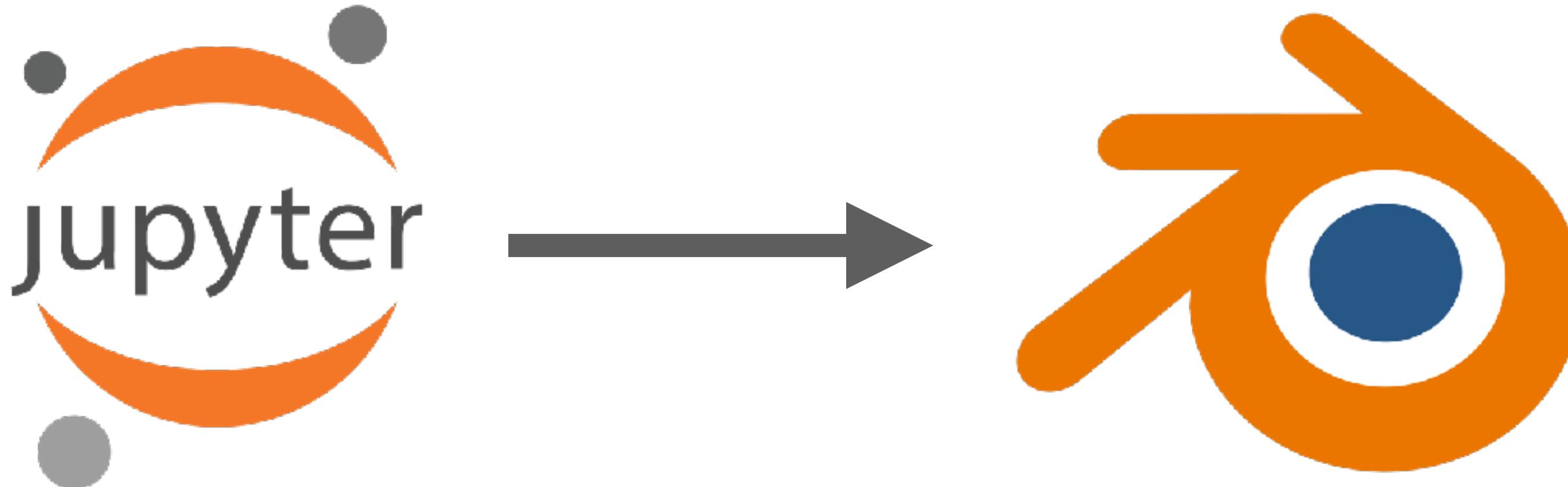


Python-Blender-Workshop

Jan-Hendrik Müller

12.11.2024



Ablaufplan

- 0: Intro
- 1: Objects in Blender
- 2: Using Python Packages
- 3: Using Meshes
- 4: Manipulating Data Using Pandas
- 5: Introduction to Geometry Nodes
- 6: Advanced Pandas Operations
- 7: Using Attributes in Shaders
- 8: Exploring Climate Data
- 9: (Data From JavaScript)

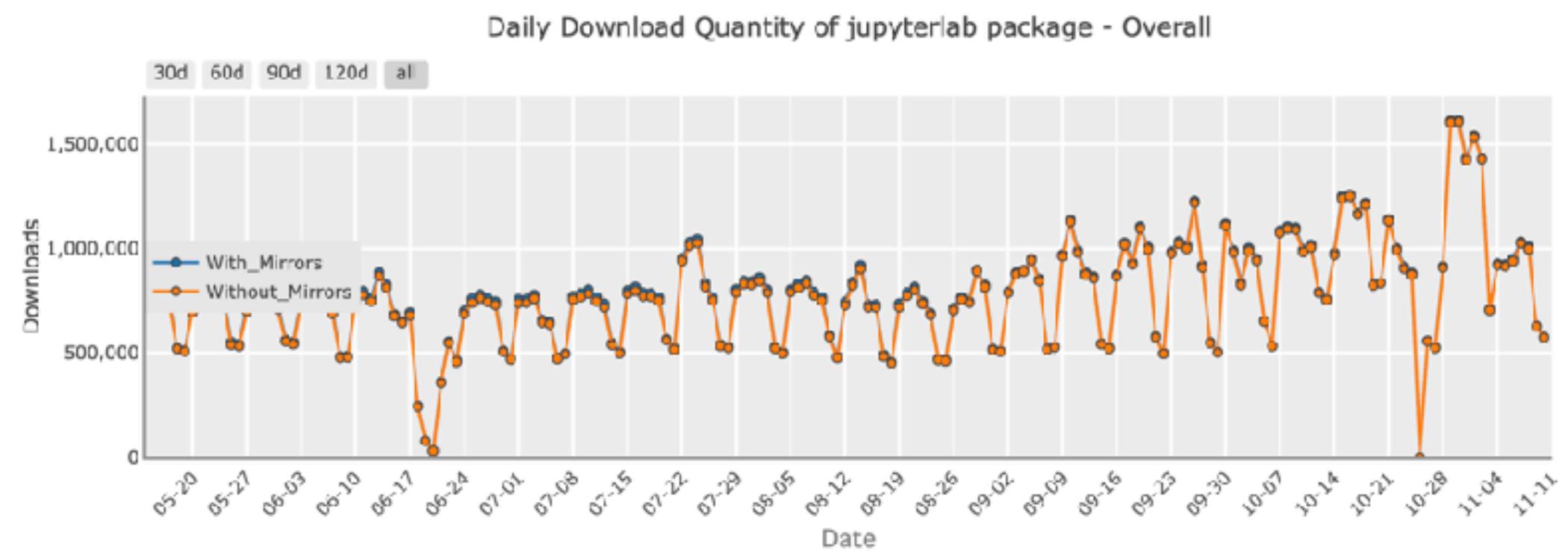


<https://github.com/Octoframes/bpy-workshop/blob/main/task1.ipynb>



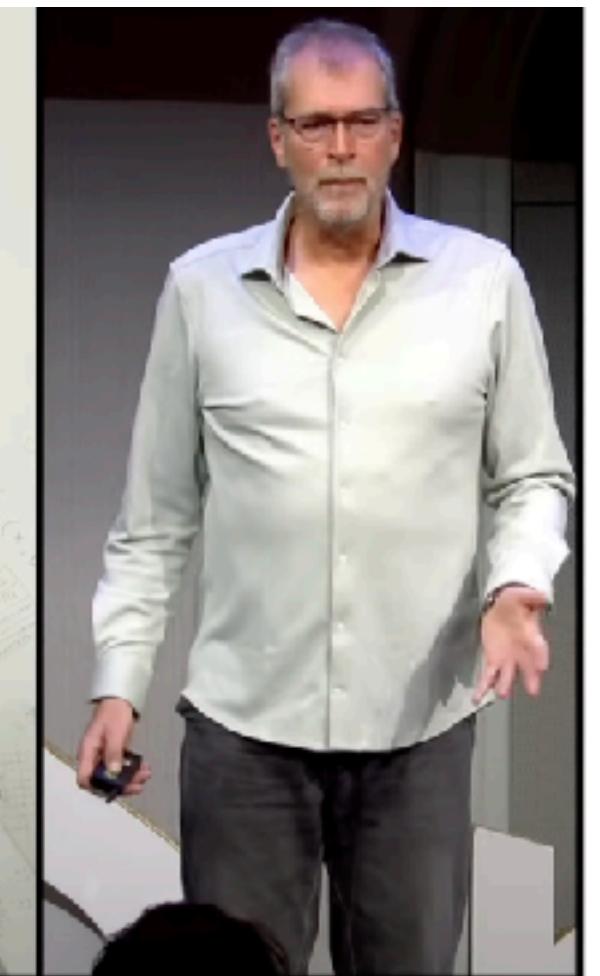


Downloads last month: 29,176,117



How many (active) users in 2023

- 20 Million downloads (without ad blocker)



Keynote – Blender Conference 2024

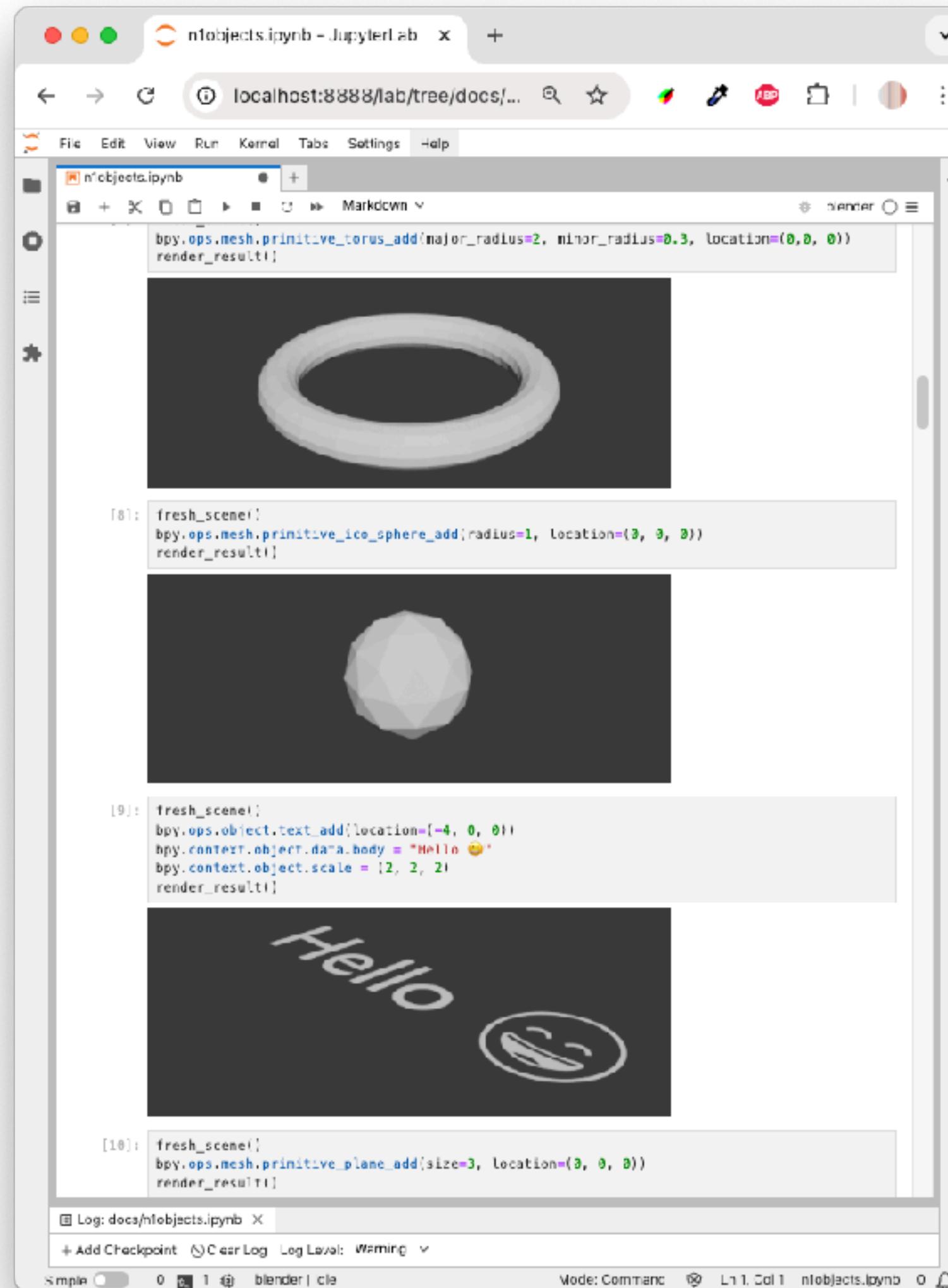


Abonnieren

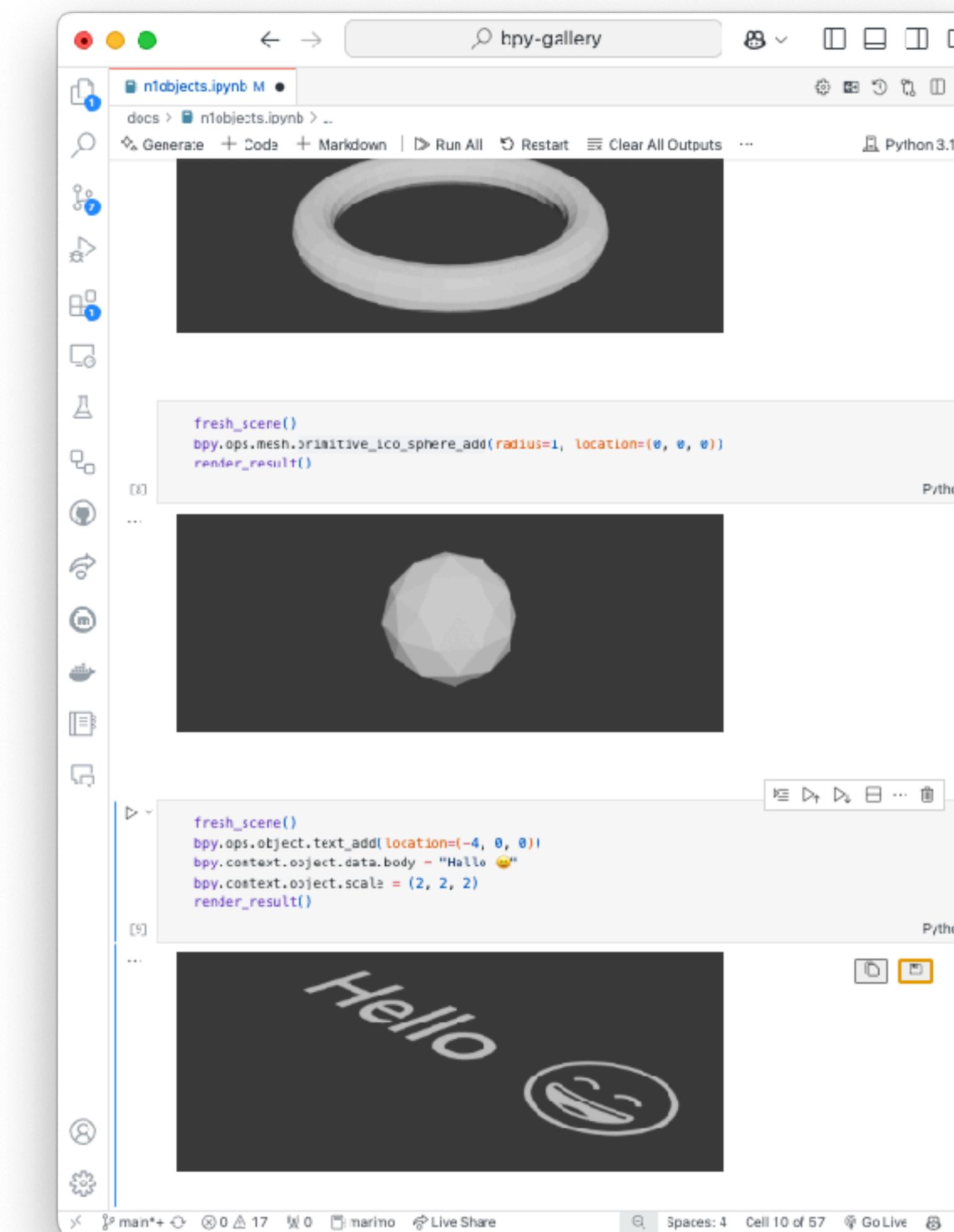
1405 Tellen Herunterladen Clip ...

Notebook Platformen

Jupyter Lab



VS Code Notebook



Notebook Platformen

Satyrn

The screenshot shows a notebook interface with three code cells and their corresponding 3D render outputs:

- Cell 7:** Code:

```
fresh_scene()
bpy.ops.mesh.primitive_torus_add(major_radius=2, minor_radius=0.3, 1
render_result()
```

 Output: A 3D rendering of a torus (donut shape) on a black background.
- Cell 8:** Code:

```
fresh_scene()
bpy.ops.mesh.primitive_ico_sphere_add(radius=1, location=(0, 0, 0))
render_result()
```

 Output: A 3D rendering of a low-poly icosphere on a black background.
- Cell 9:** Code:

```
fresh_scene()
bpy.ops.object.text_add(location=(-4, 0, 0))
bpy.context.object.data.body = "Hello 😊"
bpy.context.object.scale = (2, 2, 2)
render_result()
```

 Output: A 3D rendering of a text object with the text "Hello 😊" and a small 3D logo, both scaled up.

Marimo

The screenshot shows a notebook interface with three code cells and their corresponding 3D render outputs:

- Cell 1:** Code:

```
fresh_scene()
bpy.ops.mesh.primitive_torus_add(major_radius=2, minor_radius=0.3,
location=(0,0, 0))
render_result()
```

 Output: A 3D rendering of a torus (donut shape) on a black background.
- Cell 2:** Code:

```
fresh_scene()
bpy.ops.mesh.primitive_ico_sphere_add(radius=1, location=(0, 0, 0))
render_result()
```

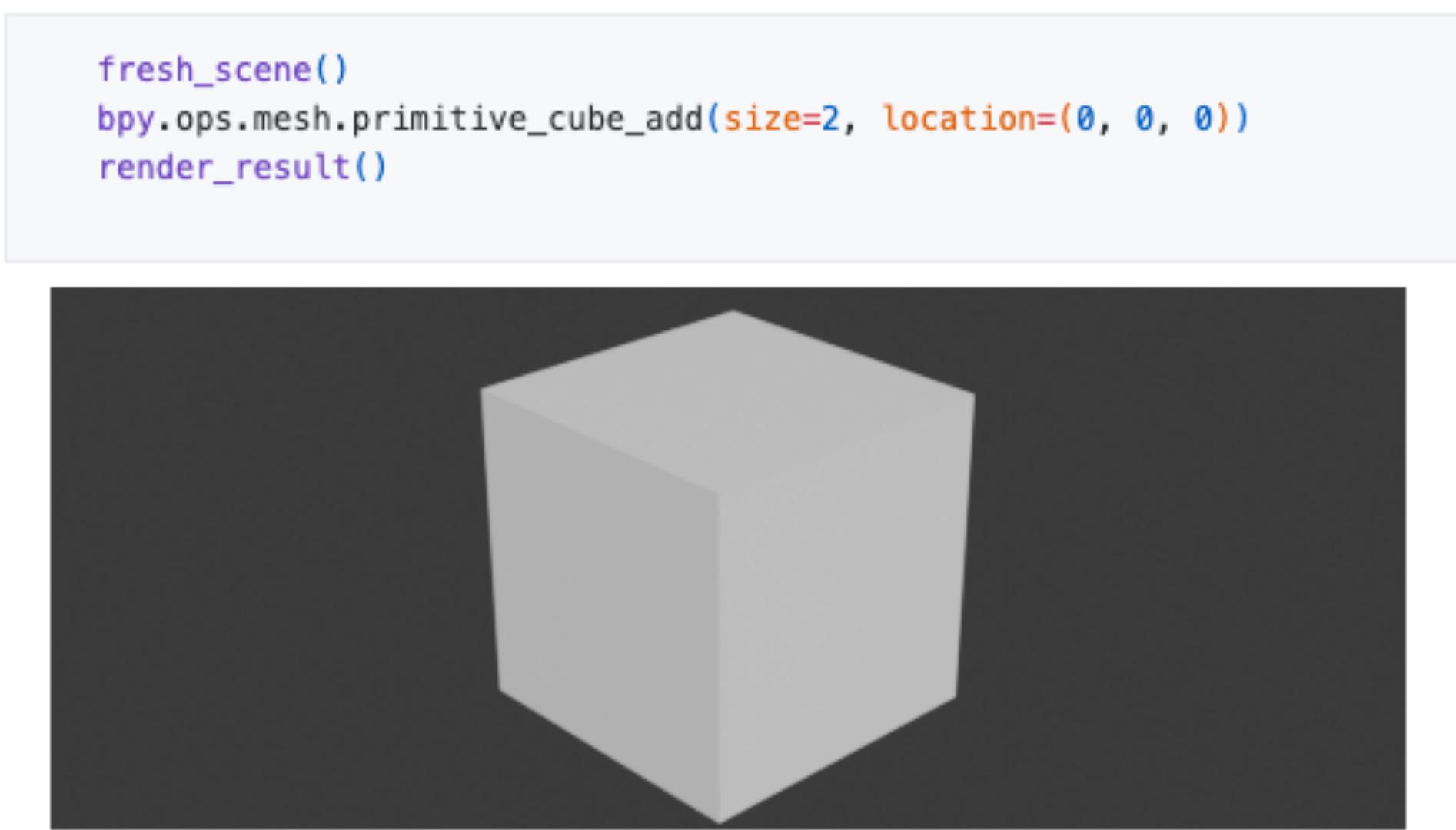
 Output: A 3D rendering of a low-poly icosphere on a black background.
- Cell 3:** Code:

```
fresh_scene()
bpy.ops.object.text_add(location=(-4, 0, 0))
bpy.context.object.data.body = "Hello 😊"
bpy.context.object.scale = (2, 2, 2)
render_result()
```

 Output: A 3D rendering of a text object with the text "Hello 😊" and a small 3D logo, both scaled up.

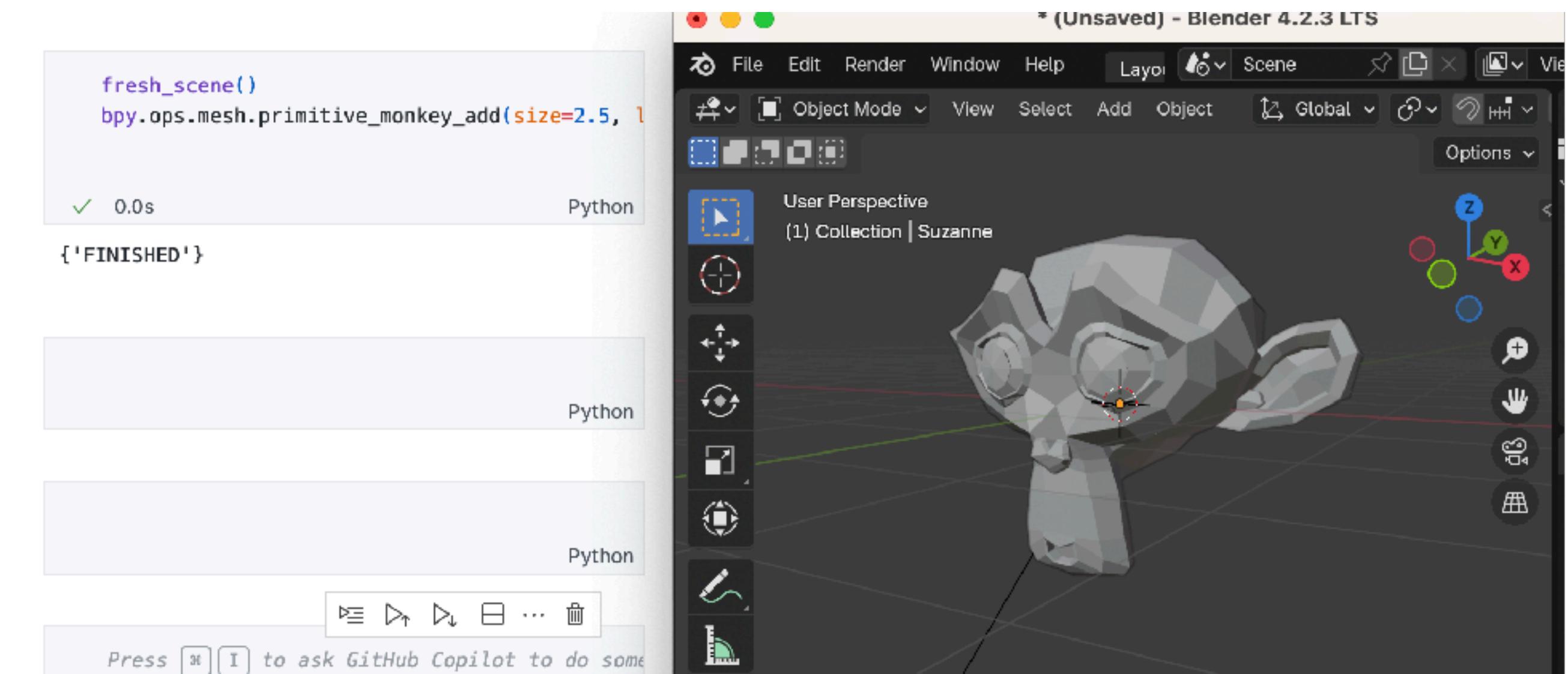
Headless mode

Ohne Blender GUI



GUI mode

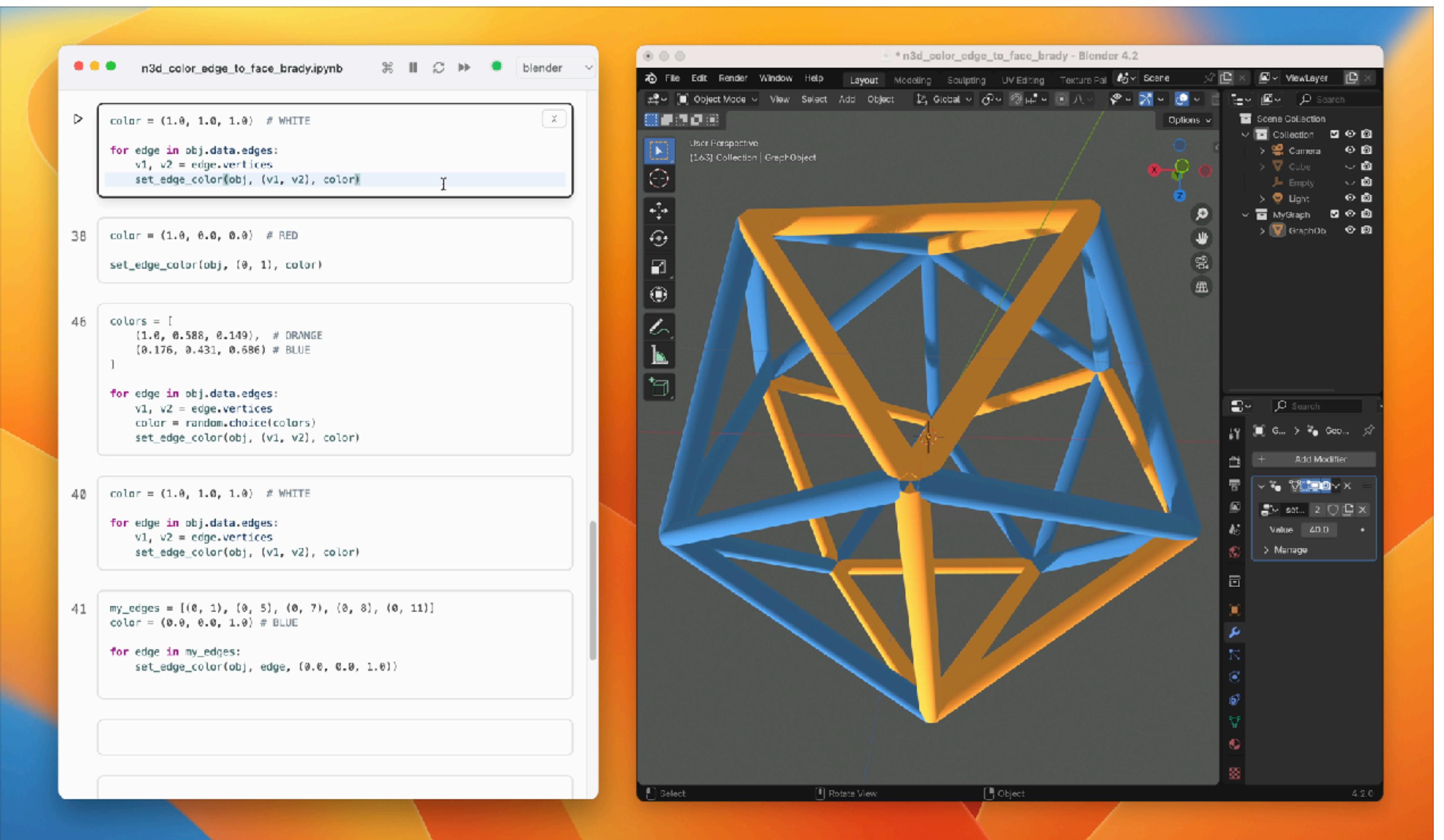
Blender GUI an der Seite



Notebook Pros & Cons

+ Interaktives Coding

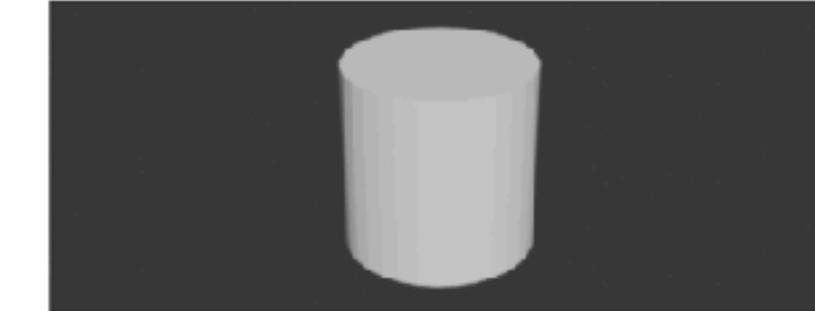
+ Einfaches Debugging



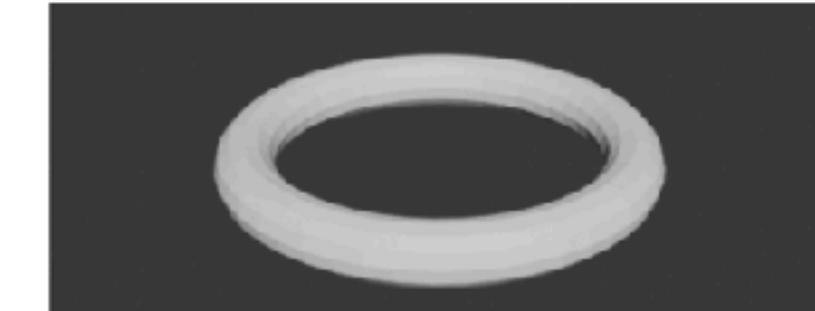
Notebook Pros & Cons

+ Dokumentation und Code in einem

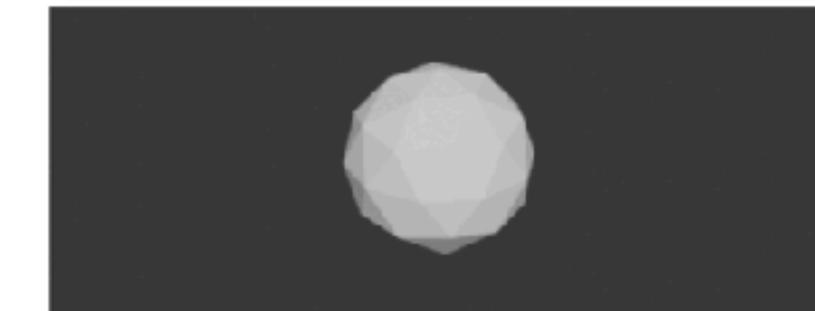
```
In [6]: fresh_scene()  
bpy.ops.mesh.primitive_cylinder_add(radius=1, depth=2, location=[0, 0, 0])  
render_result()
```



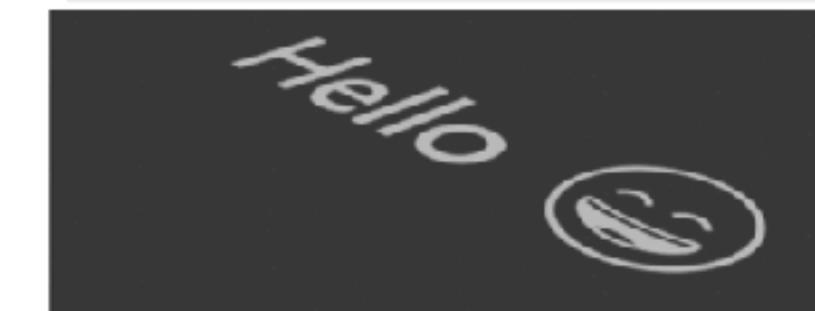
```
In [7]: fresh_scene()  
bpy.ops.mesh.primitive_torus_add(major_radius=2, minor_radius=0.3, location=[0, 0, 0])  
render_result()
```



```
In [8]: fresh_scene()  
bpy.ops.mesh.primitive_ico_sphere_add(radius=1, location=[0, 0, 0])  
render_result()
```

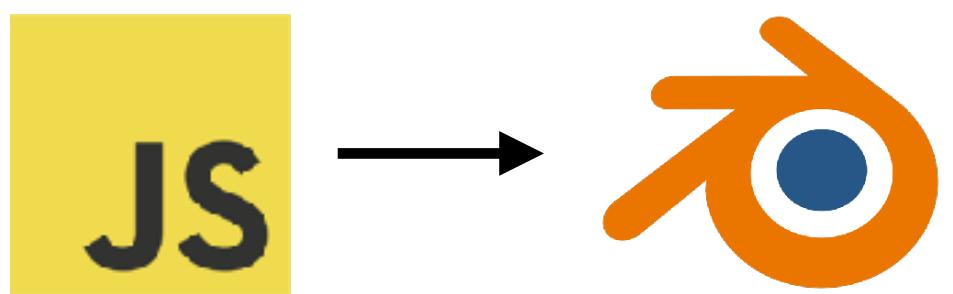


```
In [9]: fresh_scene()  
bpy.ops.object.text_add(location=(-4, 0, 0))  
bpy.context.object.data.body = "Hello 😊"  
bpy.context.object.scale = (2, 2, 2)  
render_result()
```

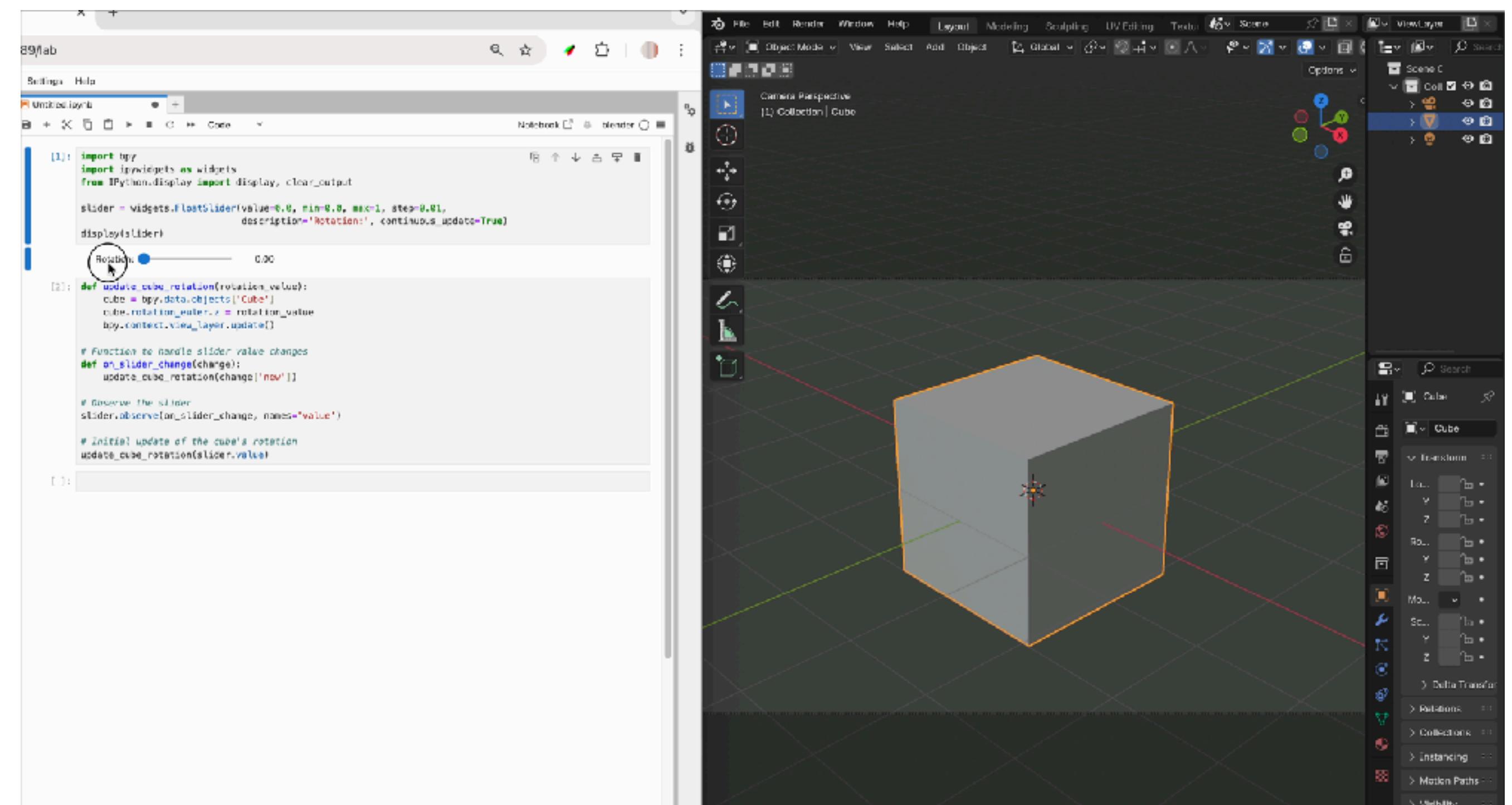


[https://kolibril13.github.io/
bpy-gallery/n1objects/](https://kolibril13.github.io/bpy-gallery/n1objects/)

Notebook Pros & Cons



+ Integration von Widgets



Notebook Pros & Cons

- Hidden state

Linearly executed notebook

```
[1]: one = 1
[2]: two = 2
[3]: three = one + two
[4]: three
[4]: 3
```

Notebook with hidden state

```
[1]: one = 1
[2]: two = 2
[4]: three = one + two
[5]: three
[5]: 42
```

<https://ploomber.io/images/blog/nbs-myths/hidden-state.png>

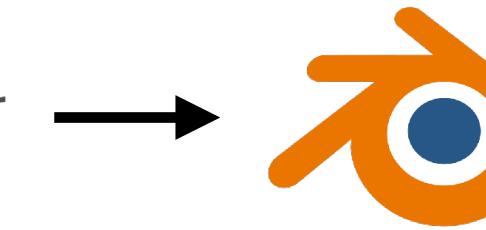
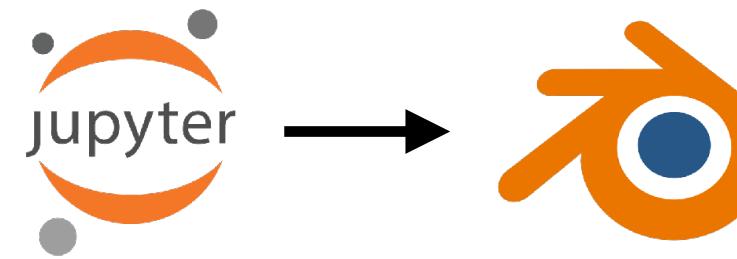
Notebook Pros & Cons

- Code in Notebooks schnell unübersichtlich bei großer Codbasis

Notebook Pros & Cons

-> Gut geeignet, um Ideen auszuprobieren!

Python Modul laden



The image shows a dual-screen setup. On the left screen is a Jupyter Notebook interface with a tab titled "multiple_networks.ipynb". The notebook contains Python code for generating and rendering network graphs. The first cell (cell 2) imports NetworkX and bpy, creates a balanced tree graph with 4 nodes, and draws it with spheres. The second cell (cell 3) creates a balanced tree graph with 6 nodes and draws it with spheres. The third cell uses IPython's display function to render the graph as an image and display it. On the right screen is the Blender 4.1 interface in Object Mode. It displays a complex, branching network graph where each node is represented by a glowing orange sphere. The Blender Outliner panel on the right lists various objects and materials. The Render properties panel at the bottom shows settings for the render resolution (1000x1000), file path ("tmp/test.png"), and render type ("render_still=True"). The Node Editor panel at the bottom shows a Principled BSDF node with "Value" set to 14.200, connected to a Material Output node.

```
[2]: import networkx as nx
import bpy

G = nx.balanced_tree(4, 4)
node_positions = nx.spring_layout(G, dim=3, scale=1.9)
edges = list(G.edges)
draw_network(node_positions, edges, sphere_radius = 0.05)

[3]: G = nx.balanced_tree(6, 2)
node_positions = nx.spring_layout(G, dim=3, scale=1.9)
edges = list(G.edges)
draw_network(node_positions, edges, sphere_radius = 0.1)

[ ]: from IPython.display import display, Image

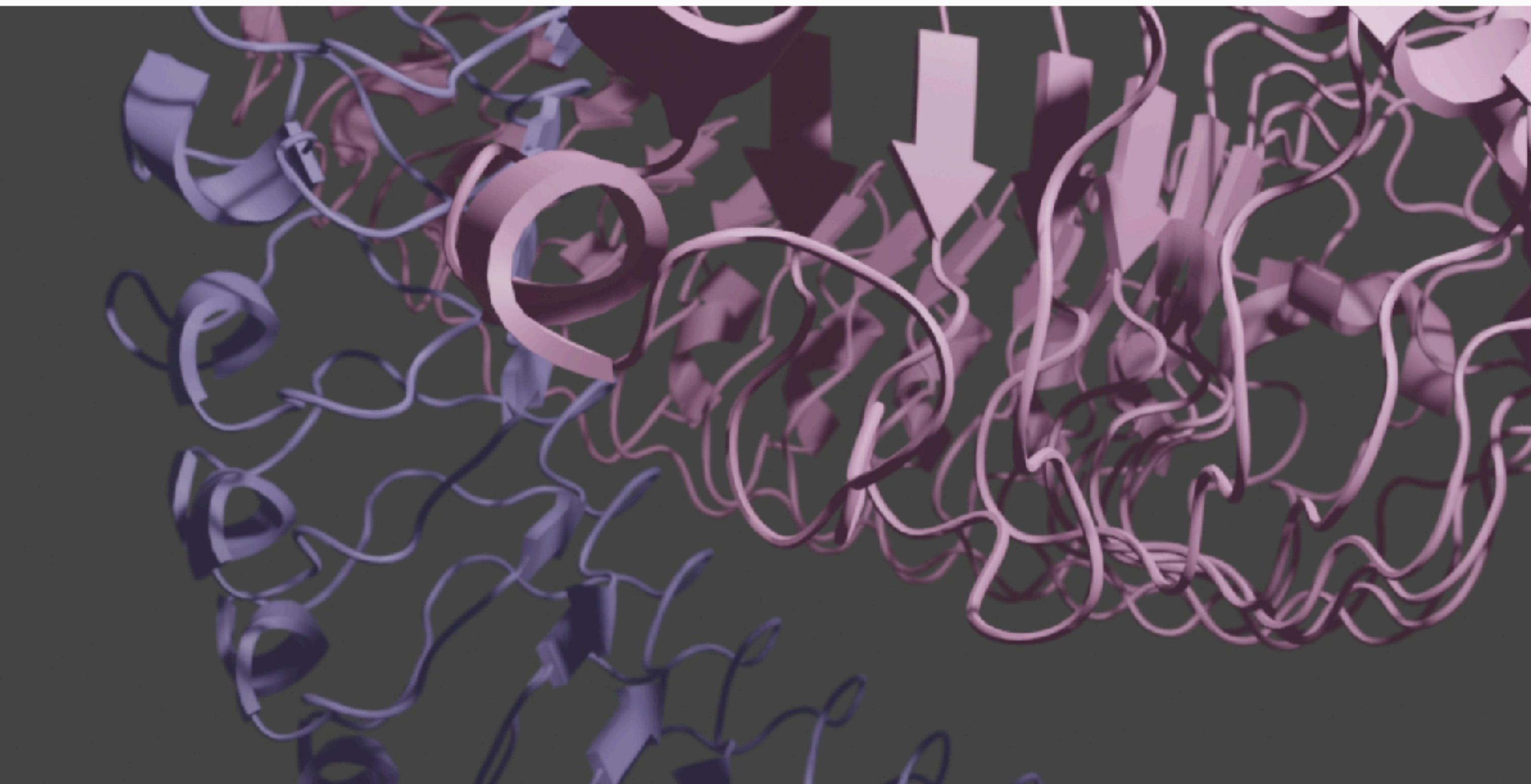
bpy.context.scene.render.resolution_x = 1000
bpy.context.scene.render.resolution_y = 1000

path = "/tmp/test.png"
bpy.context.scene.render.filepath = path
bpy.ops.render.render(write_still=True)

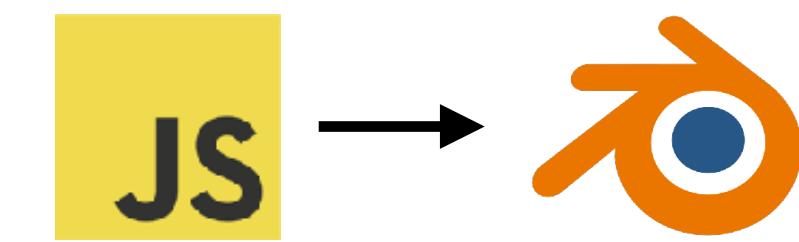
display(Image(filename=path , width=400))
```

Python Modul laden

```
[2]: import molecularnodes as mn  
molecule = mn.load.molecule_rcsb("7TY6", starting_style="cartoon", center_molecule=True)  
molecule.location = (0, 2, 1)  
scale = 8  
molecule.scale = (scale,scale,scale)  
  
render_image()
```



Blender & Whiteboard



The image shows a Blender 3D Viewport with a DNA model selected. A 3D cursor is positioned at the top right, with its local coordinate system (X, Y, Z) indicated by colored spheres (red for X, green for Y, blue for Z). The Viewport has a wireframe grid background. On the left, there's a code editor window showing two snippets of Python code:

```
[2]: from tldraw import TldrawWidgetCoordinates
widget = TldrawWidgetCoordinates()
widget

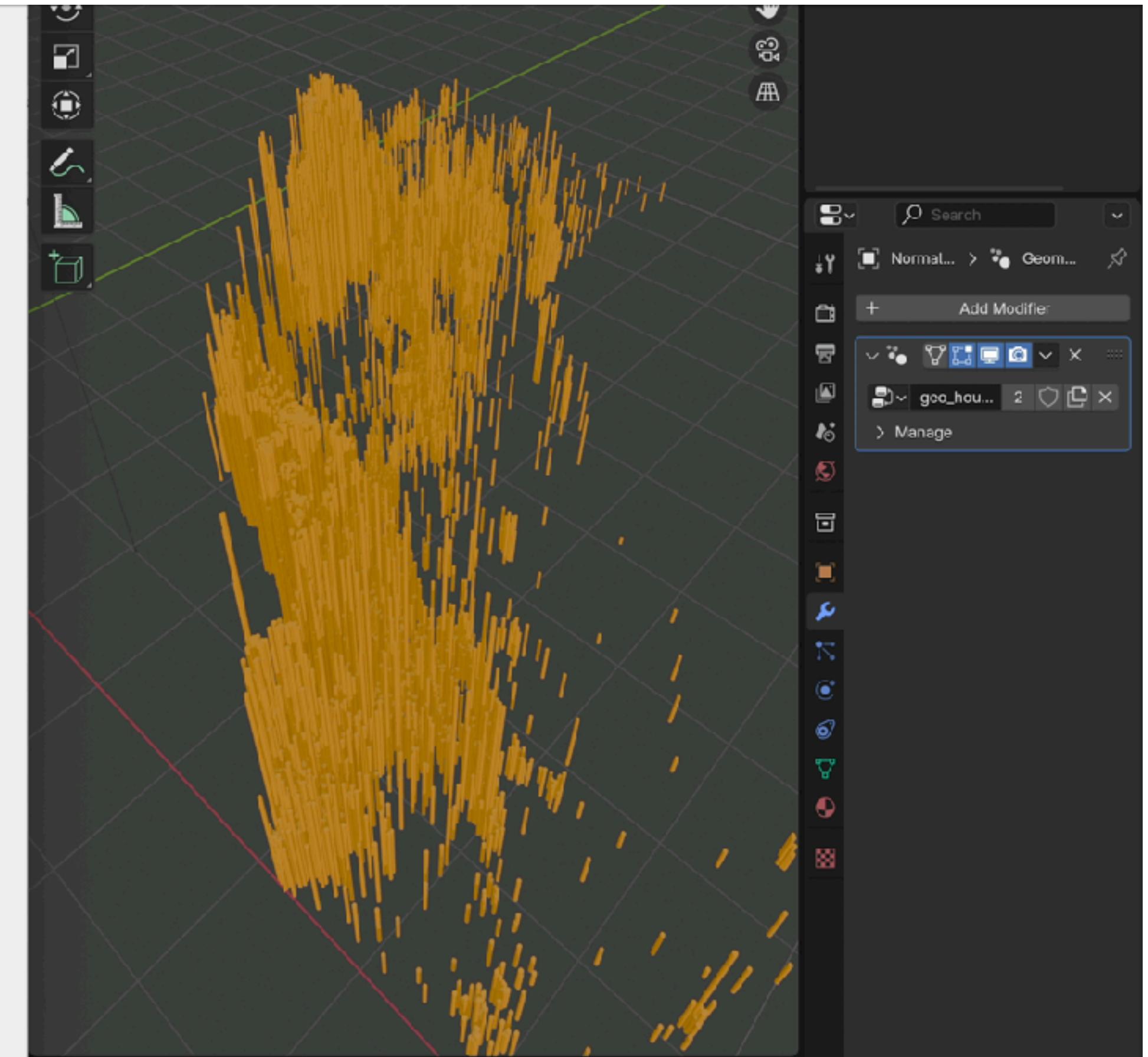
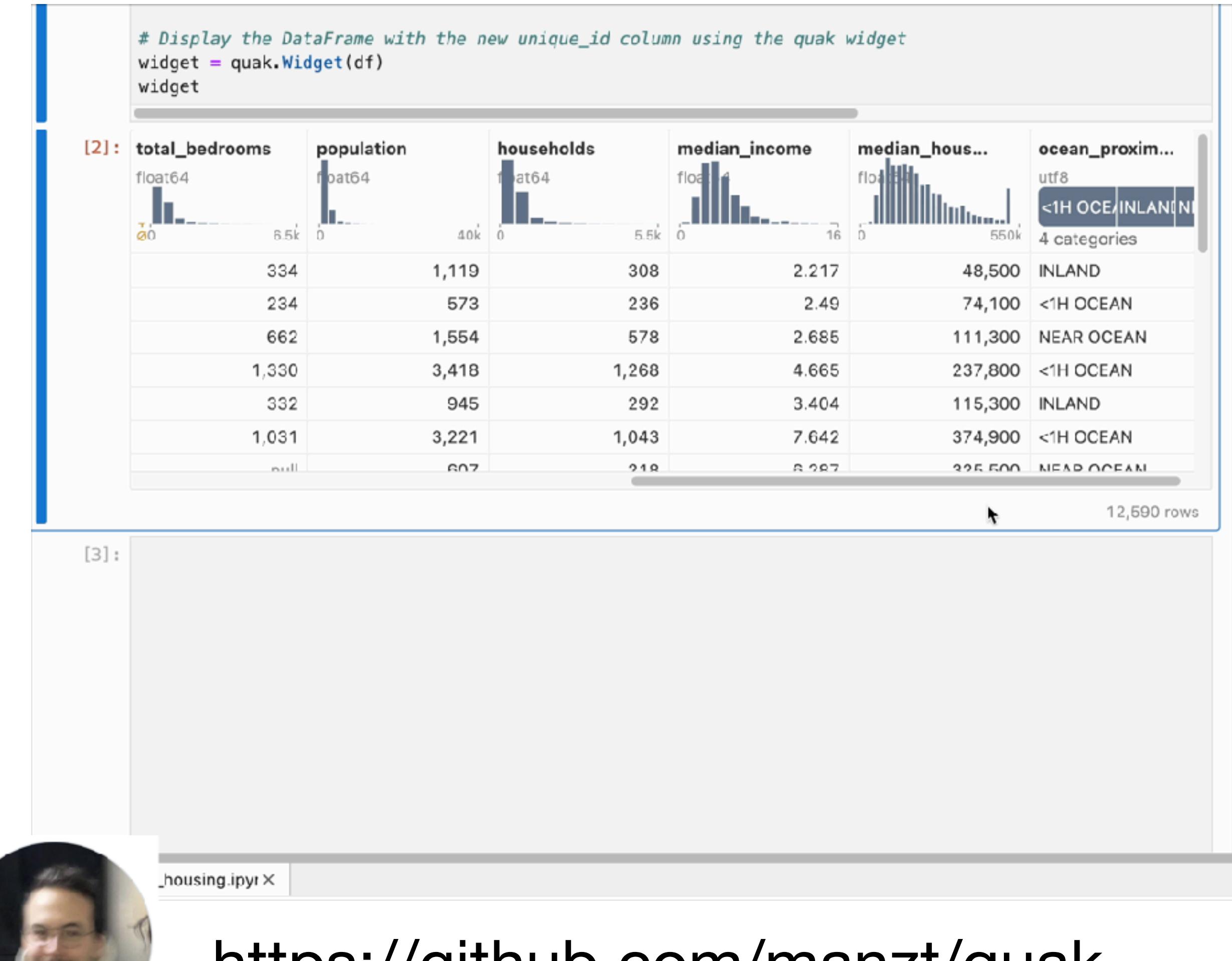
[3]: import bpy
import ipywidgets as widgets
from IPython.display import display

# Function to scale down the coordinates
def scale_down_points(points, scale_factor=0.01)
    scaled_points = [(x * scale_factor, y * scale_factor, z * scale_factor) for x, y, z in points]
    return scaled_points

# Function to create the curve
def create_curve_from_points():
    # Create a new curve object
    curve_data = bpy.data.curves.new(name='Curve')
    curve_data.dimensions = '3D'
    curve_data.fill_mode = 'FULL'
```

Blender & interaktive Tabellen

js → 



Trevor Manz
manzt · hehim

<https://github.com/manzt/quak>

Installation

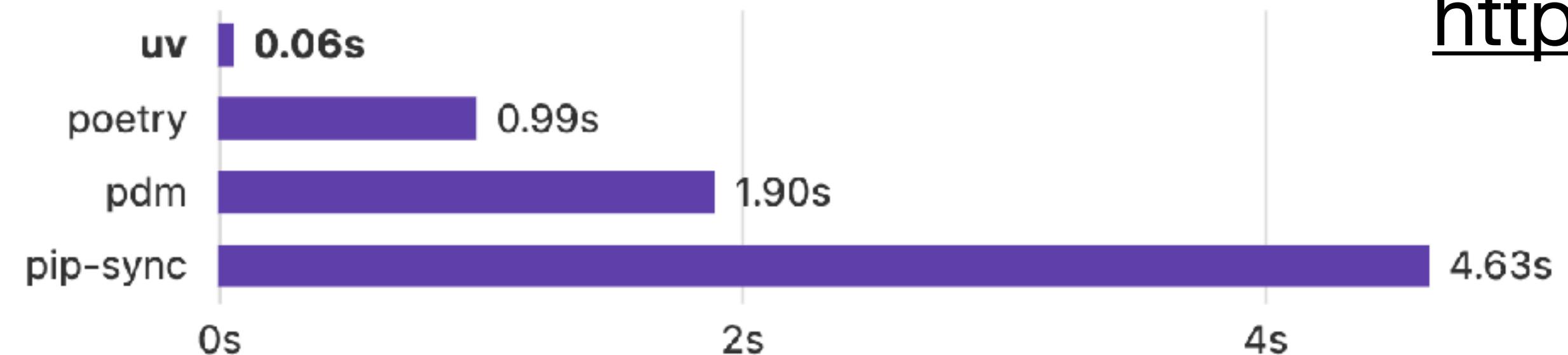


Full instructions at

https://kolibril13.github.io/bpy-gallery/n0getting_started/

Package manager UV

An extremely fast Python package and project manager, written in Rust.



<https://docs.astral.sh/uv/>

Install with



```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Installation mit uv

blender notebook setup

[blender_notebook.txt](#)

```
1 # Install Blender from https://www.blender.org/download/
2 # Install uv
3 curl -LsSf https://astral.sh/uv/install.sh | sh
4 mkdir blender_python_workshop && cd blender_python_workshop
5 uv venv --python 3.11
6 uv pip install jupyterlab blender_notebook
7 uv run blender_notebook install --blender-exec="/Applications/Blender.app/Contents/MacOS/Blender"
8 uv run jupyter lab
```



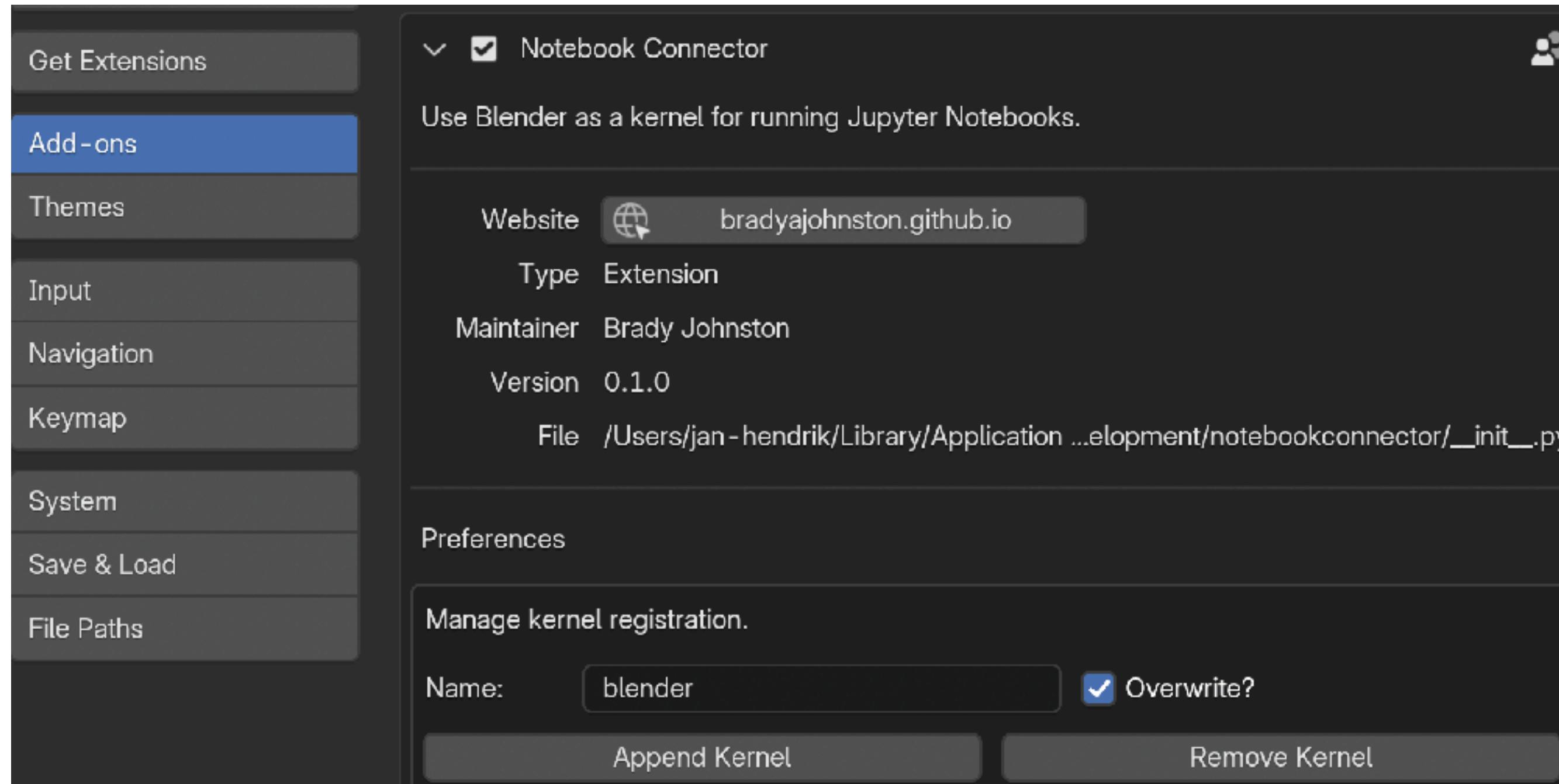
Write

Preview

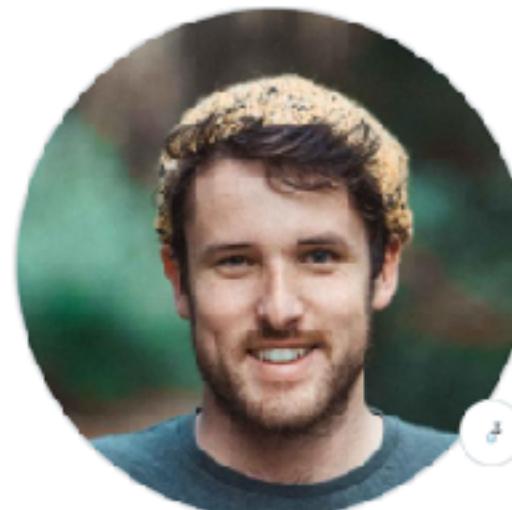
H B I ≡ <> ⌂ ⌃ ⌄

Leave a comment

In Zukunft: Notebook von Blender aus starten



Developer:



Brady Johnston
BradyAJohnston · he/him



Jan-Hendrik Müller

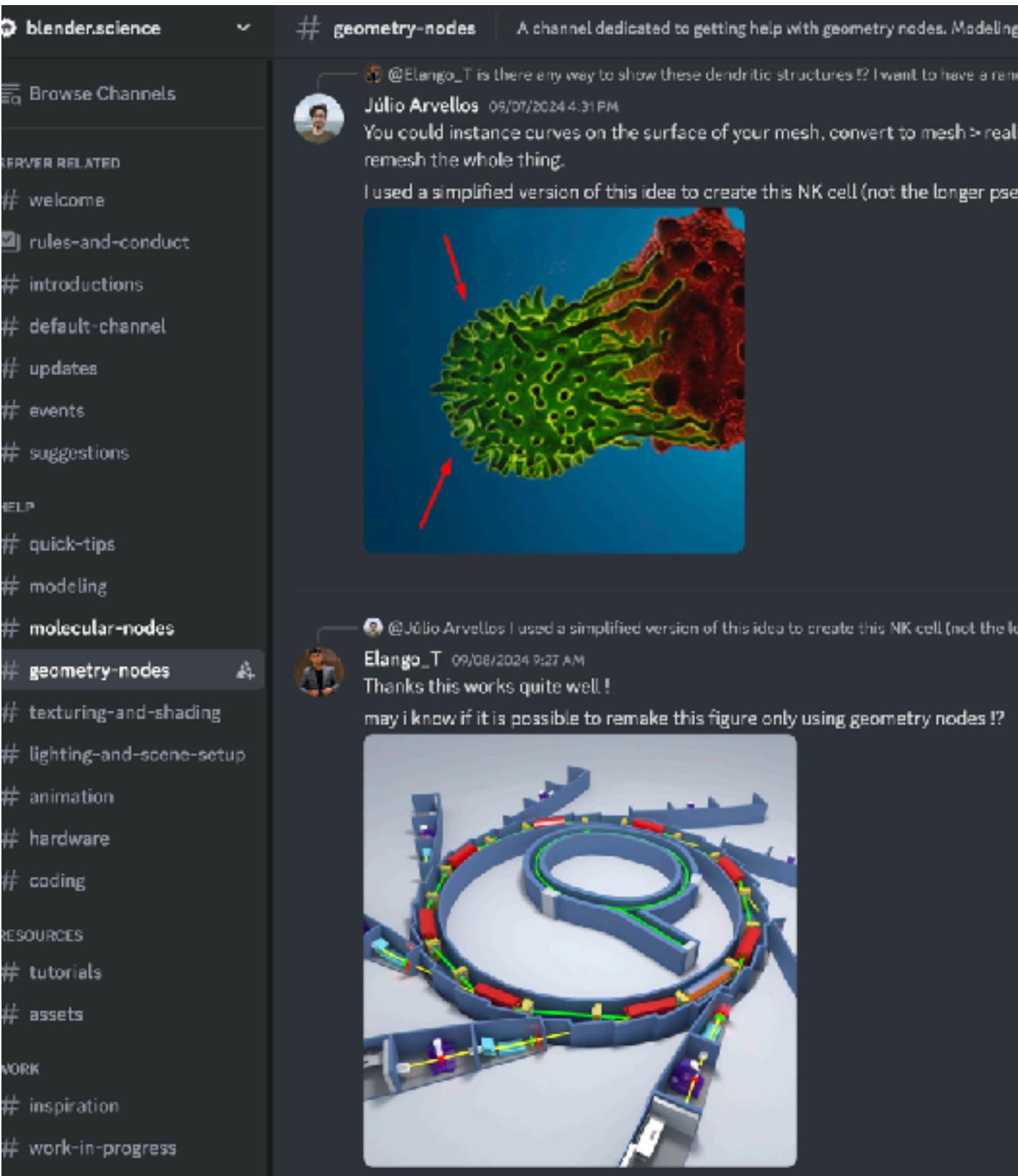


Sofus Albert Høgsbro Rose

Austausch Blender+Python

Science Blender Discord:

<https://discord.gg/k2Gd2cb4Kk>



Kontakt:

<https://twitter.com/kolibril13>

<https://www.linkedin.com/in/jan-hendrik-müller-765014209/>

<https://bsky.app/profile/kolibril13.bsky.social>

Als nächstes: Hands-On Session

<https://github.com/Octoframes/bpy-workshop/blob/main/task1.ipynb>

String Data

