



# Data Visualization

# Agenda

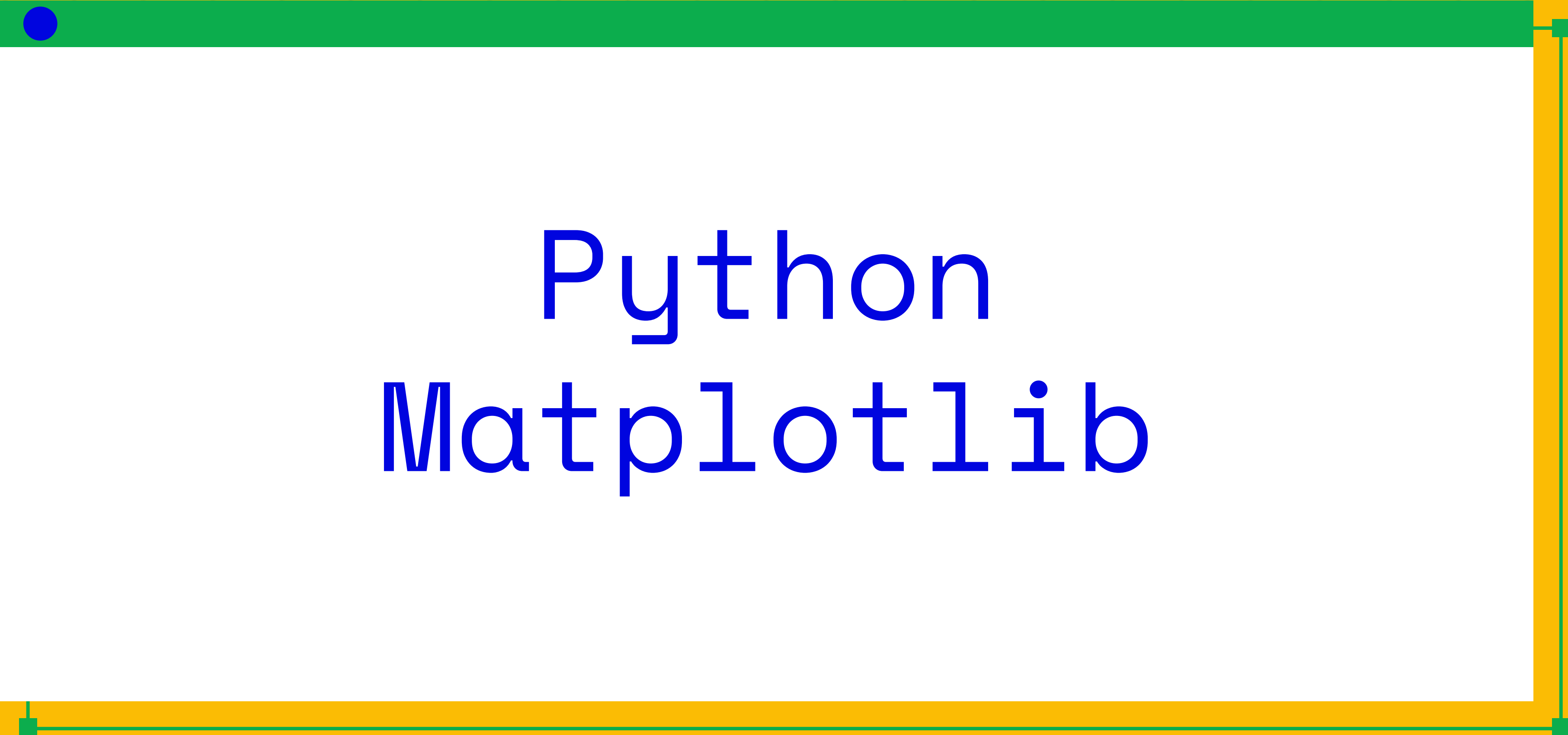
1. Google Sheets Charts
2. Python Matplotlib
3. Questions



# Google Sheets Charts

# All Google Sheets Materials are on GitHub

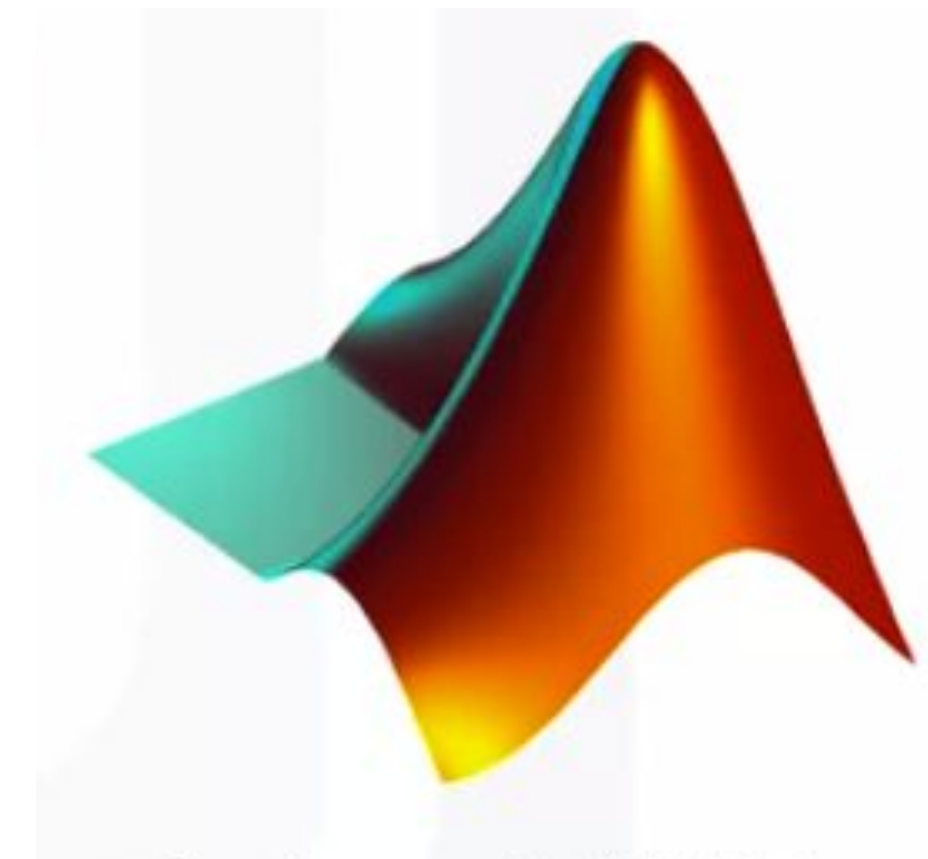
1. Source Excel Files
2. Steps in PDF Files
3. Solution Links



# Python Matplotlib

# John Hunter (Matplotlib Creator)

- Neurobiologist
- Part of a team analyzing **Electrocorticography Signals (ECoG)**
  - **Electrocorticography** is the process of recording electrical activity in the brain
- The team
  - used a proprietary software (**MATLAB** based version) for analysis
  - had only one license and were taking turns in using it
- **John** replace the proprietary software with **Matplotlib**

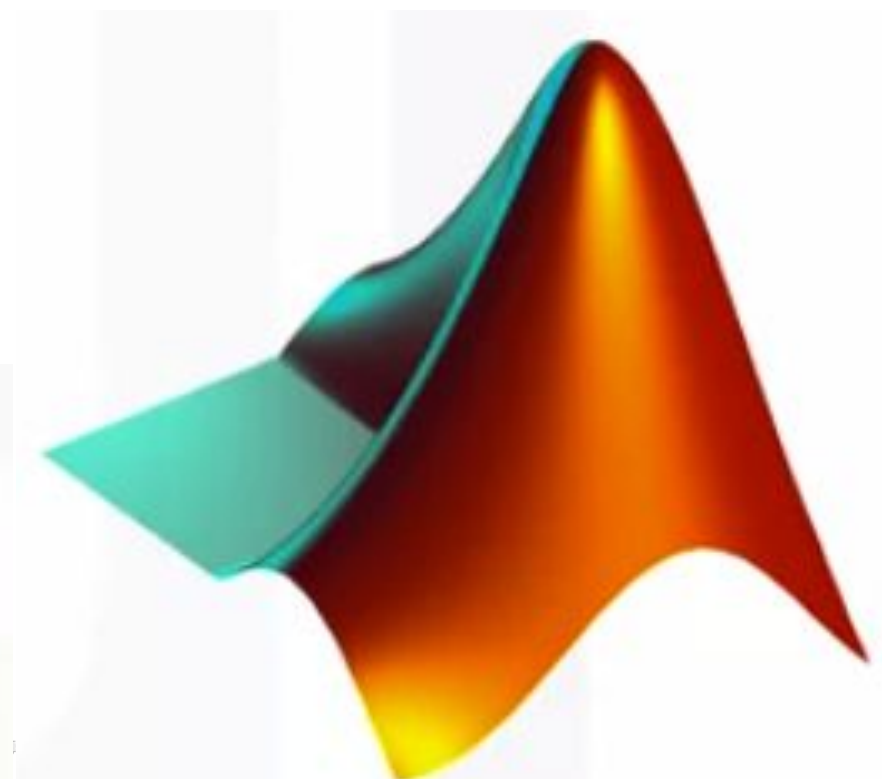


# Python Matplotlib

- MatLab-style Plotting Library (Created in 2002)
- Originally developed as an **ECoG** visualization tool
- Most popular data visualization library in Python
- Well supported in different environments
  - Python scripts
  - web app servers
  - iPython (Interactive shell)
  - **Jupyter Notebook**



EEG/ECOG Visualization Tool



Analogous to Matlab  
scripting interface

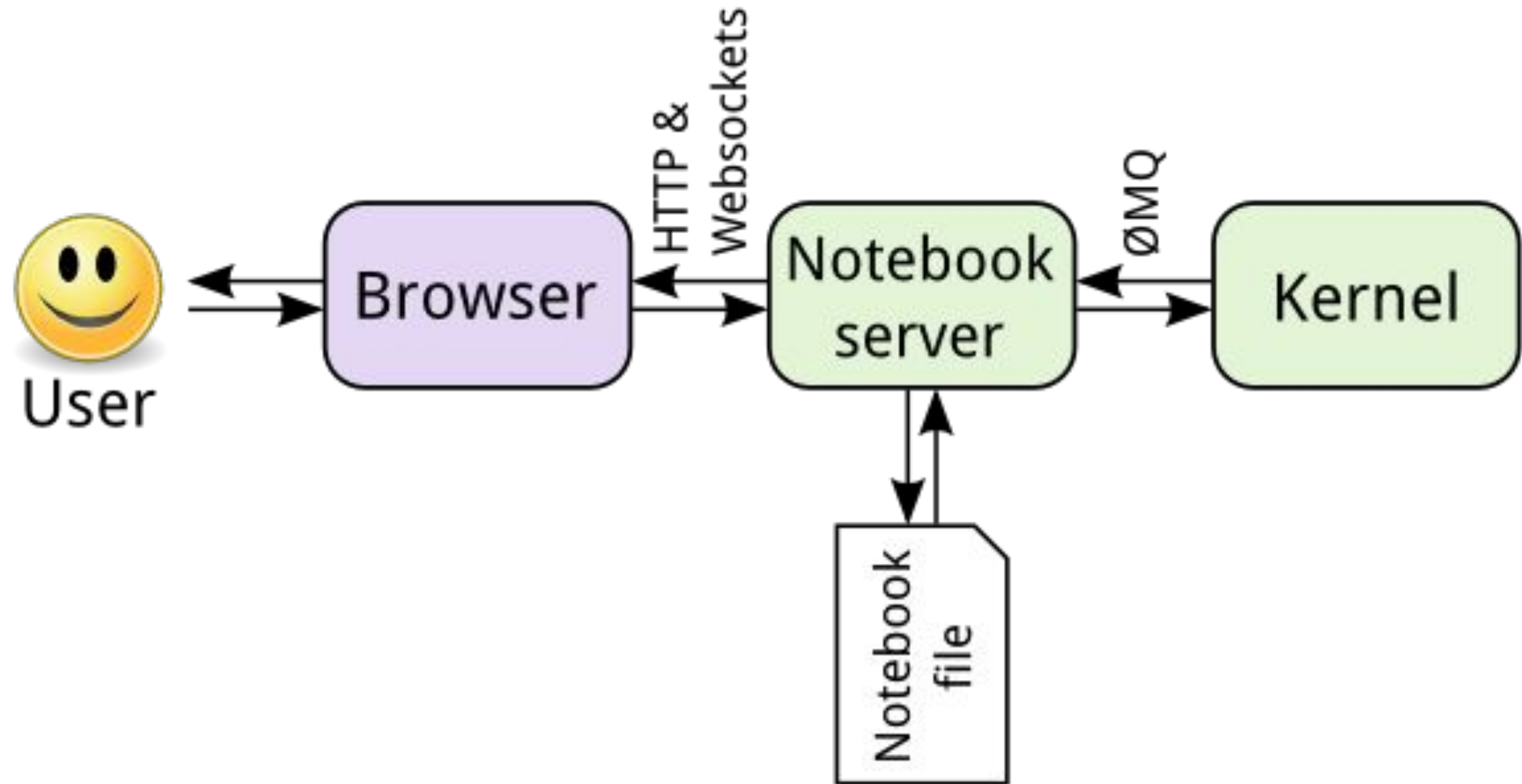
# Jupyter Notebook

- open source web app
- allows to create & share documents that contain code and text
- spun off from **iPython** in **2014**
- **Jupyter** name is a reference to three programming languages:
  - **Julia**
  - **Python**
  - **R**
- **Jupyter** logo
  - homage to **Galileo**'s discovery of the **moons of Jupiter**
  - documented in **notebooks** attributed to **Galileo**





# Jupyter Notebook Workflow



[https://ipython.org/ipython-doc/3/development/how\\_ipython\\_works.html](https://ipython.org/ipython-doc/3/development/how_ipython_works.html)

# Matplotlib Architecture

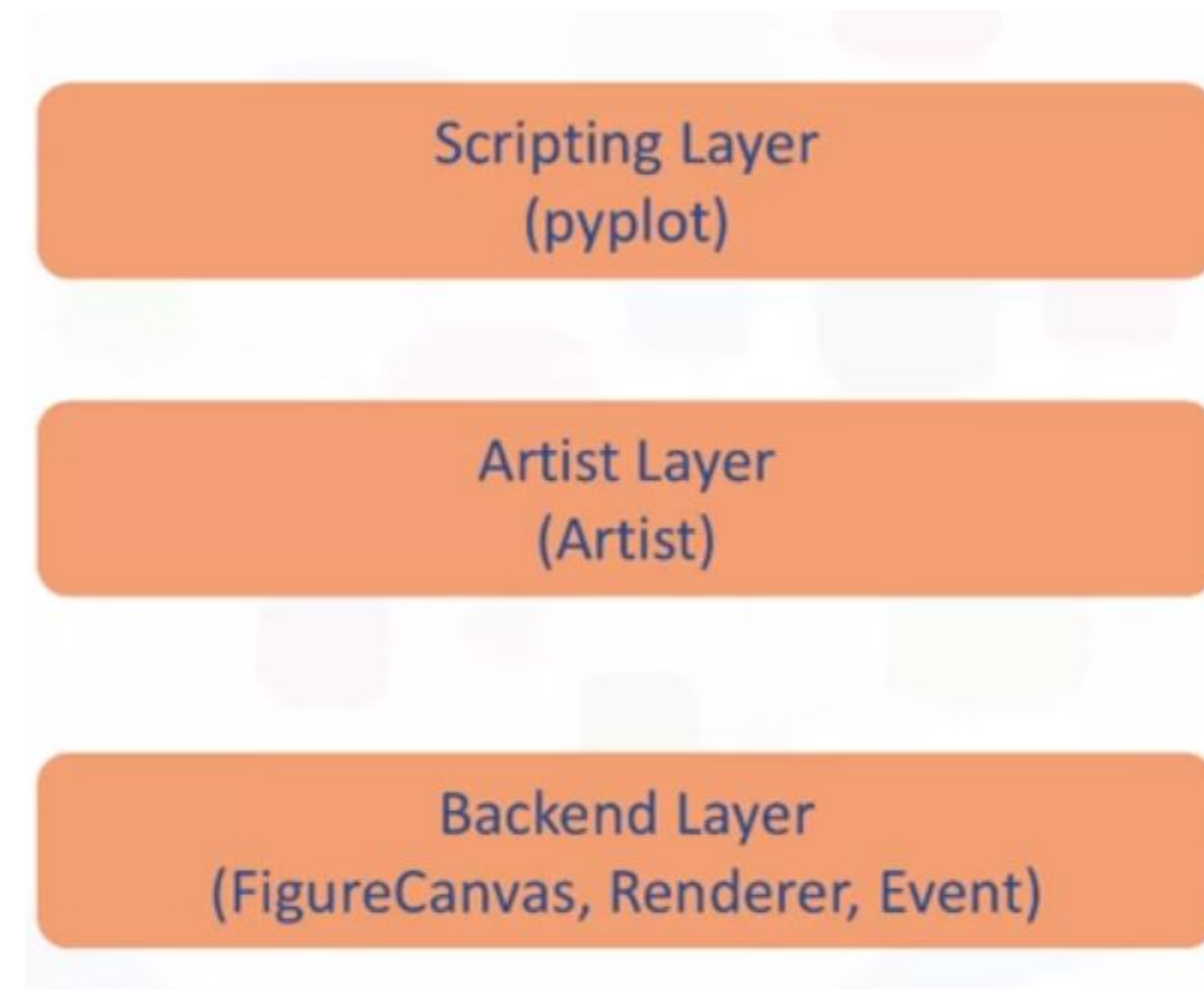
## 1. Back-end Layer

## 2. Artist Layer

- appropriate programming paradigm for
  - web app server
  - UI app
  - script to be shared with others

## 3. Scripting Layer (idea from MATLAB)

- appropriate layer for everyday purposes
- lighter interface to simplify common tasks
- for a quick and easy generation of plots



# Matplotlib Architecture: 1) Back-end Layer

has built-in classes, such as:

1. **FigureCanvas**: *matplotlib.backend\_bases.FigureCanvas***Base**

- defines and encompasses the area into which the figure is drawn

2. **Renderer**: *matplotlib.backend\_bases.Renderer***Base**

- knows how to draw (generate image) on the **FigureCanvas**

3. **Event**: *matplotlib.backend\_bases.Event*

- handles user inputs such as keyboard strokes and mouse clicks

- [https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/backend\\_bases.py](https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/backend_bases.py)

- [https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/backends/backend\\_agg.py](https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/backends/backend_agg.py)

# Matplotlib Architecture: 2) Artist Layer

- Contains one main abstract class (the **Artist**)
- **Artist**
  - knows how to use the **Renderer** to draw (put ink) on the **FigureCanvas**
  - <https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/artist.pyi>
- Everything you see on a **Matplotlib figure** is an **Artist instance**
  - **Example:** title, lines, tick labels, images, ...
  - all of them correspond to an individual **Artist instance**



# Matplotlib Architecture: 2) Artist Layer Types

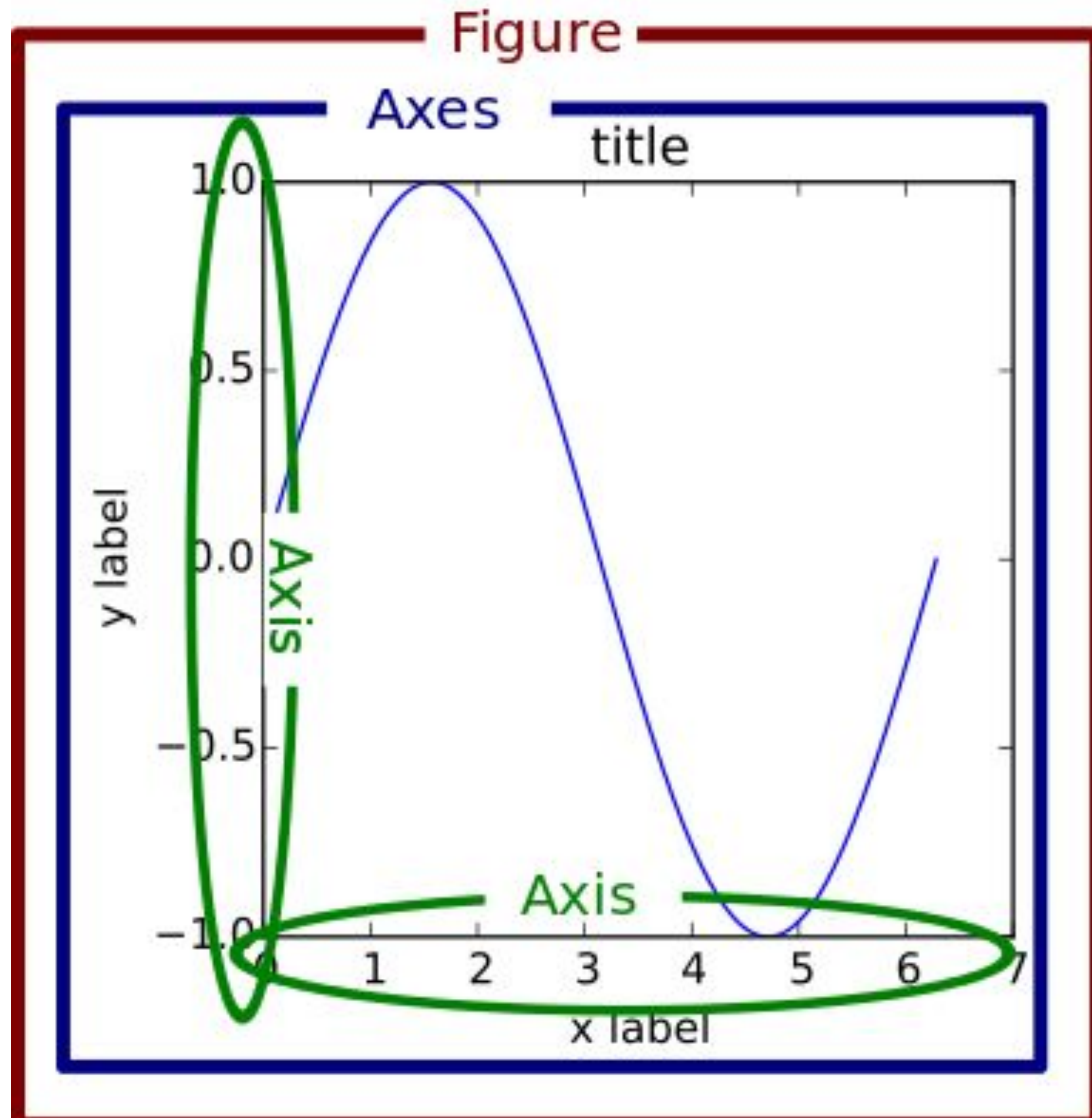
1. **Primitive Artist:** as Line, Rectangle, Circle, Text

2. **Composite Artist:** may contain other **Artists**

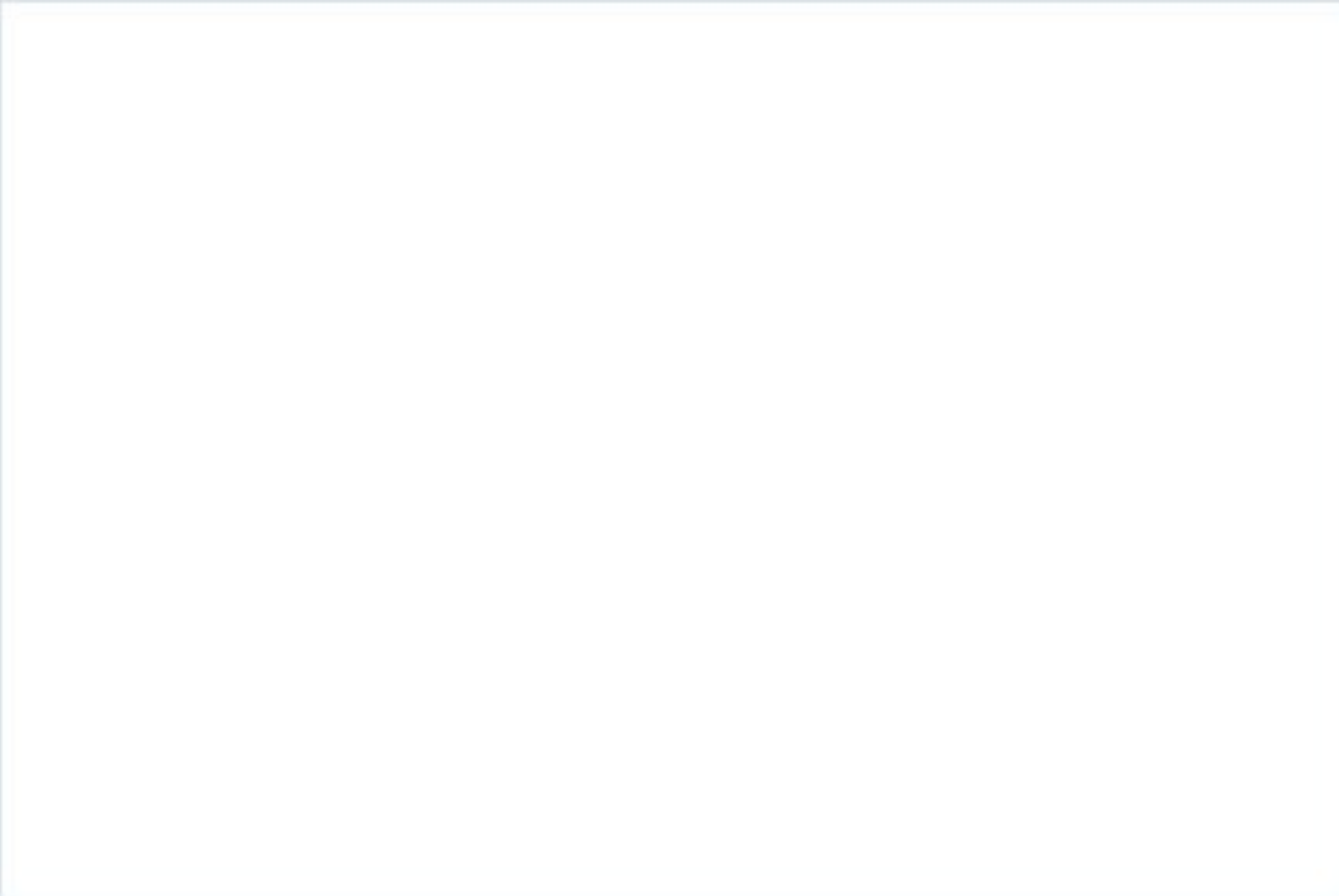
- **Example 1: Figure Artist** <https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/figure.py>
  - top-level Matplotlib object
  - contains and manages all of the elements in a given graphic
- **Example 2: Axes Artist** <https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/axes/axes.py>
  - most important Composite Artist
  - where most of the plotting methods are defined
    - including methods to create/manipulate ticks, axis lines, grid, background
- **Other Examples: Tick Artist**

# Axes

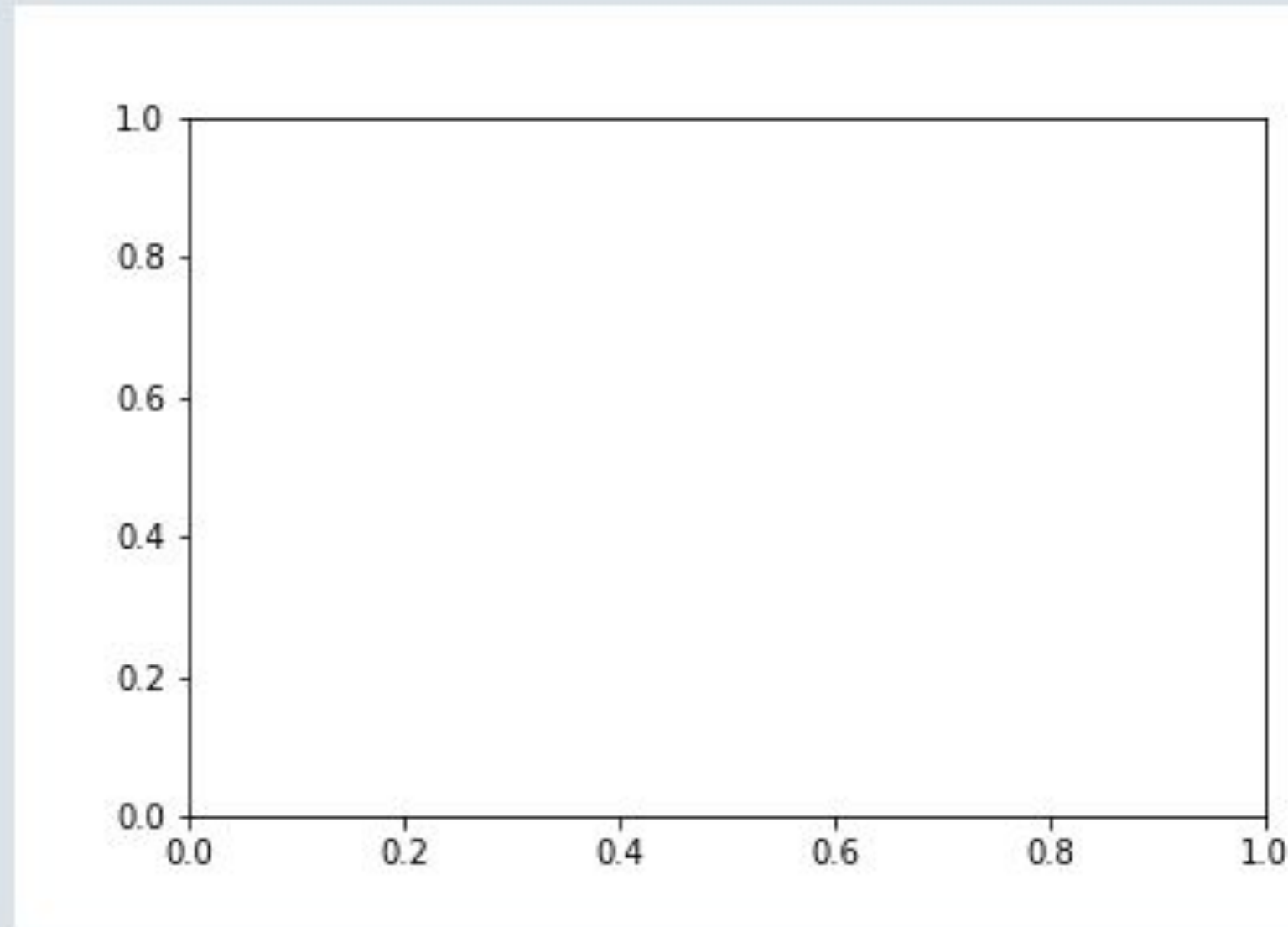
- The plotting area
  - including all axis
  - don't mean plural of **Axis**
- When pronounced with short e
  - axes is the plural of axe
- When pronounced with long e
  - axes is the plural of axis



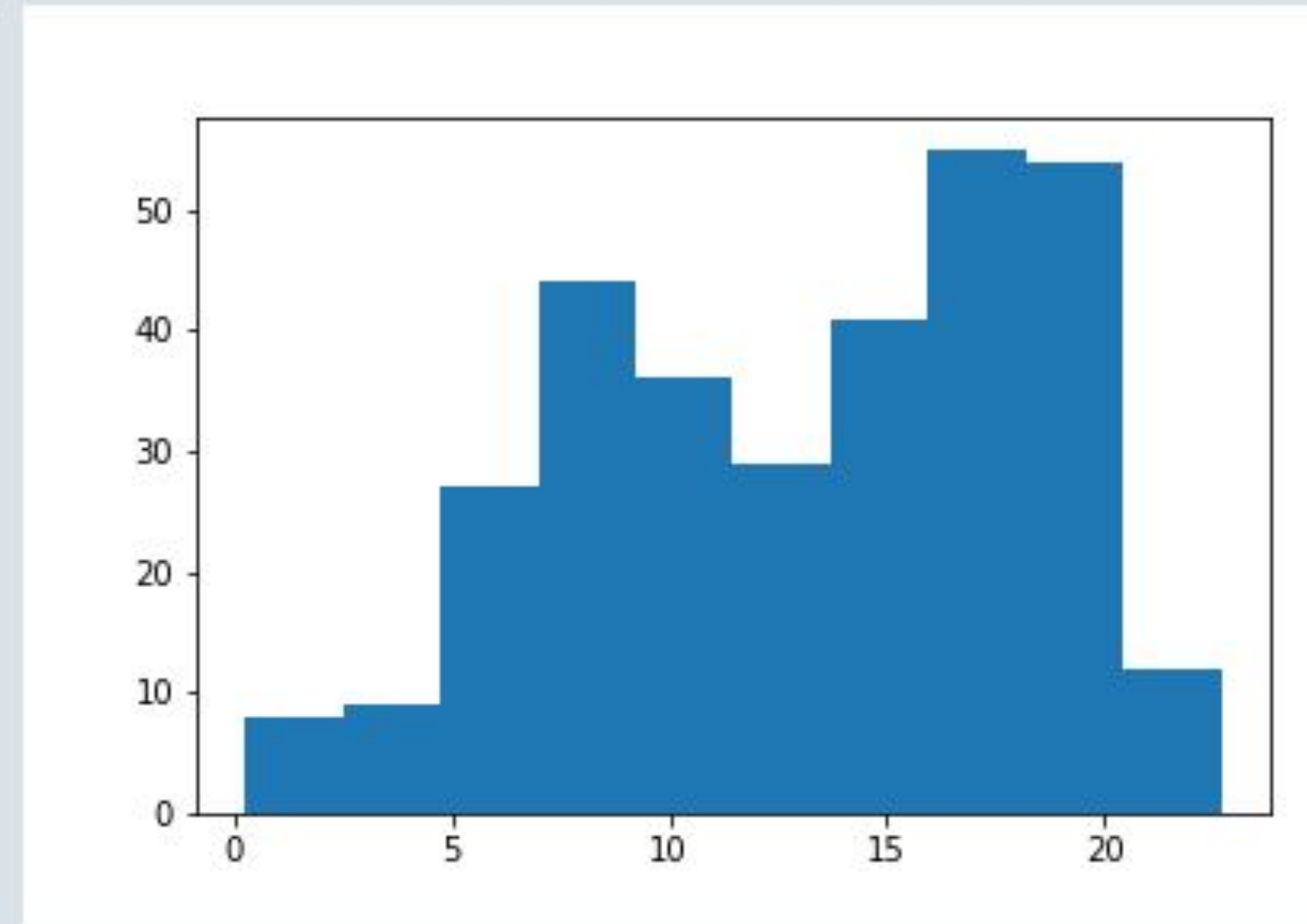
# Figure and Axes



Create a Figure,



add an Axes to the Figure...




then plot the data inside that Axes!

# Matplotlib Architecture: 3) Scripting Layer

- Developed for scientists who are not professional programmers
- Essentially the `Matplotlib.pyplot` that automates:
  - defining **FigureCanvas**
  - defining **Artist**
  - connecting **Artist** with **FigureCanvas**
  - <https://github.dev/matplotlib/matplotlib/blob/main/lib/matplotlib/pyplot.py>
- Comparing with **Layer 2 (Artist Layer)** which is:
  - heavy and for developers
  - not for individuals who want to perform **quick EDA** of some data





# Questions

# Links

<https://github.com/fcai-b/dv>

# References

1. <https://www.coursera.org/learn/foundations-data>
2. <https://www.coursera.org/learn/what-is-datascience>
3. <https://www.coursera.org/learn/python-for-data-visualization>
4. <https://www.coursera.org/learn/google-sheets---advanced-topics>