

Introduction to Jupyter notebooks

*Case Study:
RESTful web service for solving
multidimensional time-
independent Schrödinger
equation*



"Ss. Cyril and Methodius" University - Skopje
FACULTY OF COMPUTER
SCIENCE AND ENGINEERING

What is a Notebook?

A *notebook* combines the functionality of

- a word processor — handles formatted text
- integrates code and its output into a single document
- combines visualizations, narrative text, mathematical equations, and other rich media

Single document where you can run code, display the output, add explanations, formulas, charts, and make your work more transparent, understandable, repeatable, and shareable.

IP[y]: Notebook spectrogram Last Checkpoint: a few seconds ago (autosaved) IPython (Python 3)

File Edit View Insert Cell Kernel Help

Code Cell Toolbar: None

Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#) using windowing, to reveal the frequency content of a sound signal.

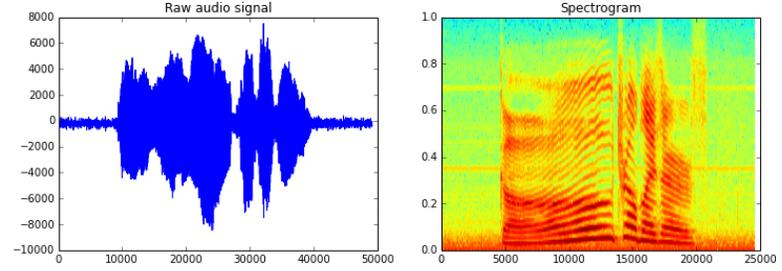
$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i kn} \quad k = 0, \dots, N-1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin specgram routine:

```
In [2]: %matplotlib inline
from matplotlib import pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```



The figure displays two side-by-side plots. The left plot, titled 'Raw audio signal', shows a blue waveform with a peak amplitude of approximately 8000 and a duration of about 50,000 samples. The right plot, titled 'Spectrogram', shows a heatmap of frequency content over time, with the x-axis representing time (0 to 25,000 samples) and the y-axis representing frequency (0.0 to 1.0). The spectrogram shows a clear pattern of energy concentrated in the lower frequency range, with a distinct peak around 10,000 samples.

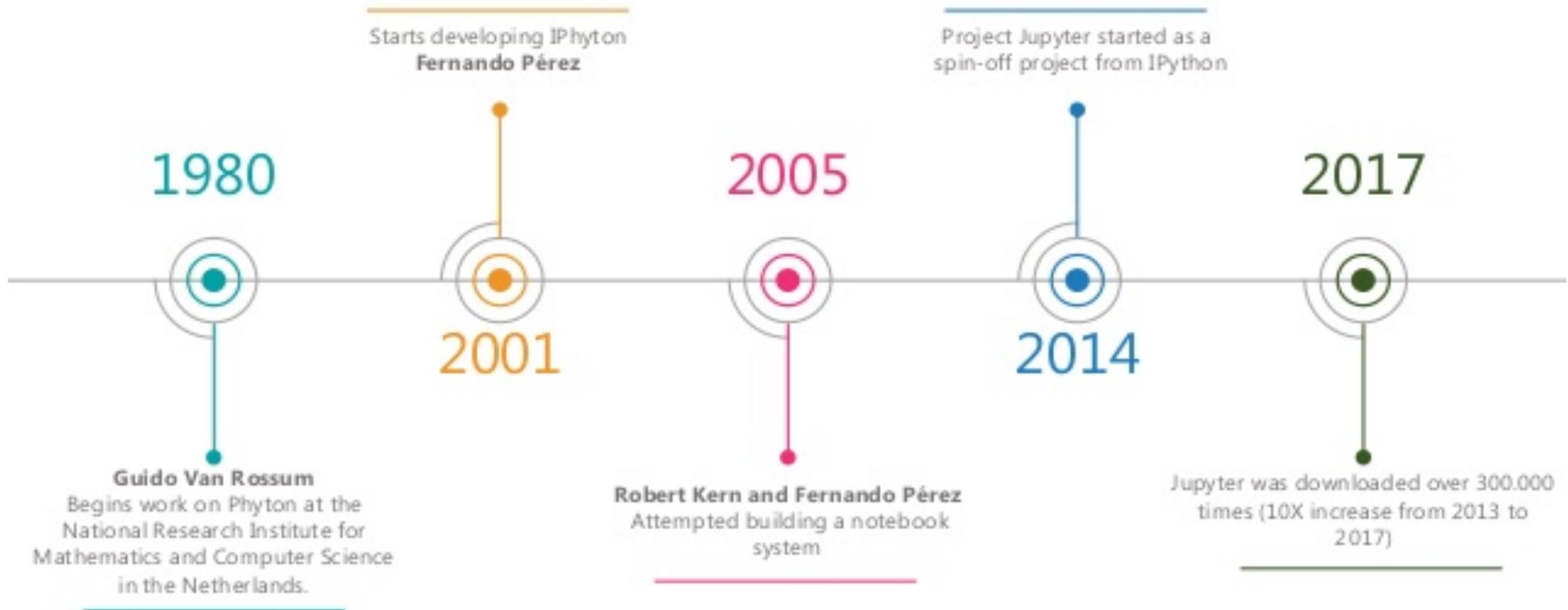
What is Jupyter Notebook?

- Incredibly powerful tool for interactively developing and presenting data science projects
- Open-source web application that you can use to create and share documents
- Can be installed locally (Python installed)
 - pip3 install jupyter (Linux, Mac OS x)
 - Anaconda Navigator (Windows)
- or run Online without installing anything
<https://jupyter.org/try>



Evolution of Jupyter

EVOLUTION OF JUPYTER



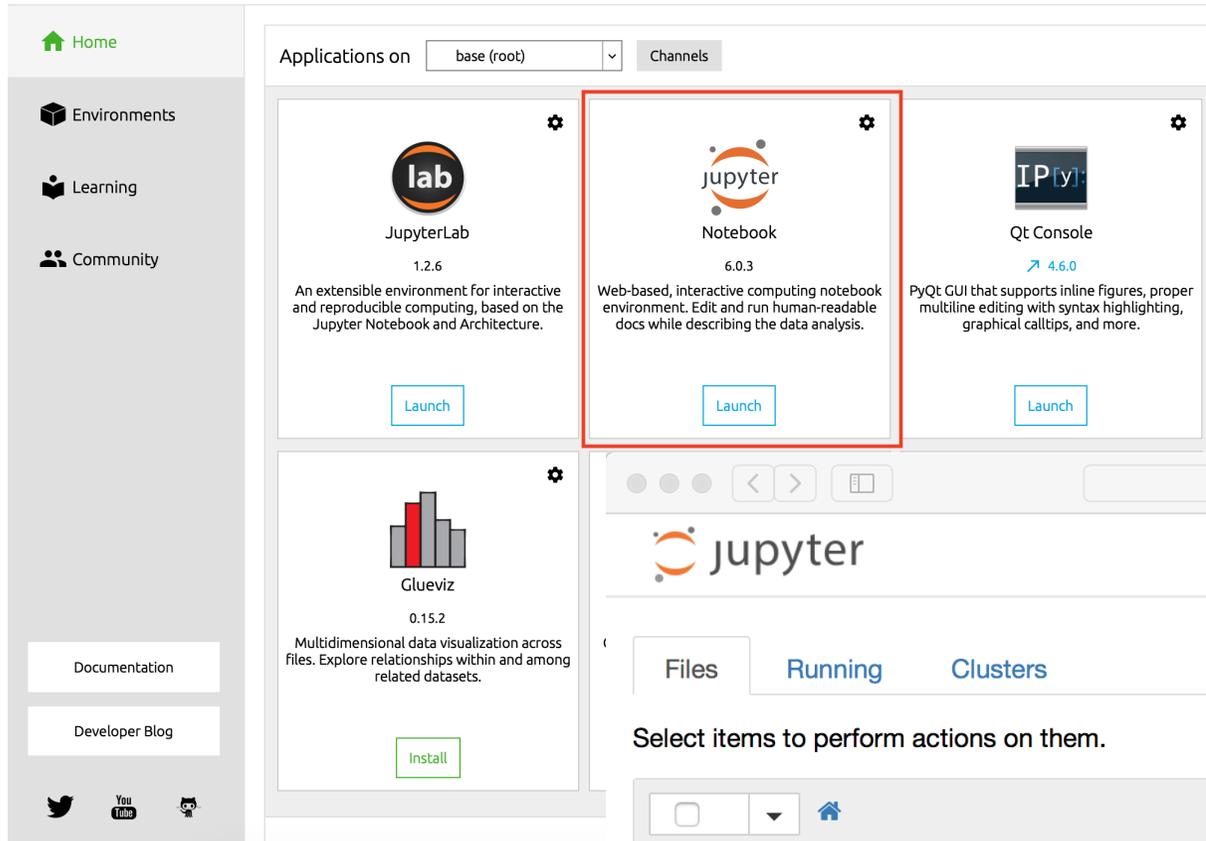


Jupyter is a loose acronym meaning **J**ulia, **P**ython, and **R**



How to launch Jupyter Notebook

 ANACONDA NAVIGATOR

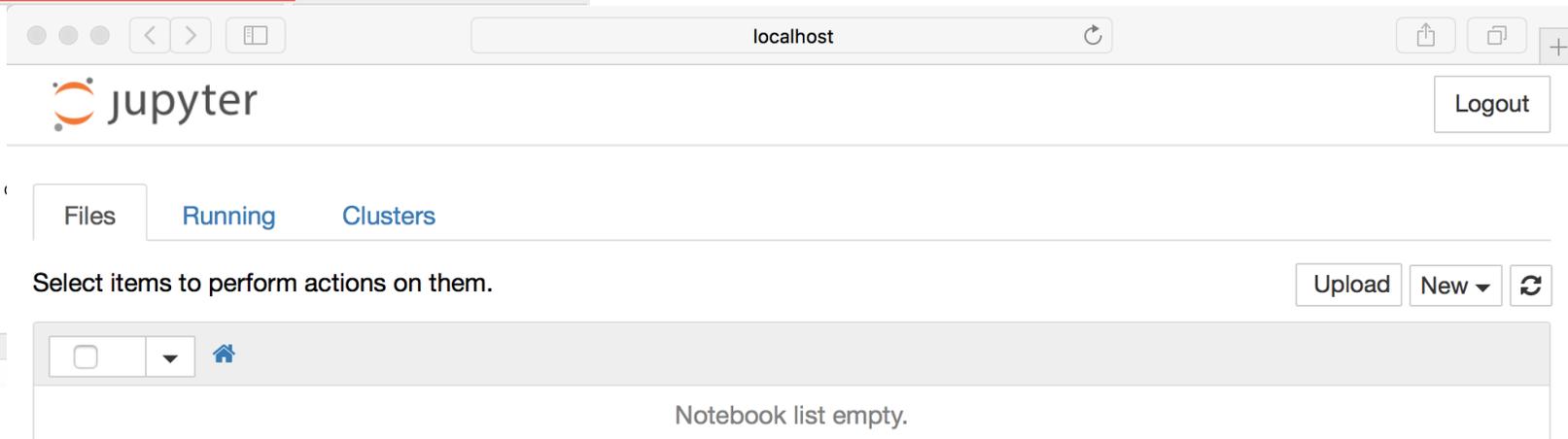


The screenshot shows the Anaconda Navigator interface. On the left is a sidebar with 'Home', 'Environments', 'Learning', and 'Community'. The main area displays 'Applications on base (root)'. Three application cards are visible: 'JupyterLab 1.2.6', 'Jupyter Notebook 6.0.3' (highlighted with a red box), and 'Qt Console 4.6.0'. Below them is a 'Glueviz 0.15.2' card. Each card has a 'Launch' button.

Shell

```
$ jupyter notebook
```

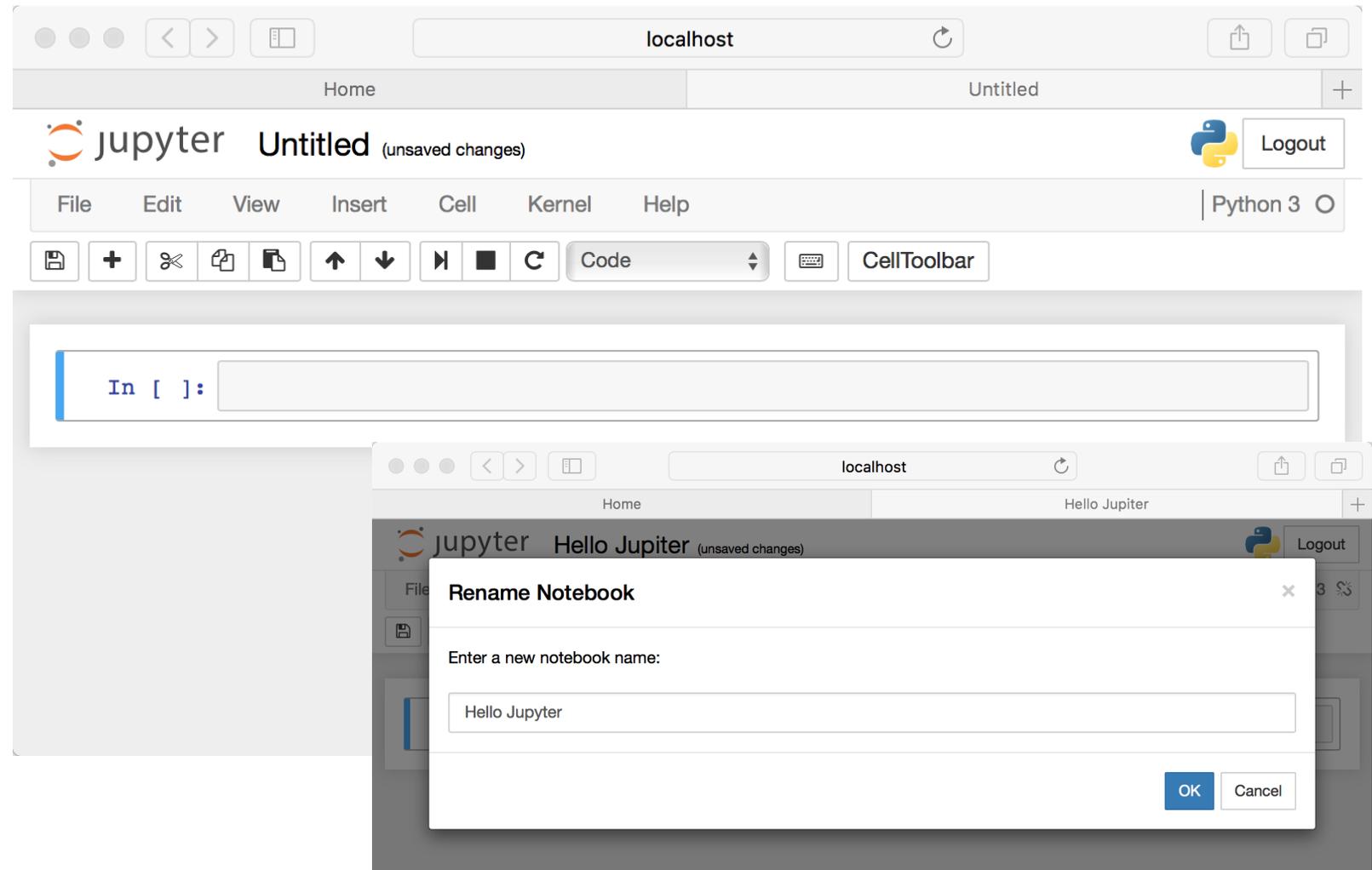
This will start up Jupyter and your default browser should start (or open a new tab) to the following URL: <http://localhost:8888/tree>



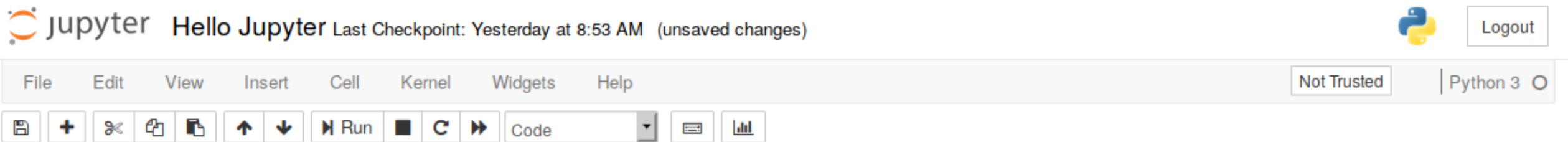
The screenshot shows a web browser window at 'localhost'. The page title is 'jupyter'. There is a 'Logout' button in the top right. Below the title bar are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' followed by 'Upload', 'New', and a refresh icon. Below this is a 'Notebook list empty.' message.

Creating Your First Notebook

- File-> New Notebook
- If you have Python 2 and Python 3 installed, you can create a Notebook that uses either of these
- If you switch back to the dashboard, you will see the new file Untitled.ipynb



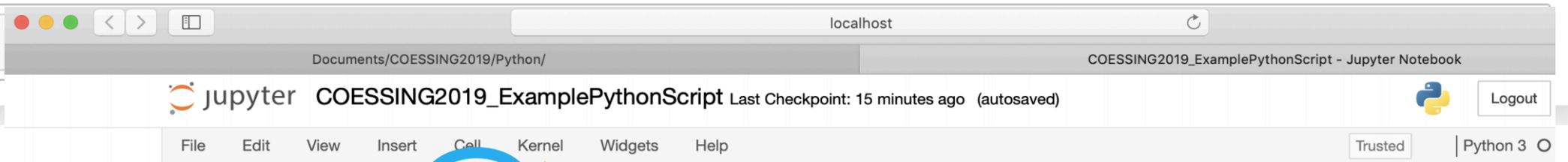
Running Cells



In [2]: `print('Hello Jupyter')`

Hello Jupyter

In []:



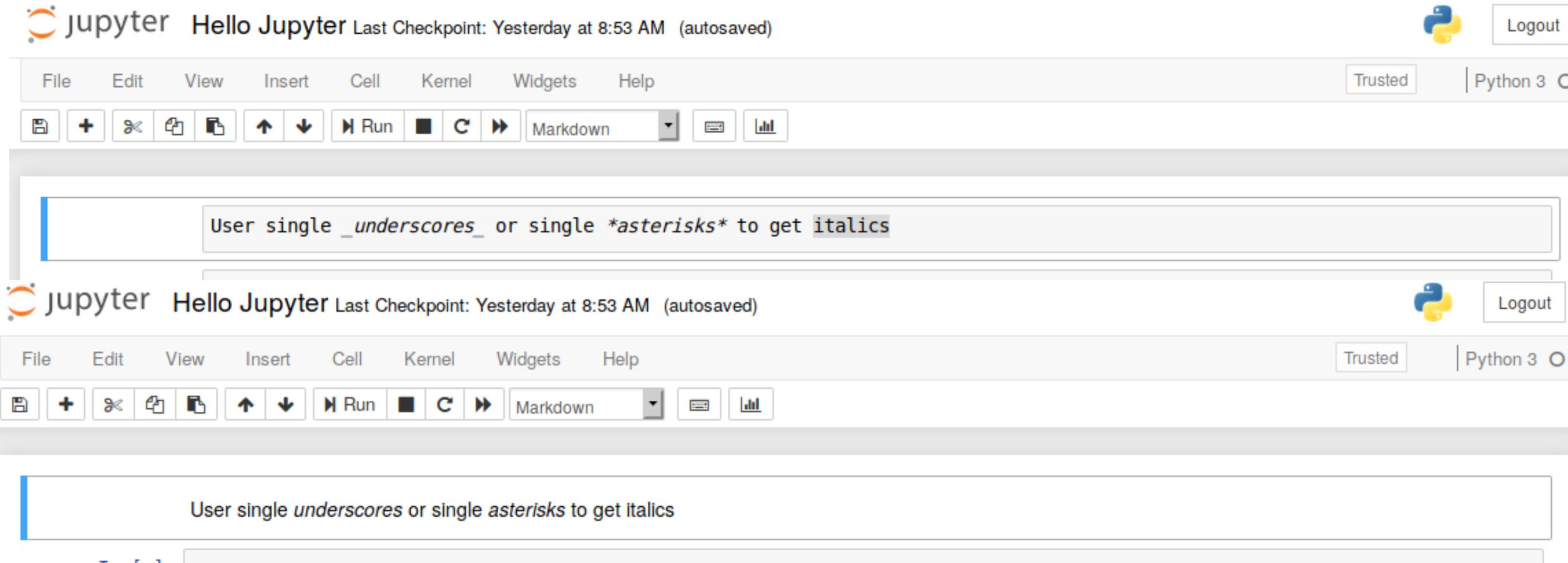
In [1]: `import numpy as np
import matplotlib.pyplot as plt
import pandas as pd`

Click here to
evaluate the cell

If there is a number here, then the cell has been evaluated! This means that the code written in that cell has been run. A cell is evaluated by either clicking "Run" or by typing Shift+Return.

Cell Type

- Cell -> Cell Type
 - Code, Markdown and Raw NBConvert
- Markdown - markup language that is a superset of HTML.



The image shows two screenshots of the Jupyter Notebook interface. The top screenshot shows a code cell with the text: `User single underscores or single asterisks to get italics`. The bottom screenshot shows the same cell after execution, where the text is rendered as: *underscores* or single *asterisks* to get italics. The interface includes a top bar with the Jupyter logo, the text "Hello Jupyter", and "Last Checkpoint: Yesterday at 8:53 AM (autosaved)". On the right, there is a Python logo and a "Logout" button. Below the top bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". A "Trusted" button and "Python 3" are also visible. The toolbar contains icons for saving, adding, deleting, copying, pasting, undo, redo, and running the cell. The cell type is set to "Markdown".

Headers

jupyter Hello Jupyter Last Checkpoint: Yesterday at 8:53 AM (unsaved changes)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3



```
# Header 1
## Header 2
### Header 3
```

jupyter Hello Jupyter Last Checkpoint: Yesterday at 8:53 AM (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3



Header 1

Header 2

Header 3

Lists

 jupyter Hello Jupyter Last Checkpoint: Yesterday at 8:53 AM (unsaved changes)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3

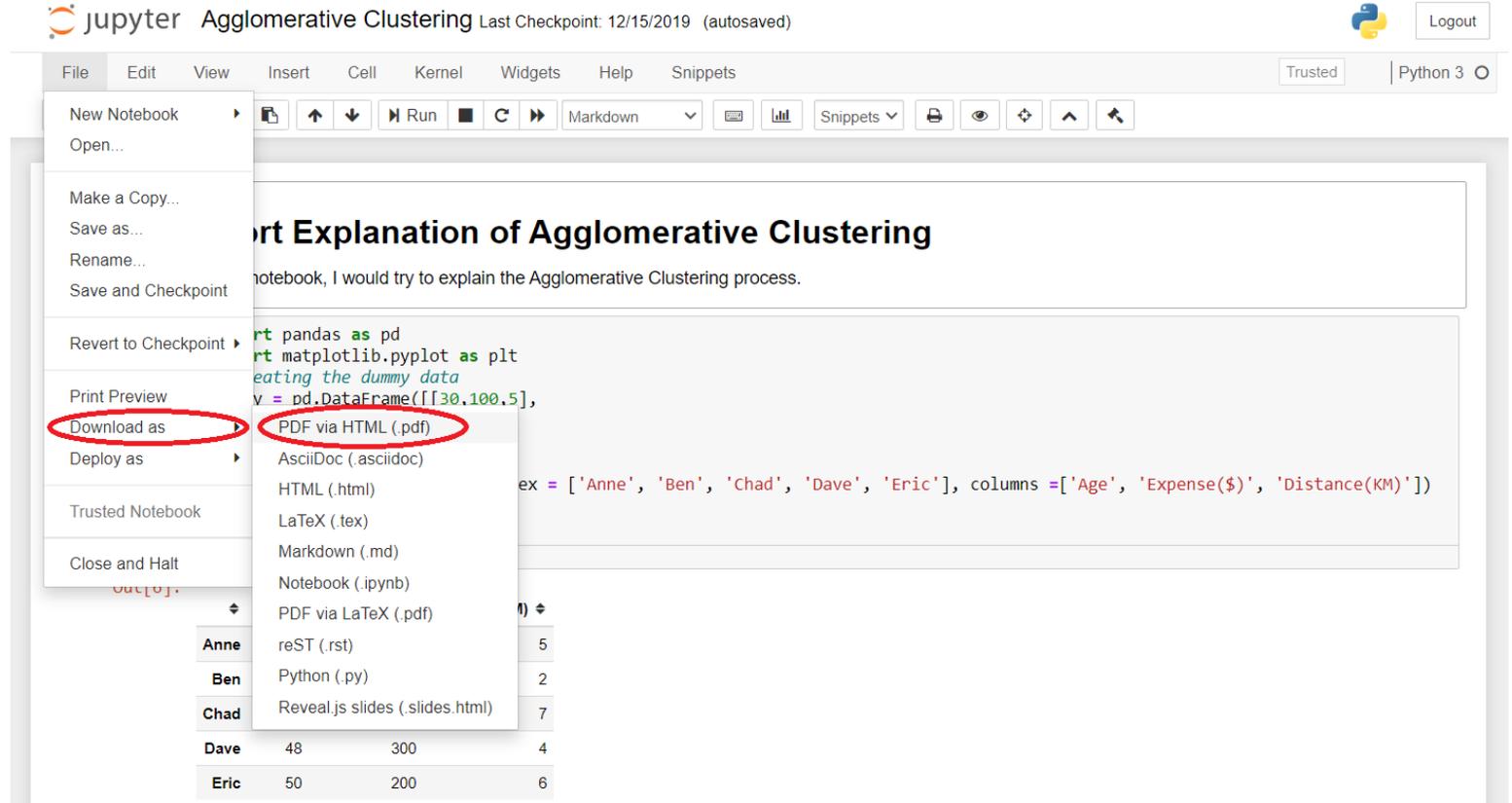


```
* item 1
* item 2
* item 3
  * sub-item 1
```

In []:

Exporting Notebooks

- HTML
- LaTeX
- PDF
- RevealJS
- Markdown
- ReStructured Text
- Executable script



The screenshot shows the JupyterLab interface for a notebook titled "Agglomerative Clustering". The "File" menu is open, and the "Download as" option is highlighted with a red circle. A sub-menu is visible, showing "PDF via HTML (.pdf)" also highlighted with a red circle. Other options in the sub-menu include AsciiDoc (.asciidoc), HTML (.html), LaTeX (.tex), Markdown (.md), Notebook (.ipynb), PDF via LaTeX (.pdf), reST (.rst), Python (.py), and Reveal.js slides (.slides.html). The notebook content includes a title "Short Explanation of Agglomerative Clustering" and a code cell with the following Python code:

```
import pandas as pd
import matplotlib.pyplot as plt

# Creating the dummy data
data = pd.DataFrame([[30,100,5],
                    [48,300,4],
                    [50,200,6]])

# Creating the dummy data
data = pd.DataFrame([[30,100,5],
                    [48,300,4],
                    [50,200,6]])

# Creating the dummy data
data = pd.DataFrame([[30,100,5],
                    [48,300,4],
                    [50,200,6]])

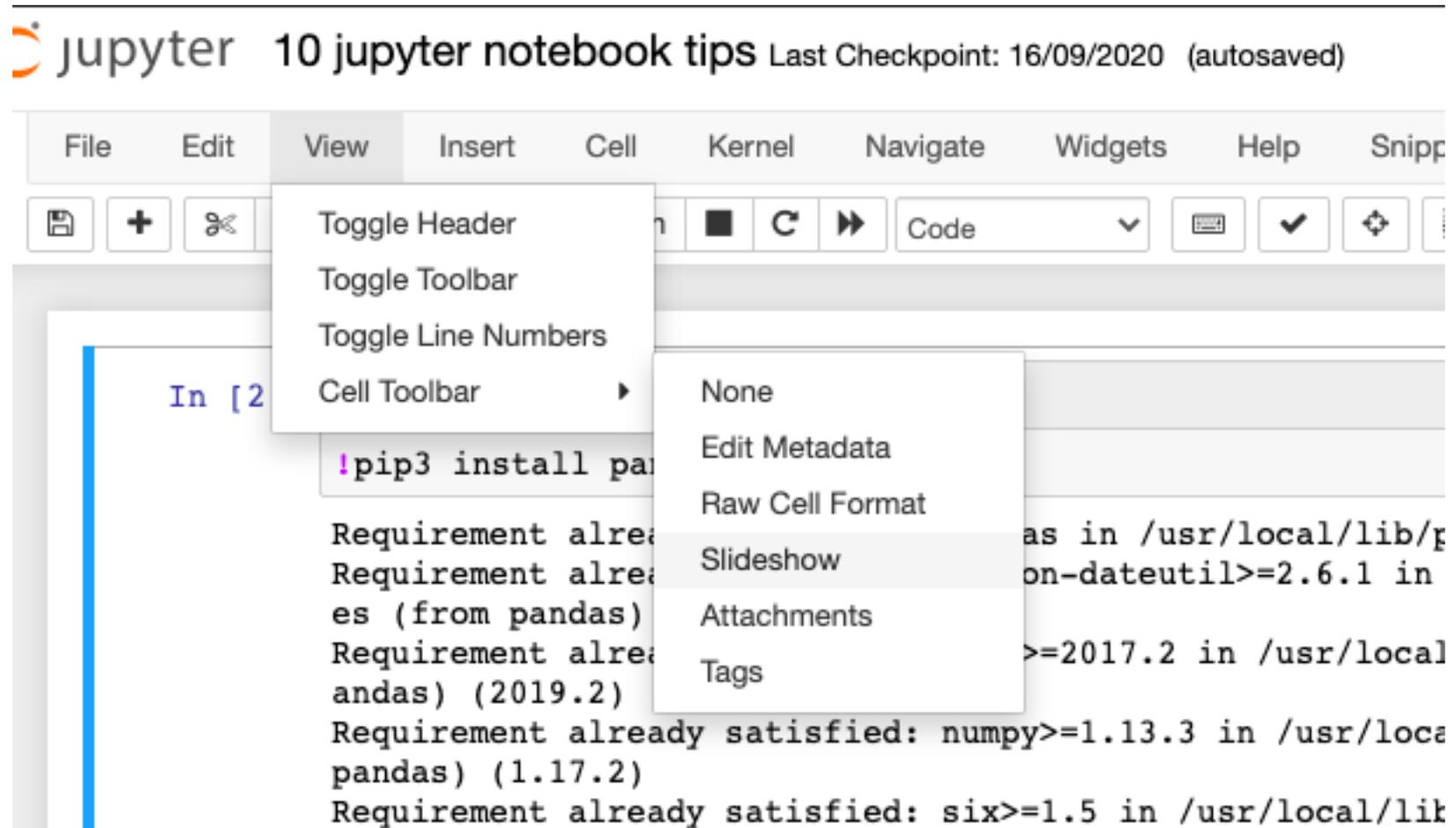
# Creating the dummy data
data = pd.DataFrame([[30,100,5],
                    [48,300,4],
                    [50,200,6]])
```

The code cell output shows a table with columns 'Age', 'Expense(\$)', and 'Distance(KM)'. The data is as follows:

	Age	Expense(\$)	Distance(KM)
Anne	30	100	5
Ben	48	300	4
Chad	50	200	6

Creating Presentations

View -> Cell Toolbar -> Slideshow



The screenshot shows the Jupyter Notebook interface. The top bar displays the Jupyter logo, the notebook title "10 jupyter notebook tips", and the last checkpoint information "Last Checkpoint: 16/09/2020 (autosaved)". Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Navigate, Widgets, Help, and Snippets. The "View" menu is open, showing options: Toggle Header, Toggle Toolbar, Toggle Line Numbers, and Cell Toolbar. The "Cell Toolbar" sub-menu is also open, showing options: None, Edit Metadata, Raw Cell Format, Slideshow (highlighted), Attachments, and Tags. The main content area shows a code cell with the command `!pip3 install pandas` and its output, which includes installation requirements for pandas and its dependencies (numpy and six).

Creating Presentations

Once you have done this each cell should have a Slide Type option in the right-hand corner of the cell

In [4]:

```
%matplotlib inline

import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# Data for plotting sine function
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)

plt.plot(t, s);
```



Slide Type

-
- ✓ Slide
- Sub-Slide
- Fragment
- Skip
- Notes

Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#) using windowing, to reveal the frequency content of a sound signal.

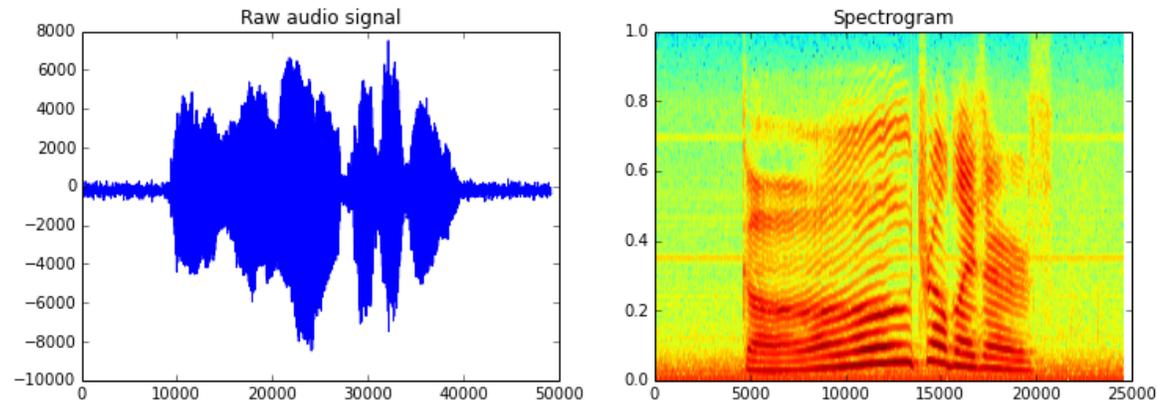
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin spectrogram routine:

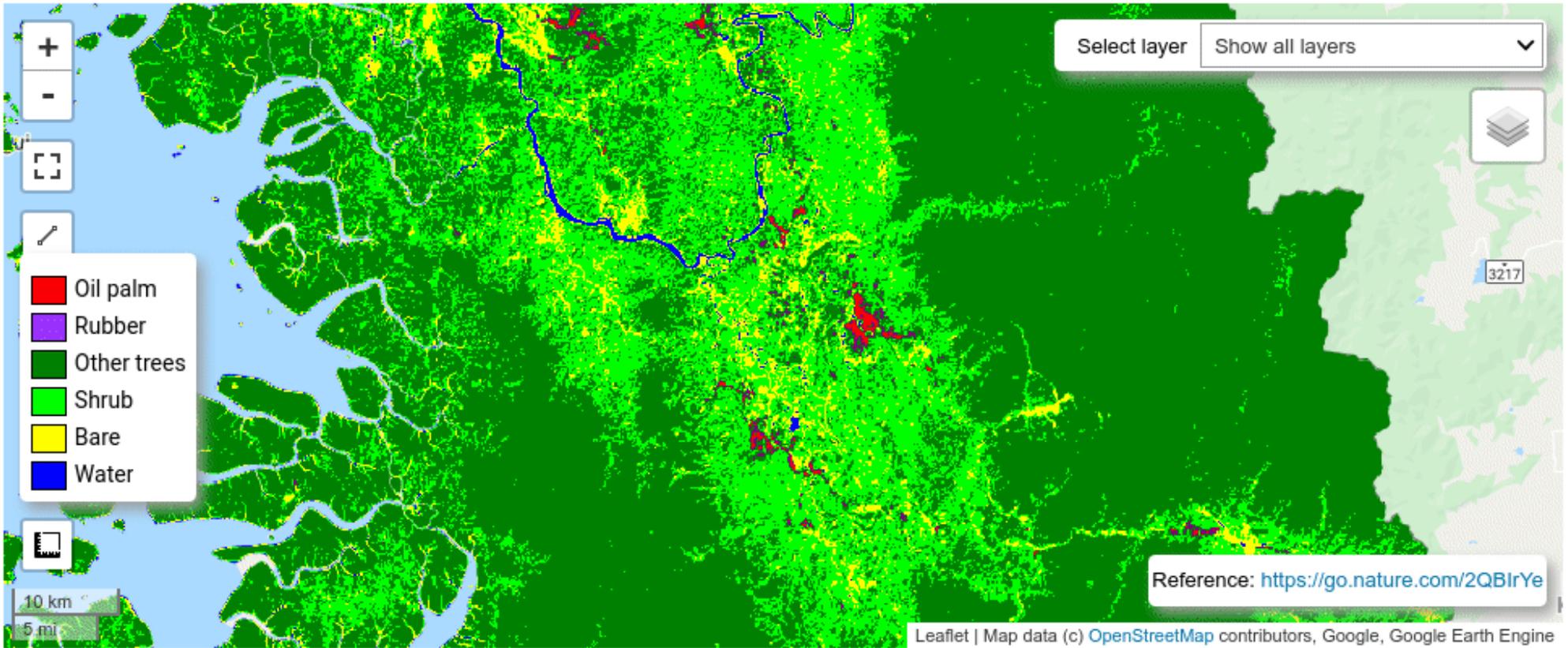
```
In [2]: %matplotlib inline
from matplotlib import pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```



Create an interactive map

The default basemap is `Google Satellite`. [Additional basemaps](#) can be added using the `Map.add_basemap()` function.

```
In [2]: Map = geemap.Map(center=[40,-100], zoom=4)
Map.add_basemap('ROADMAP') # Add Google Map
Map
```



Home Page - Select or create a r... x 02_Data_processing - Jupyter No... x +

https://hub.gke.mybinder.org/user/slicer-slicernotebooks-ybnh0x9d/notebooks/02_Data_processing.ipynb#

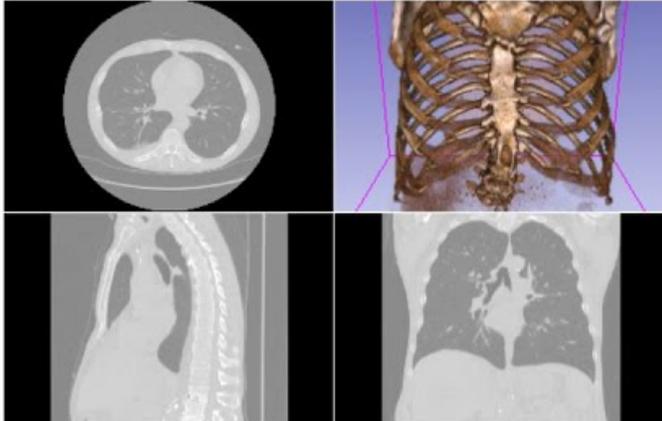
jupyter 02_Data_processing (unsaved changes) Visit repo Copy Binder link

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Slicer 4.11

```
# Volume render a small (30%) image
slicernb.AppWindow.setWindowSize(scale=0.3)
slicernb.showVolumeRendering(volume)
slicernb.showSliceViewAnnotations(False)

display(slicernb.ViewDisplay("FourUp"))

# Disable volume rendering to make sure it does not slow down execution
slicernb.showVolumeRendering(volume, False)
```

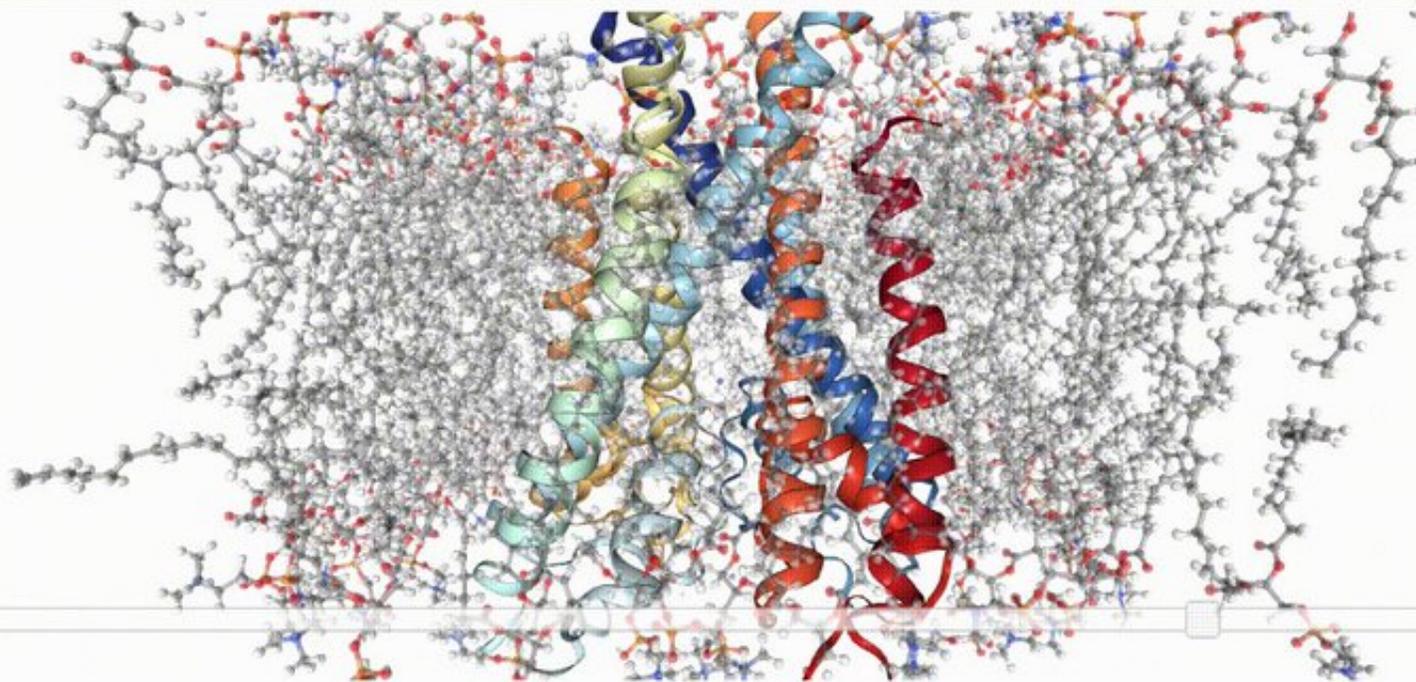


```
In [ ]: # Print all parameters of a CLI module
cliModule = slicer.modules.grayscalemodelmaker
n=cliModule.cliModuleLogic().CreateNode()
for groupIndex in range(n.GetNumberOfParameterGroups()):
    for parameterIndex in range(n.GetNumberOfParametersInGroup(groupIndex)):
        print('Parameter {{0}}/{{1}}: {{2}} ({{3}})'.format(groupIndex, parameterIndex, n.GetParameterName(groupIndex, parameterIndex),
```

01_Data_loading_a...ipynb ... Show all x

```
In [1]: import pytraj as pt
import nglview as nv
```

```
In [2]: traj = pt.load('sim.nc', top='sim.prmtop')
view = nv.show_pytraj(traj)
view
```



```
In [3]: view.clear()
view.add_cartoon('protein', color_scheme='residueindex')
view.add_ball_and_stick('not protein', opacity=0.5)
```

Jupyter DEMO

A gallery of interesting Jupyter Notebooks:

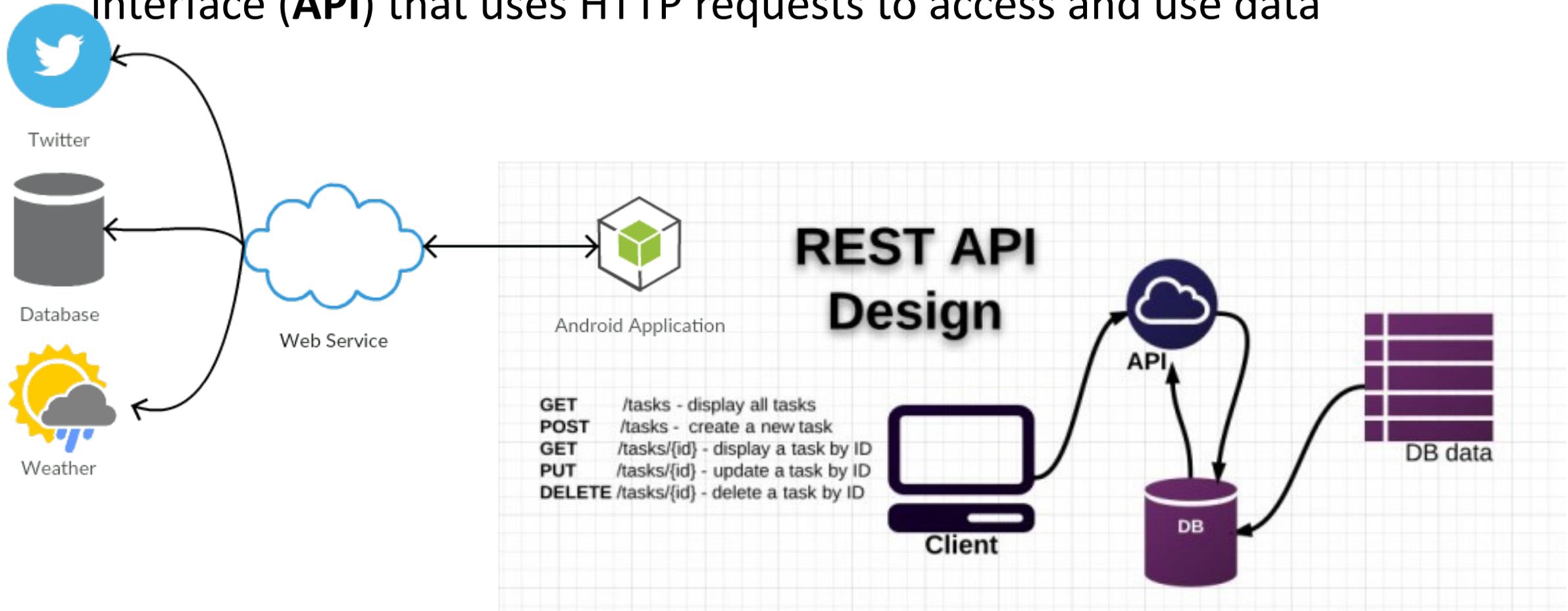
<https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks#programming-and-computer-science>

RESTful web service for solving multidimensional time-independent Schrödinger equation using Hermite DVR approach

- solution of **one-dimensional**, **two-dimensional** and **three-dimensional** time-independent Schrödinger equation based on the the Gauss-Hermite Discrete Variable Representation (DVR) approach
- The solution of [one-dimensional](#) Schrödinger equation is illustrated in the case of following model potentials:
 - [Morse potential](#);
 - [Simple Harmonic Oscillator \(SHO\) potential](#);
 - [Sombrero potential \(Mexican hat\)](#);
 - [Woods-Saxon potential](#).
- Solutions of [two-dimensional](#) and [three-dimensional](#) Schrödinger equations are illustrated for the following two model potentials: multidimensional Morse potential and multidimensional SHO potential.

What is RESTful API?

- A **RESTful API** is an architectural style for an application program interface (**API**) that uses HTTP requests to access and use data



- **Separation between the client and the server**
- **Visibility, reliability and scalability**
- **The REST API is always independent of the type of platform or languages**
 - With a REST API you can have PHP, Java, Python or Node.js servers....
 - The only thing is that it is indispensable that the responses to the requests should always take place in the language used for the information exchange, normally XML or JSON.

Example: 1D Morse potential

- Returns a one-dimensional Morse potential $V(x)$:
 - $V(x) = D * (1 - \exp(-a * (x - x_0)))^2 - D$
- Parameters:
 - **npts** - number of points (default value 10)
 - **D** - dissociation depth (default value 3.0)
 - **a** - inverse "width" of the potential (default value 0.5)
 - **x0** - equilibrium bond distance (default value 0.0)
 - **prec** - precision (default value 6)

How to consume RESTful API?

- You **type** <http://194.149.135.58:8080/SchrodingerAPI/1dHermiteMorse> into the address bar of your **browser**.

```
[-2.41671645 -1.39124794 -0.28535681 1.09633735 7.42311473]
```

- What if we want to use the results in our code?
 - Copy them? NO
 - Consume the RESTful web service in our code
 - DEMO in Jupyter

How to consume RESTful API?

Example with default parameters:

```
import requests
response = requests.get('http://194.149.135.58:8080/SchrodingerAPI/1dHermiteMorse')
if response.status_code == 200:
    print(response.content.decode('utf-8'))
else:
    print("None")
```

```
[-2.41671645 -1.39124794 -0.28535681 1.09633735 7.42311473]
```

Example with parameters (npts=10, D=3.0, a=0.5, x0=0.0, prec=6):

```
import requests
response = requests.get('http://194.149.135.58:8080/SchrodingerAPI/1dHermiteMorse?npts=10&D=3.0&a=0.5&x0=0.0&prec=6')
if response.status_code == 200:
    print(response.content.decode('utf-8'))
else:
    print("None")
```

```
[-2.418857 -1.441745 -0.643831 0.244456 1.40058 2.844225 4.900026 6.987975 19.116035 61.33091 ]
```

Schrödinger equation RESTful API DEMO

Why data scientists love Jupyter notebooks?

- **All in one place**
 - **Easy to share**
 - **Easy to convert**
 - **Language independent**
 - **Easy to customize**
 - **Stress-free Reproducible experiments**
 - **Effective teaching-cum-learning tool**
 - **Interactive code and data exploration**
- One analysis of the code-sharing site GitHub counted more than 2.5 million public Jupyter notebooks in September 2018
 - Jupyter was awarded the [2017 ACM Software Systems Award](#) — a prestigious honor it shares with Java, Unix, and the Web.



How Netflix uses Jupyter Notebook?

195 million

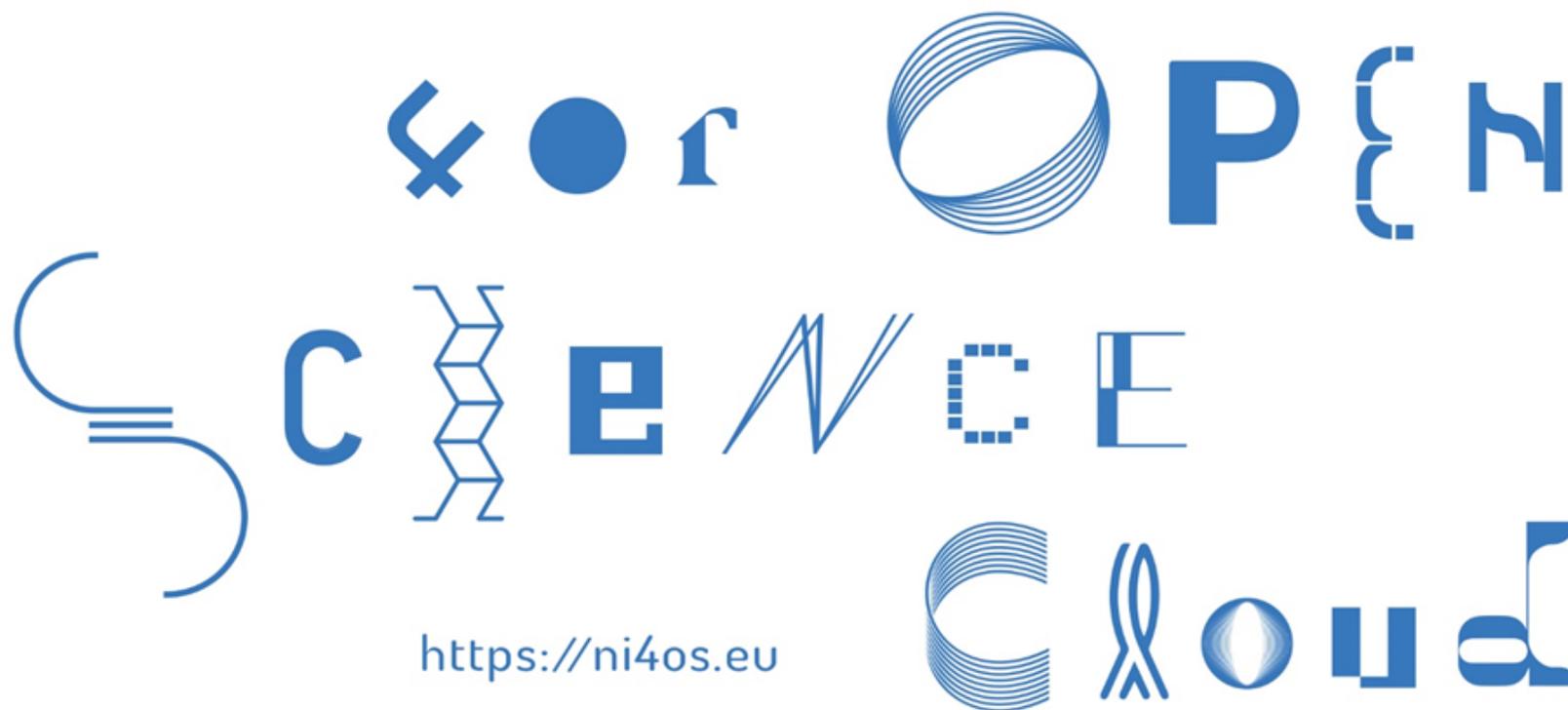
paid



- Notebooks were first introduced at Netflix to support data science workflows.
- To help the platform scale, they wanted to minimize the number of tools and the solution to this was the open-source tool: Jupyter notebooks.
- Users have programmatic access to virtually the entire platform from within a notebook.
- Netflix is going all-in on notebooks in production by migrating over 10k workflows to notebooks and using them as a way to bridge the chasm between technical and non-technical users

- Bojana Koteska
 - bojana.koteska@finki.ukim.mk
- Ljupco Pejov
 - ljupcop@pmf.ukim.mk
- Anastas Mishev
 - anastas.mishev@finki.ukim.mk





[NI4OS eu](https://twitter.com/NI4OS_eu)



[NI4OS](https://facebook.com/NI4OS)



[ni4os.eu](https://instagram.com/ni4os.eu)