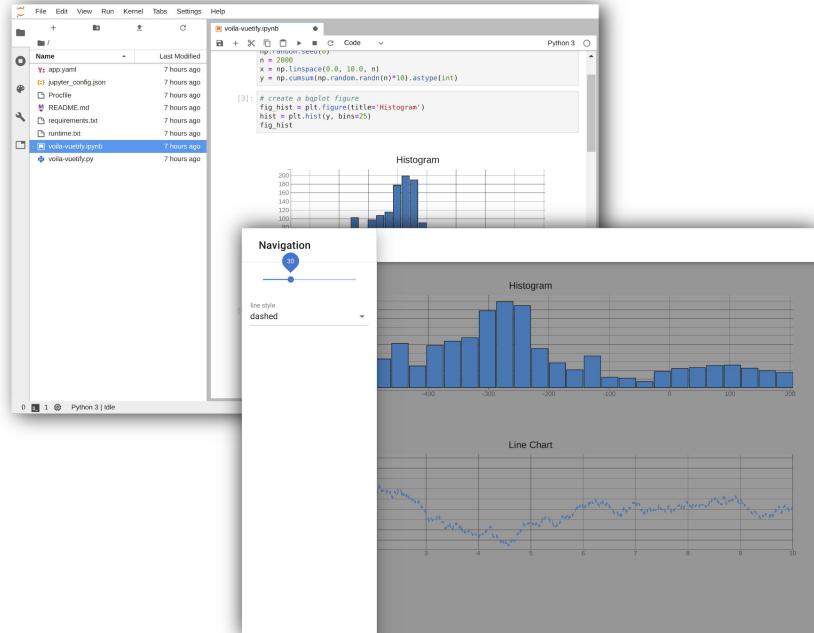
Code Meets Data

Deploy the Notebook next to data to provide unified software management and data access within an organization.



Products

1. Jupyter Notebook
2. JupyterLab
3. JupyterHub
4. Jupyter Voilà





Products

1. Jupyter Notebook
2. JupyterLab
3. JupyterHub
4. Jupyter Voilà



Voilà helps to communicate insights, by **transforming a Jupyter Notebook into a stand-alone web application** that can be *shared*.

It gives control over what readers experience in a **secure and customizable interactive dashboard**.



Jupyter Notebook

Jupyter Notebook



UNIVERSITY
OF TRENTO



Jupyter Notebook



Useful resource!

Teaching and Learning with Jupyter

<https://jupyter4edu.github.io/jupyter-edu-book>

The book explores the *Jupyter* ecosystem with a focus on *Jupyter Notebook*.

The handbook is for any educator teaching a topic that includes data analysis or computation in order to support learning.
It is not just for educators teaching courses in engineering or science, but also data journalism, business and quantitative economics, data-based decision sciences and policy, quantitative health sciences, and digital humanities.





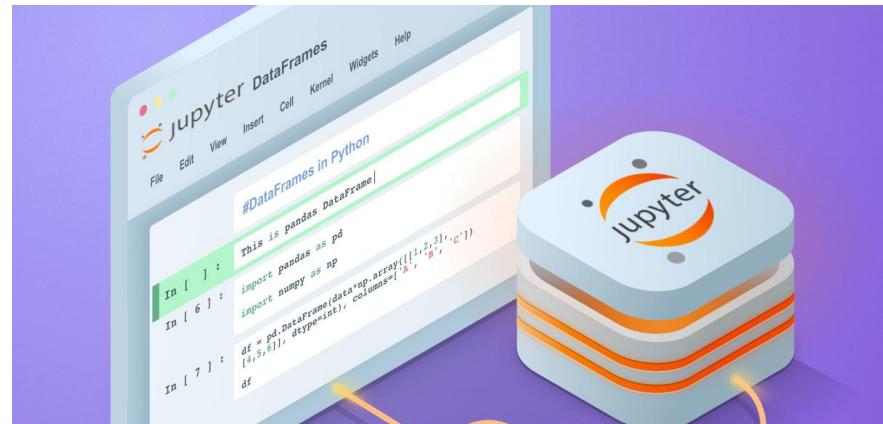
Jupyter Notebook



What ?

Jupyter Notebook is an **open-source web application** that allows to **create** and **share documents** that contain **live code, equations, visualizations** and **narrative text**.

Uses include: *data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.*



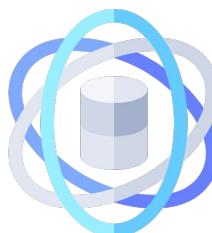
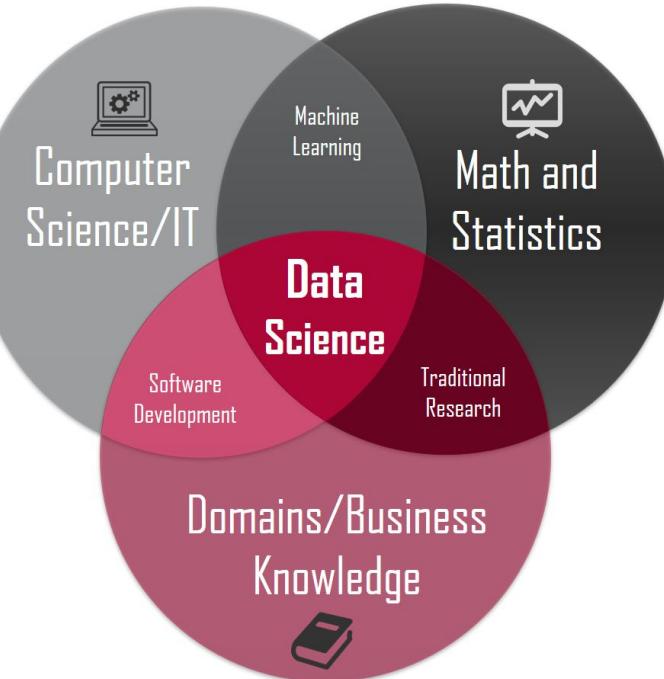
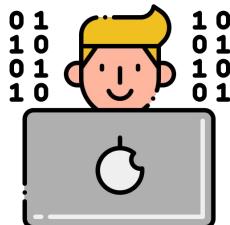
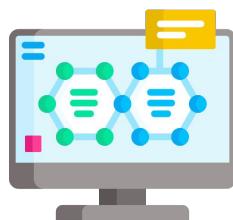
UNIVERSITY
OF TRENTO



Jupyter Notebook



Why ?





Language Support

The Jupyter system **supports over 100 programming languages** (called "*kernels*"). Out of the box, Jupyter supports only Python, but additional kernels may be installed.

Jupyter kernels allow **flexibility, dynamicity, and context switching**, bringing a **richer environment**.

For example, instructors may use Python to teach programming, while switching to R to teach statistics, and then perhaps Scala to teach big-data processing.

For an up-to-date list:

<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>





Jupyter Notebook

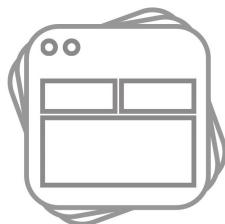


Characteristics



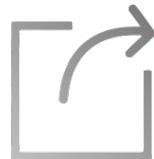
Language of choice

Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.



Interactive output

The code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.



Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.



Big data integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, TensorFlow.



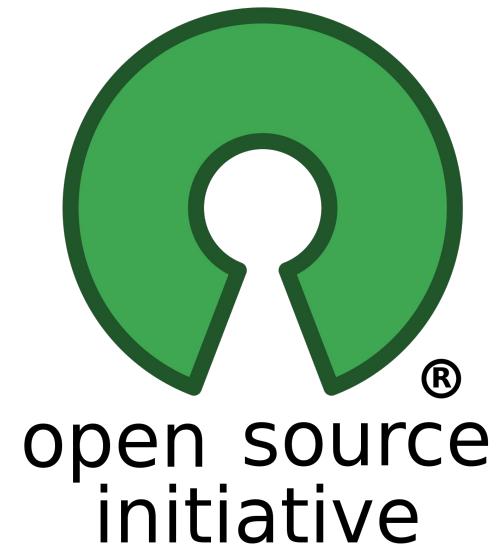
UNIVERSITY
OF TRENTO

Open Standards for Interactive Computing

The *Jupyter Notebook* is based on a **set of open standards** for interactive computing.

These open standards can be leveraged by third party developers to **build customized applications with embedded interactive computing.**

- end-to-end workflow
- data intensive & data driven analytics
- interactive data analytics over HPC
- infrastructural implications and data science environment





Jupyter Notebook



Open Standards for Interactive Computing



Notebook Document Format



Interactive Computing Protocol



The Kernel



Notebook
Document Format

Jupyter Notebooks are an **open document** format based on **JSON**.

They contain a complete **record** of the user's **sessions** and *include code, narrative text, equations and rich output.*



UNIVERSITY
OF TRENTO



Jupyter Notebook



Open Standards for Interactive Computing



Notebook Document Format



Interactive Computing Protocol



The Kernel



Interactive
Computing Protocol

The *Notebook* **communicates with computational Kernels** using the **Interactive Computing Protocol**, an *open network protocol* based on **JSON** data over

ZMQ and WebSockets.





Jupyter Notebook



Open Standards for Interactive Computing



Notebook Document Format



Interactive Computing Protocol



The Kernel



The Kernel

Kernels are **processes** that **run interactive code** in a particular programming language and **return output** to the user.

Kernels also respond to **tab completion** and **introspection** requests.



UNIVERSITY
OF TRENTO



Jupyter Notebook



Installation

1. Install *Python*

2. Verify *Python & PyPI*

3. Install *Jupyter Notebook*

4. Verify *Jupyter*

```
# 1. Update and Refresh Repository Lists  
sudo apt update  
  
# 2. Install Supporting Software  
sudo apt install software-properties-common  
  
# 3. Add Deadsnakes PPA  
sudo add-apt-repository ppa:deadsnakes/ppa  
# 3.1. Refresh Repository Lists  
sudo apt update  
  
# 4. Install Python 3  
sudo apt install python3
```



Installation

1. *Install Python*
2. **Verify Python & PyPI**
3. *Install Jupyter Notebook*
4. *Verify Jupyter*



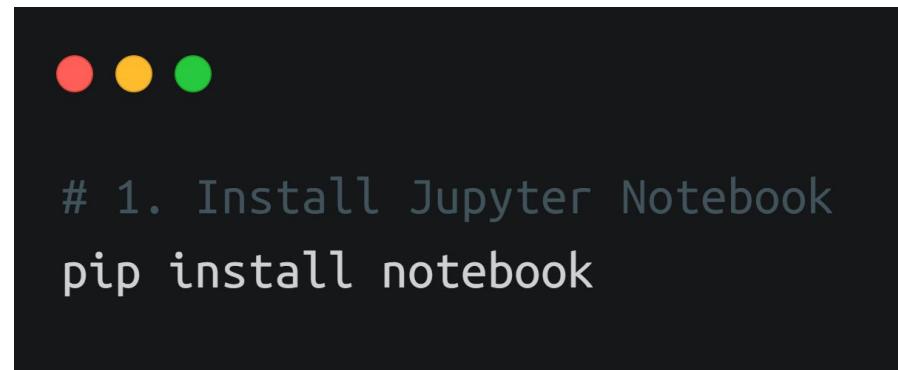
```
# 1. Verify Python  
python --version
```

```
# 2. Verify PyPI  
pip --version
```



Installation

1. *Install Python*
2. *Verify Python & PyPI*
- 3. Install Jupyter Notebook**
4. *Verify Jupyter*



A terminal window with a black background and white text. At the top are three colored circles: red, yellow, and green. Below them, the text reads "# 1. Install Jupyter Notebook" followed by the command "pip install notebook".

```
# 1. Install Jupyter Notebook
pip install notebook
```



Installation

1. *Install Python*
2. *Verify Python & PyPI*
3. *Install Jupyter Notebook*
4. **Verify Jupyter**



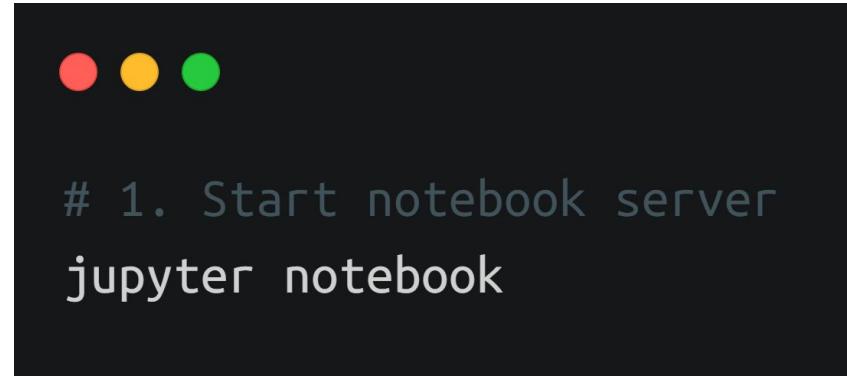
A dark terminal window with three colored dots (red, yellow, green) at the top. Below them, the text "# 1. Verify Jupyter" is displayed in light gray, followed by the command "jupyter --version" in white.

```
# 1. Verify Jupyter
jupyter --version
```



Running the Notebook

- 1. Start notebook server**
2. Notebook Dashboard
3. The Notebook

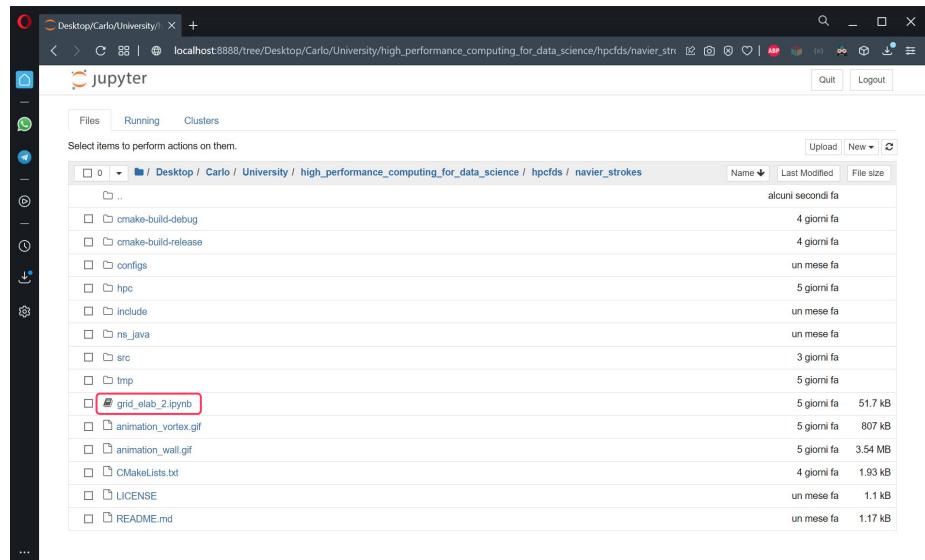


A terminal window with a black background and white text. At the top, there are three colored circles: red, yellow, and green. Below them, the text "# 1. Start notebook server" is displayed in a light blue color. Underneath that, the command "jupyter notebook" is shown in a larger, bold, white font.



Running the Notebook

1. Start notebook server
2. Notebook Dashboard
3. The Notebook





Jupyter Notebook



Running the Notebook

1. Start notebook server
2. Notebook Dashboard
3. The Notebook



UNIVERSITY
OF TRENTO

The screenshot shows a Jupyter Notebook window titled "grid_elab_2 - Jupyter Notebook". The code in cell [1] imports libraries for data manipulation and plotting. Cell [2] defines the base results path. Cells [3] through [7] show the process of listing files in the base folder, removing a 'gitkeep' file, and printing simulation metadata. The final output shows a JSON dump of the metadata for the first simulation.

```
In [1]: import os
import json
import numpy as np
import pandas as pd
import math

# Imports for the implementation of the gif
import freud
import matplotlib.pyplot as plt
import matplotlib.animation as animation

And define the C results paths.

In [2]: BASE_RESULTS = '/hpc/results/'

In [3]: files = os.listdir(BASE_RESULTS)
print(files)
['.gitkeep', 'simulation_0_1.json', 'simulation_1_2.json', 'simulation_2_3.json', 'simulation_3_4.json']

Remove gitkeep file

In [4]: files.remove('.gitkeep')
print(files)
['simulation_0_1.json', 'simulation_1_2.json', 'simulation_2_3.json', 'simulation_3_4.json']

Amount of simulations that were performed:

In [5]: len(files)
Out[5]: 4

For the moment we're considering only the first simulation

In [6]: file = open(BASE_RESULTS + files[0])
file = json.load(open(file))
print(file)

meta = file['metadata']
{'/hpc/results/simulation_0_1.json': None}

Metadata structure:

In [7]: print(json.dumps(meta, indent=2))
{
    "time_step": 0.01,
    "width": 200,
    "world": "square",
    "width_t": 100,
    "height_t": 100,
    "width_bounds": 102,
    "height_bounds": 102
}
```



Jupyter Notebook

Run current Cell

Interrupt the Kernel

Restart the Kernel & re-run whole notebook



Heading

Notebook for elaboration of raw JSON data into GIF images

Cell description

First we may want to import the libraries for data manipulation.

```
In [1]: import os
import json
import numpy as np
import pandas as pd
import math

# Imports for the implementation of the gif
import ffmpeg
import matplotlib.pyplot as plt
import matplotlib.animation as animation
```

And define the C results paths.

```
In [1]: BASE_RESULTS = '../hpc/results/'  
BASE_RESULTS
```

```
Out[1]: '../hpc/results/'
```

Cell
(Output Code)

Current Kernel language

Cell
(Input Code)



Jupyter Notebook

Jupyter Notebook in Project



Gif from raw JSON data



In our project, we use *Jupyter Notebook* to **transform** the raw **JSON** data, generated from the *HPC Navier Stokes* executable, into a **GIF** image that shows the fluid in all it's ticks.

*It's hard to understand a raw result with thousands of lines.
A GIF image is more "readable", easy and portable.*





Jupyter Notebook



Future



UNIVERSITY
OF TRENTO



Future

Jupyter Notebook is going to be **replaced** by *JupyterLab*.

For new projects, is **recommended to use the new standard** to increase *familiarity* and overall *experience* with the newer technology (event though they are very similar).



Why not using Jupyter in HPC4DS class next year ?

Each code example can be written in a *Notebook* and **shared with students** using *JupyterHub* or *Jupyter Voilà*.

The most important parts can be highlighted and described with code comments or Jupyter notes.



Jupyter Notebook

**Thanks for the
attention!**



Carlo Corradini § Massimiliano Fronza