Creating Dashboards with Python

More specifically – plotly & Dash

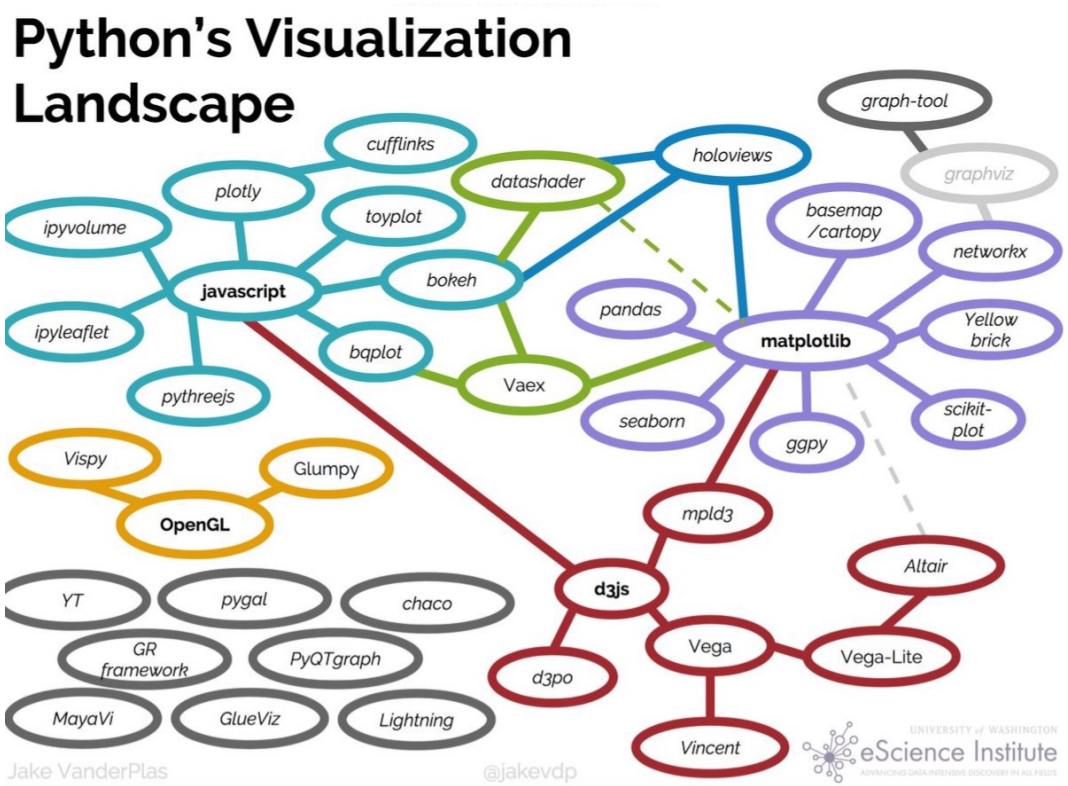
Summer 2022

Housekeeping

- In case of technical problems:
  - Something wrong on my end (e.g. power outage), I will send you an email.
  - Something wrong on your end, please send me a text message. 508-769-6446
  - jcodygroup@gmail.com
- We have 4 hours for each session
  - I will try to give you an opportunity to stand and stretch every hour.
  - We will take at least one 15-minute break near the halfway point.

About me

- **Experience:**
  - 25+ years consulting and training experience
  - Extensive work with “big data” and analytics
  - 15 years working with various data visualization tools
- **Education**
  - Ed. M., Technology, Innovation & Education, Harvard University
  - PhD Candidate, Education Policy, University of Massachusetts, Amherst



matplotlib
Version 3.4.3

seaborn: statistical data visualization

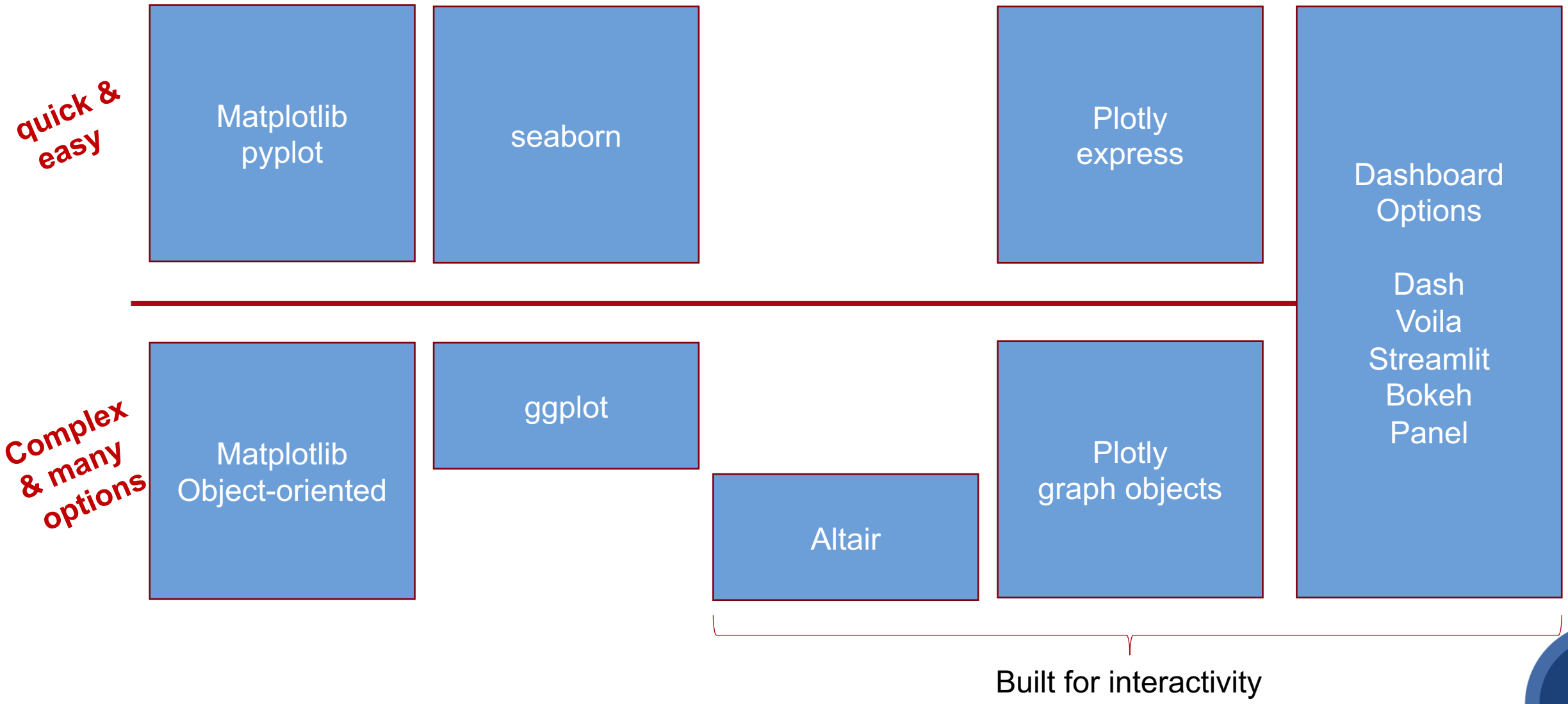
The Bokeh Visualization Library

plotnine 0.8.0 API Gallery Tutorials Site Page

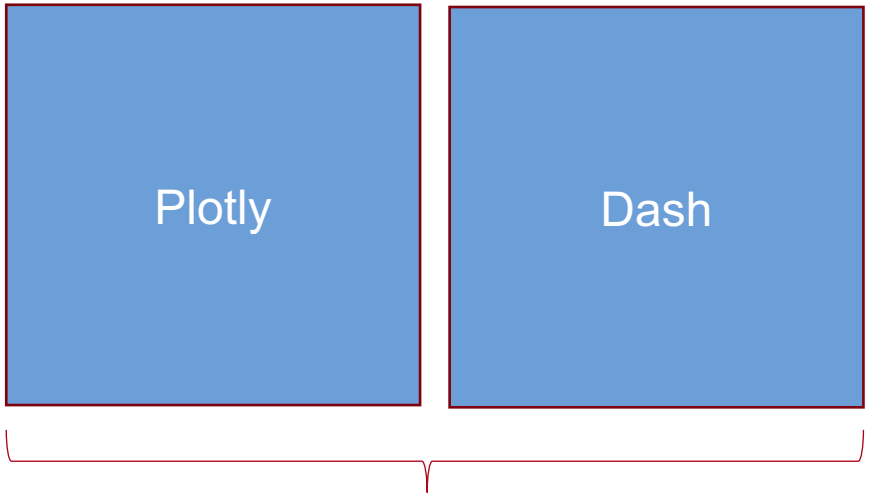
A Grammar of Graphics for Python

Altair: Declarative Visualization in Python

Visualization packages



Our tasks



Python
(notebooks
ipywidgets)

git
heroku

and through
kaggle

Our tasks

Dash Introduction

- Terminology
- Structure
- CoLab

Plotly express

- px.line
- px.scatter
- px.bar
- facets

Dash exploration

- Layout
- Add a plot
- Change plot
- Html components
- Component args.
- Div()
- Positioning
- Repositioning
- Markdown text
- Interactivity comps
- Callback
- Reusable comp
- Simple callback
- Changing variables

Plotly graph objects

- Figures
- Layout
- Traces
- Data
- Update layout
- Hover text

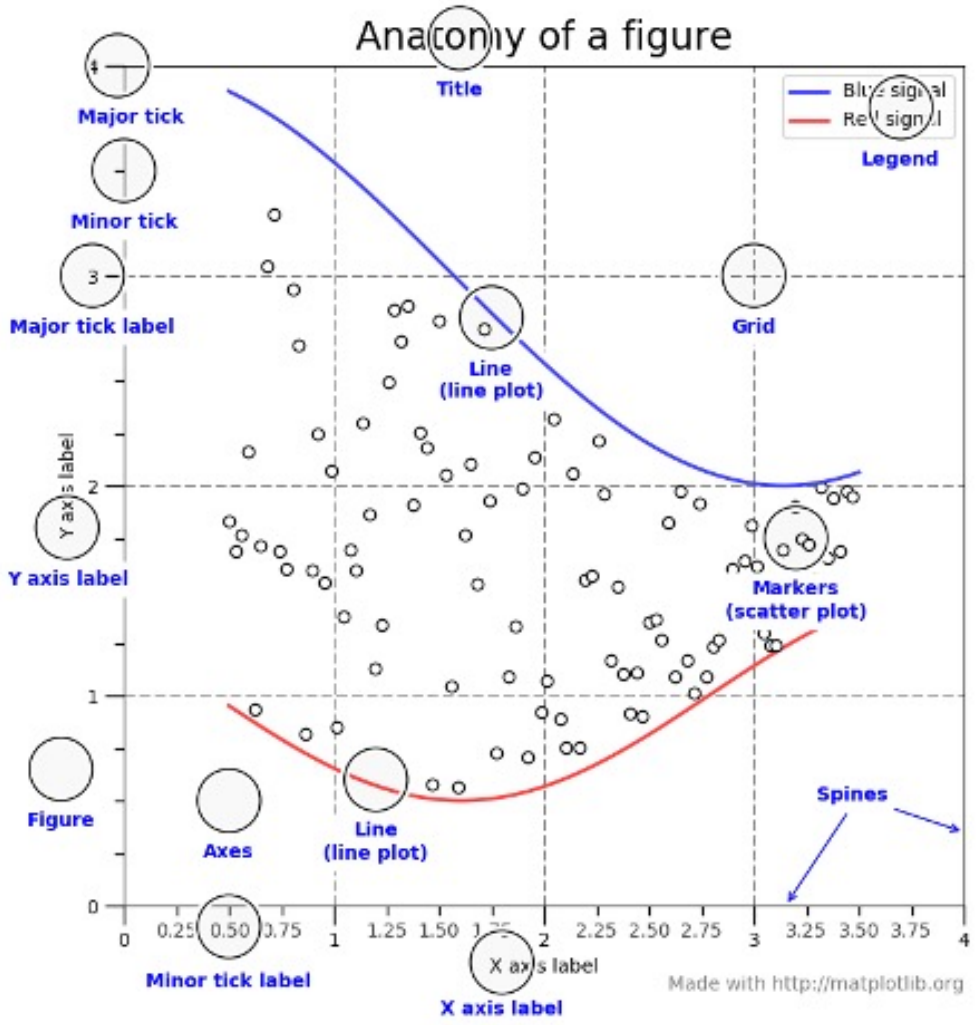
Dash again

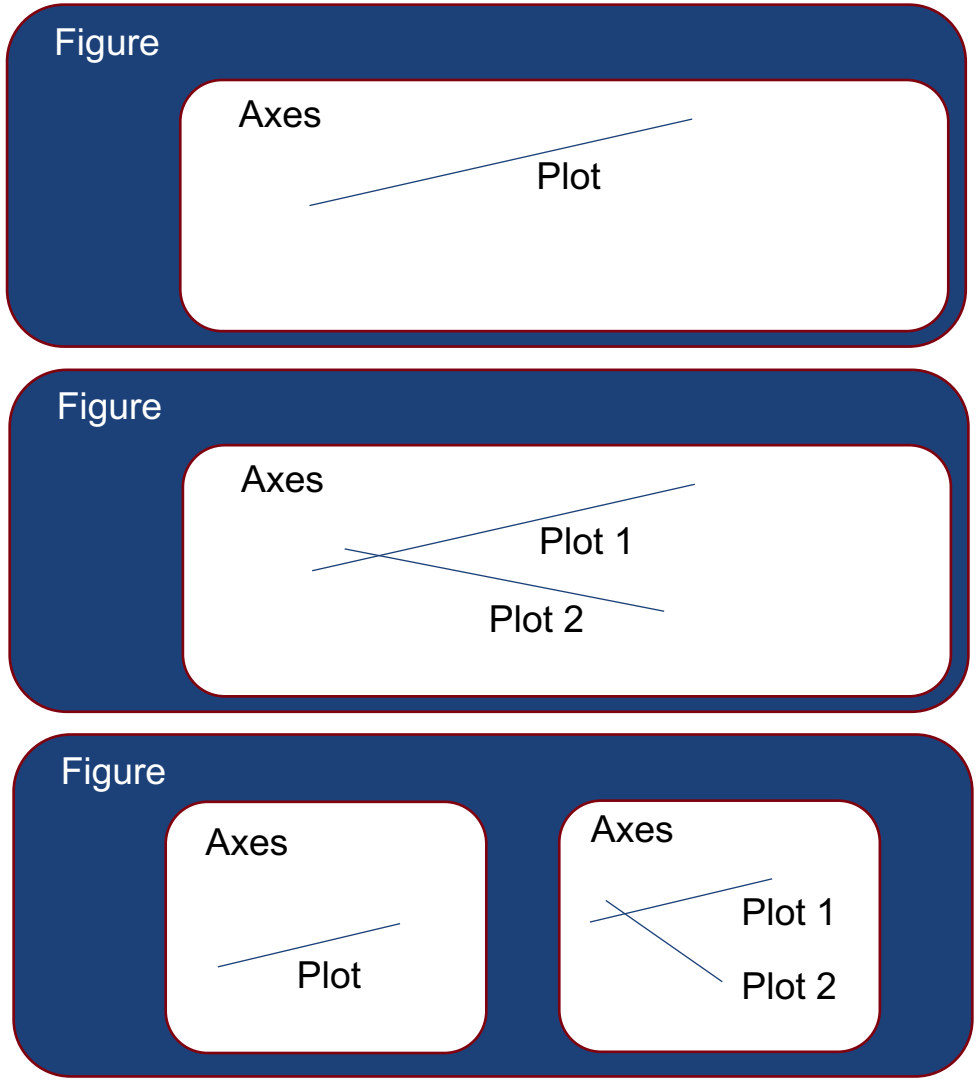
- Terminology
- Structure

Deploy to Web

git & heroku

Matplotlib/Seaborn: Two big concepts to keep in mind

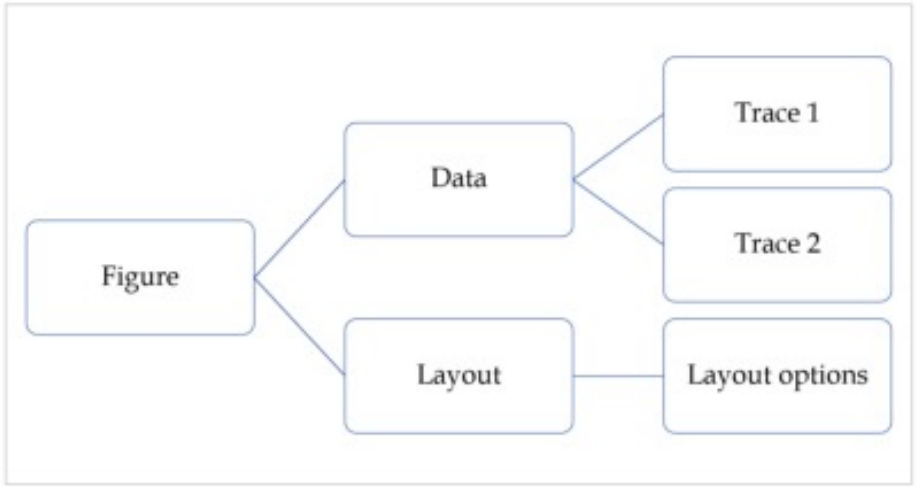




plotly graph objects

The **plotly.graph_objs module** is the most important module that contains all of the class definitions for the objects that make up the plots you see. Following graph objects are defined:

- Figure,
- Data,
- Layout,
- Different graph traces like **Scatter**, **Box**, **Histogram** etc.



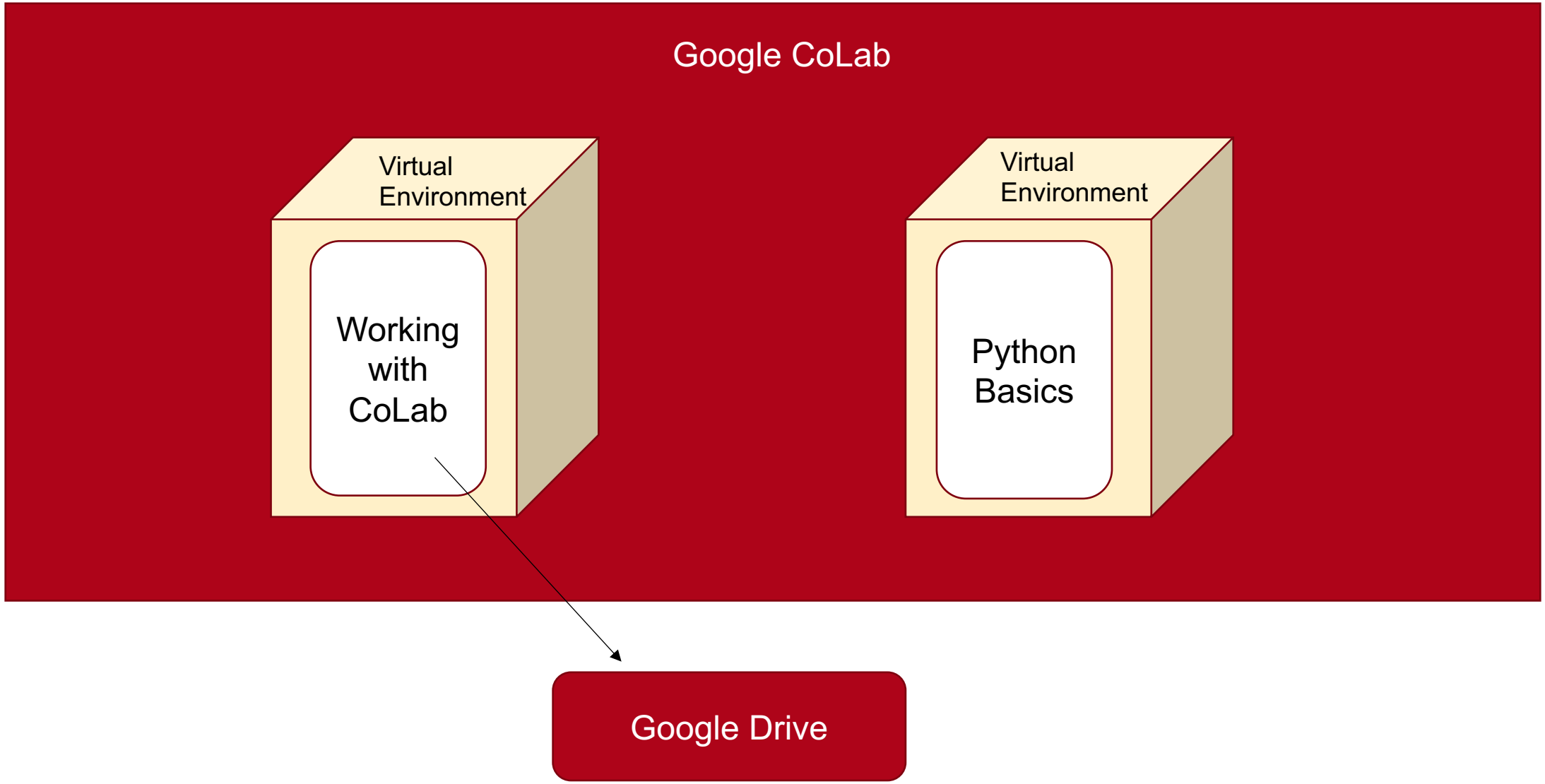
All graph objects are dictionary- and list-like objects used to generate and/or modify every feature of a Plotly plot.

Dashboard Structure

```
# The general structure of a dashboard application:

imports .....

app = JupyterDash(__name__)      # This is the start of the application

get the data....

create a figure(plot)...

app.layout =                     # Describe what the page will look like

    layout code

    dcc.Graph()                  # What plot will be included

@app.callback(
    what are the inputs?
    what are the outputs?

    resusable component )        # This processes the input and creates the output

app.run_server(mode='inline')    # .run_server() is the method to run the code


```

```
# An example of a callback from documentation
# Just changes the text that appears - no plotting

from jupyter_dash import JupyterDash
from dash.dependencies import Output, Input
from dash import dcc
from dash import html

app = JupyterDash(__name__)

app.layout = html.Div([
    html.H6("Change the value in the text box to see callbacks in action!"),
    html.Div([
        "Input: ",
        dcc.Input(id='my-input', value='initial value', type='text')
    ]),
    html.Br(),
    html.Div(id='my-output'),

])


@app.callback(
    Output(component_id='my-output', component_property='children'),
    Input(component_id='my-input', component_property='value')
)
def update_output_div(input_value):
    return 'Output: {}'.format(input_value)

app.run_server(mode='inline')
#app.run_server(mode='external', port = 8071)
```

Day 1 Recap


Google CoLab
Virtual Environment
Working with CoLab
Virtual Environment
Python Basics
Google Drive