Pokaż mi co masz!

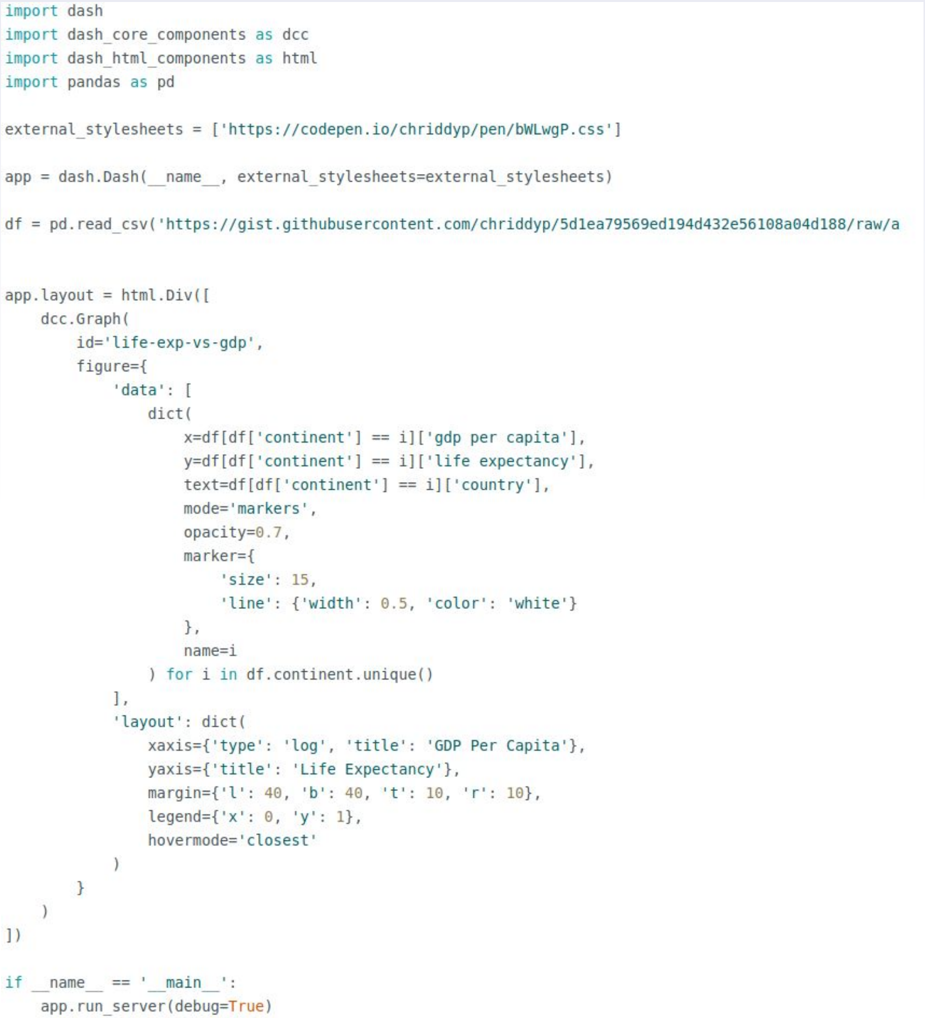
Wstęp do wizualizacji danych z

plotly | Dash

Wojciech Łaguna

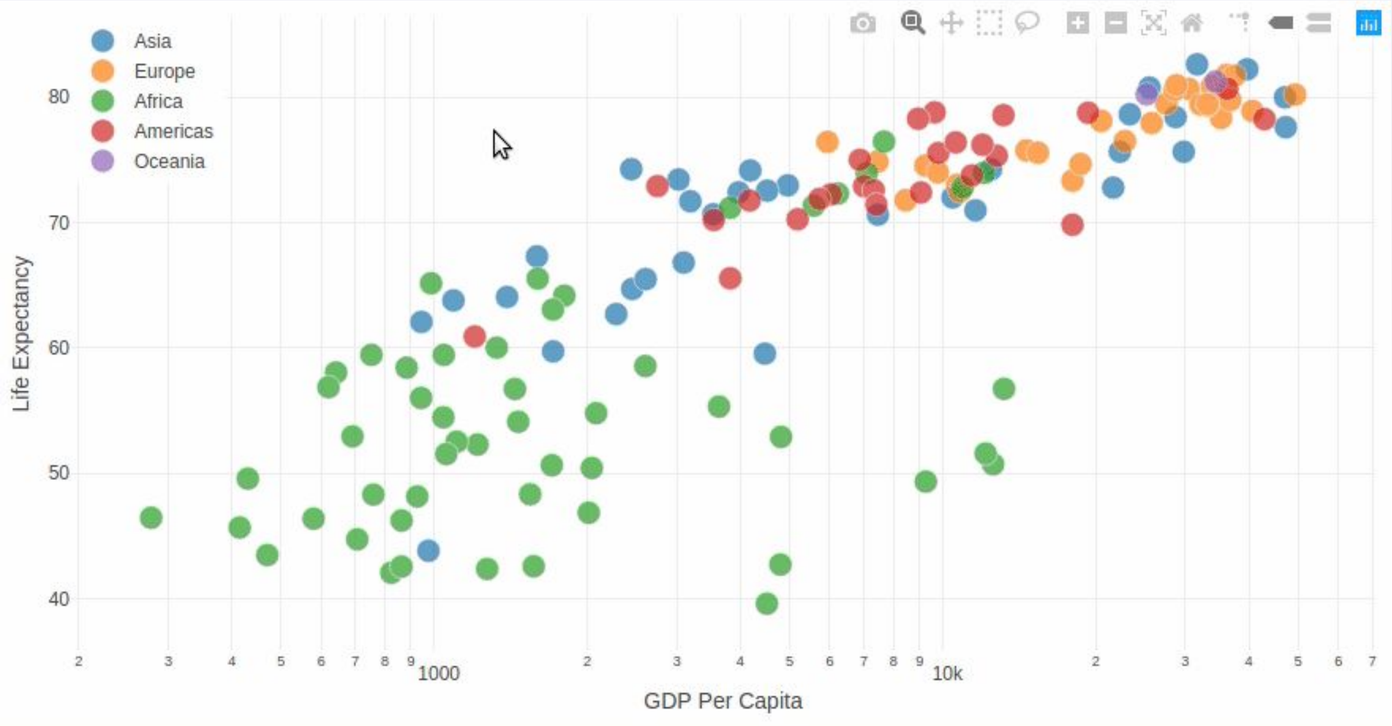
O czym my mówimy?

- Kto ma dane?
- Kto rozumie dane?
- Kto interpretuje dane?
- Czy p-value wystarczy?



```
import dash
import dash_core_components as dcc
import dash_html_components as html
import pandas as pd

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__, external_stylesheets=external_stylesheets)

df = pd.read_csv('https://gist.githubusercontent.com/chriddyp/5d1ea79569ed194d432e56108a04d188/raw/a


app.layout = html.Div([
    dcc.Graph(
        id='life-exp-vs-gdp',
        figure={
            'data': [
                dict(
                    x=df[df['continent'] == i]['gdp per capita'],
                    y=df[df['continent'] == i]['life expectancy'],
                    text=df[df['continent'] == i]['country'],
                    mode='markers',
                    opacity=0.7,
                    marker={
                        'size': 15,
                        'line': {'width': 0.5, 'color': 'white'}
                    },
                    name=i
                ) for i in df.continent.unique()
            ],
            'layout': dict(
                xaxis={'type': 'log', 'title': 'GDP Per Capita'},
                yaxis={'title': 'Life Expectancy'},
                margin={'l': 40, 'b': 40, 't': 10, 'r': 10},
                legend={'x': 0, 'y': 1},
                hovermode='closest'
            )
        }
    )
])

if __name__ == '__main__':
    app.run_server(debug=True)
```

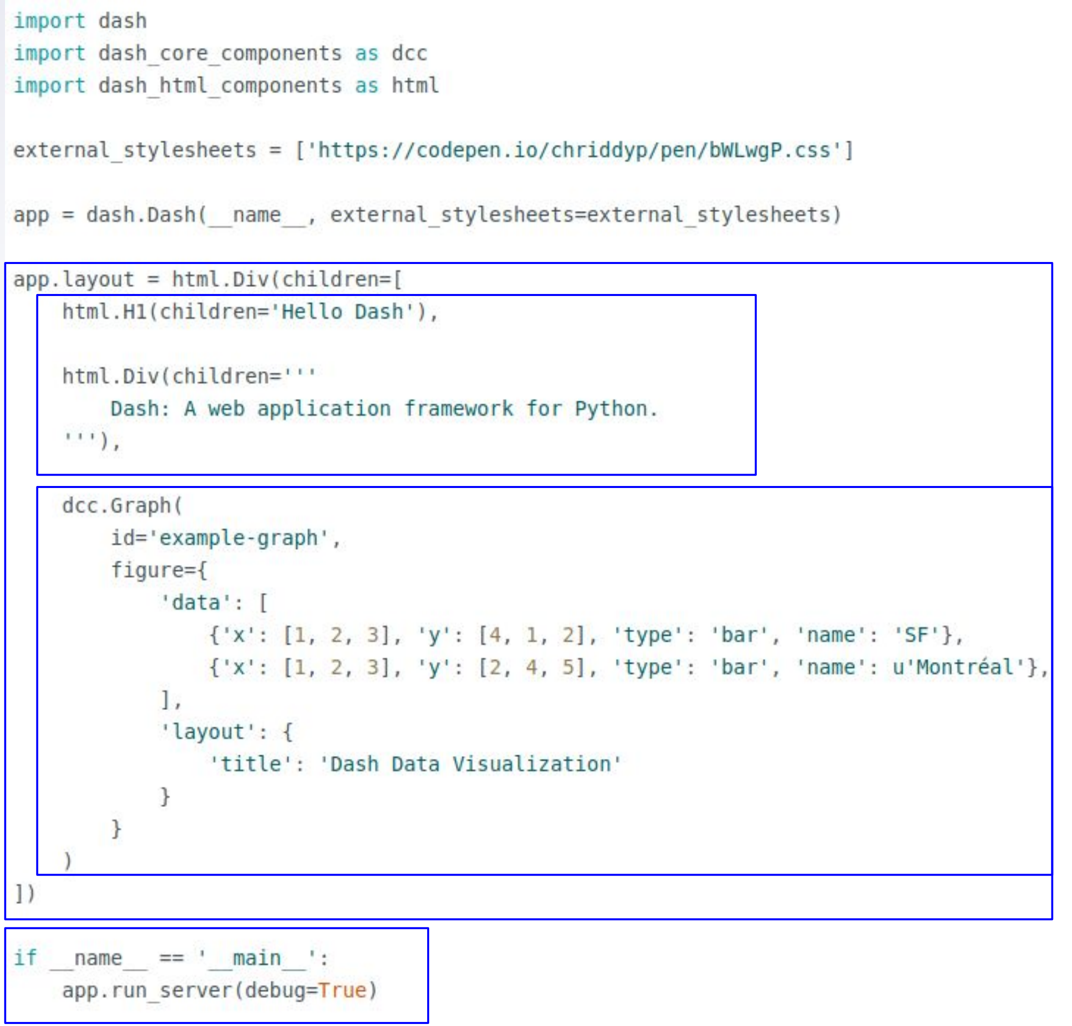
plotly | Dash


Asia
Europe
Africa
Americas
Oceania
Life Expectancy
80
70
60
50
40
2
3
4
5
6
7
8
9
1000
2
3
4
5
6
7
8
9
10k
2
3
4
5
6
7
GDP Per Capita

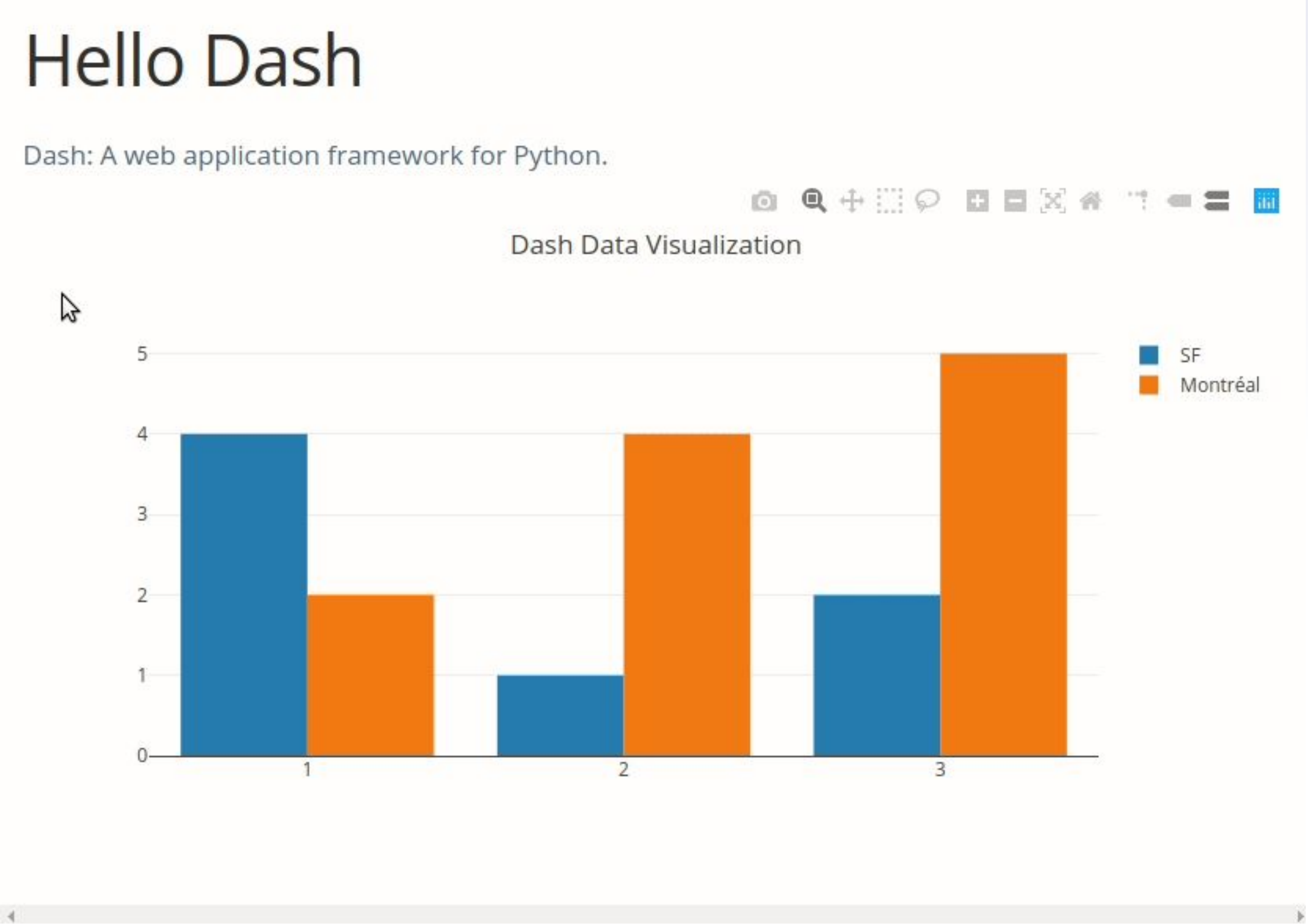
plotly | Dash

plotly | Dash

- Narzędzie do budowania aplikacji webowych
- Zbudowane z wykorzystaniem:
  - Flask
  - Plotly.js
  - React.js
- Wizualizacje bez znajomości JSa
- Responsywność
- Licencja MIT
- Bezproblemowy “deploy”



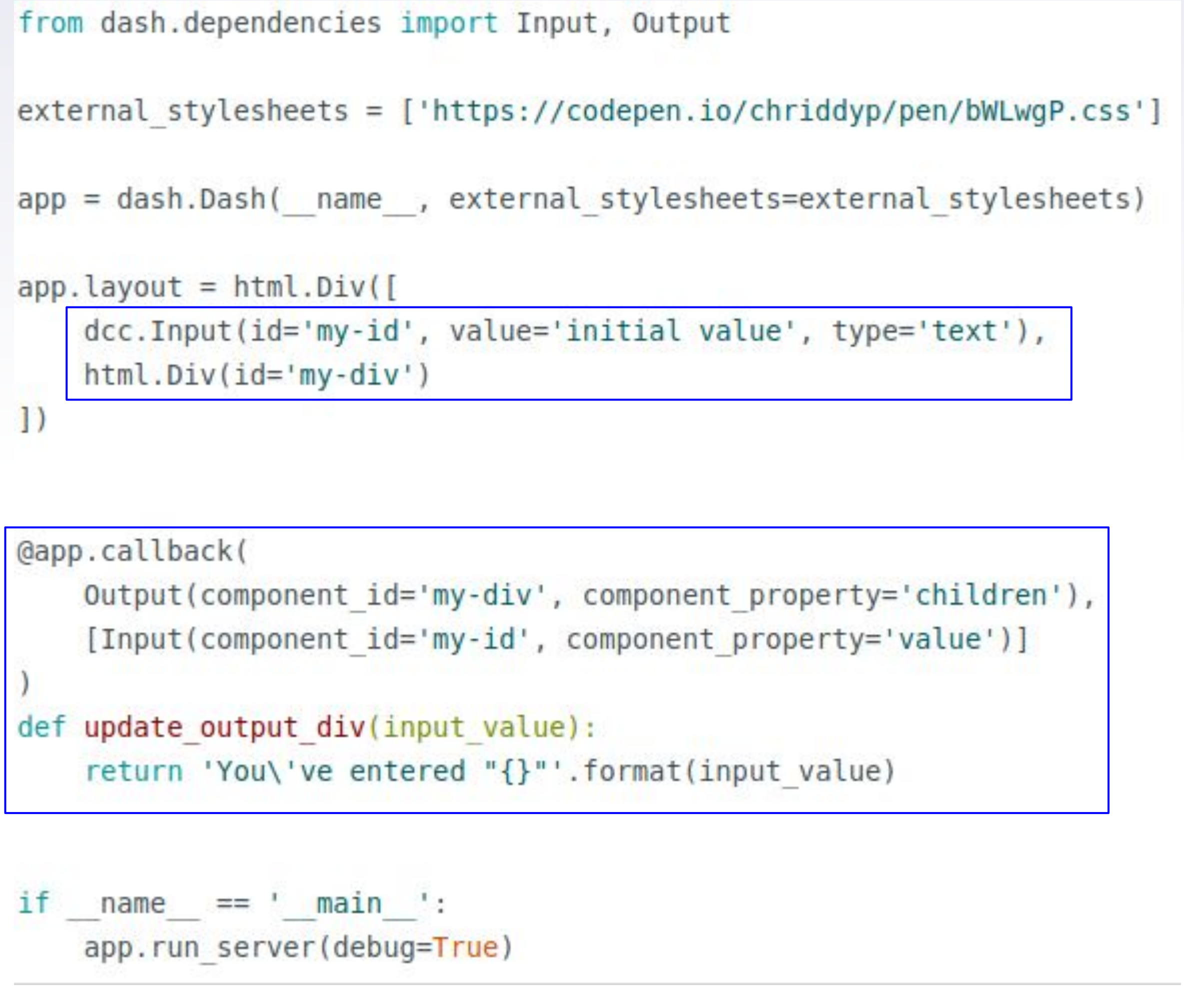
```
import dash
import dash_core_components as dcc
import dash_html_components as html

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__, external_stylesheets=external_stylesheets)

app.layout = html.Div(children=[
    html.H1(children='Hello Dash'),

    html.Div(children='''
        Dash: A web application framework for Python.
    '''),

    dcc.Graph(
        id='example-graph',
        figure={
            'data': [
                {'x': [1, 2, 3], 'y': [4, 1, 2], 'type': 'bar', 'name': 'SF'},
                {'x': [1, 2, 3], 'y': [2, 4, 5], 'type': 'bar', 'name': u'Montréal'},
            ],
            'layout': {
                'title': 'Dash Data Visualization'
            }
        }
    )
])

if __name__ == '__main__':
    app.run_server(debug=True)
```



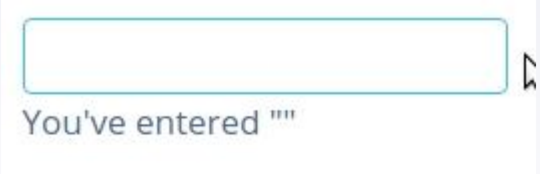
Hello Dash

Dash: A web application framework for Python.

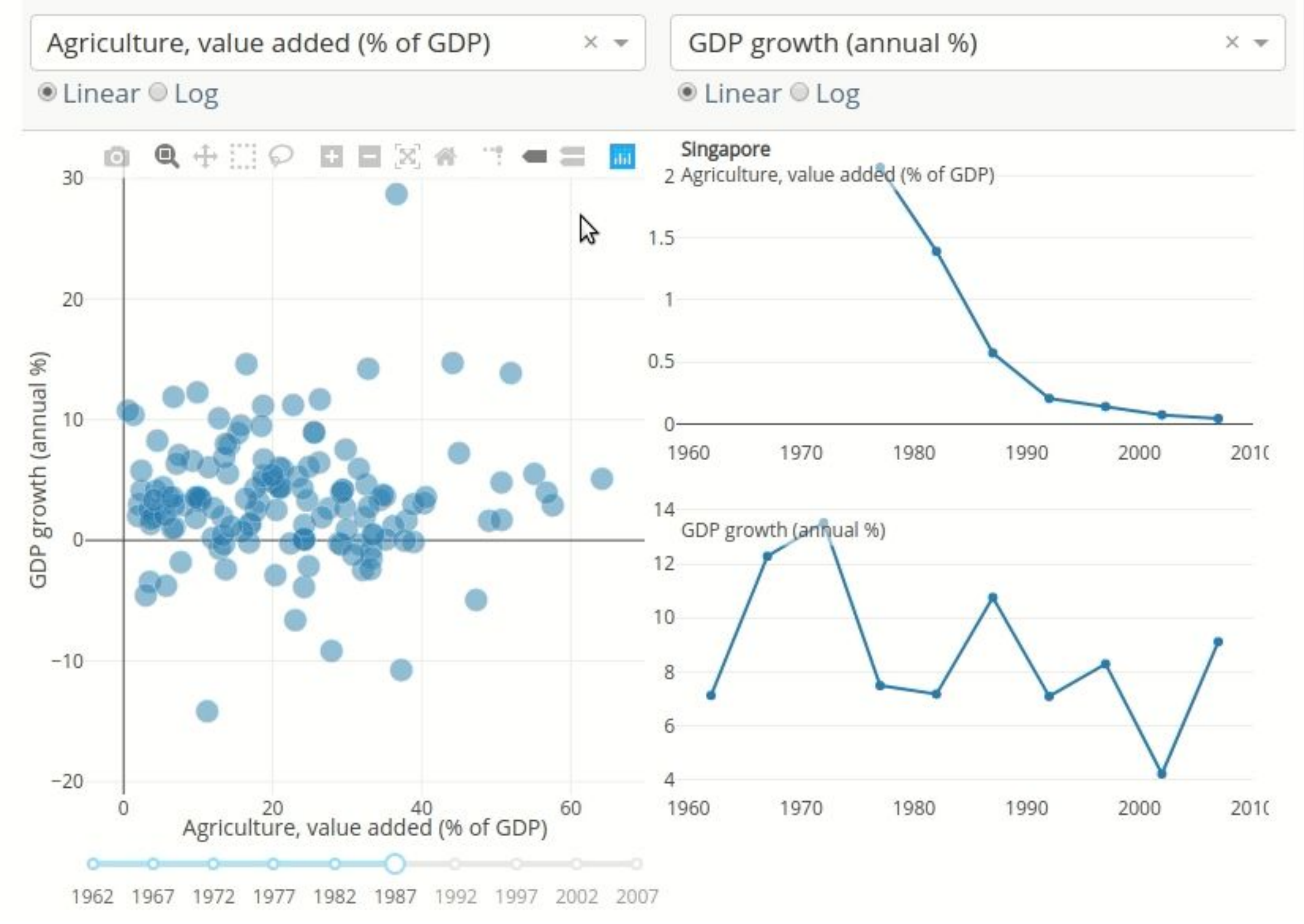
Callbacks



```
from dash.dependencies import Input, Output

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__, external_stylesheets=external_stylesheets)

app.layout = html.Div([
    dcc.Input(id='my-id', value='initial value', type='text'),
    html.Div(id='my-div')
])


@app.callback(
    Output(component_id='my-div', component_property='children'),
    [Input(component_id='my-id', component_property='value')]
)
def update_output_div(input_value):
    return 'You\'ve entered "{}"'.format(input_value)


if __name__ == '__main__':
    app.run_server(debug=True)
```



You've entered ""



Agriculture, value added (% of GDP) × ▾

● Linear ○ Log

GDP growth (annual %) × ▾

● Linear ○ Log

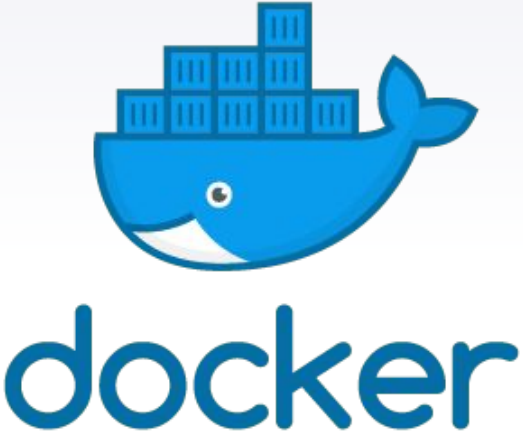
Na produkcję!

NGINX

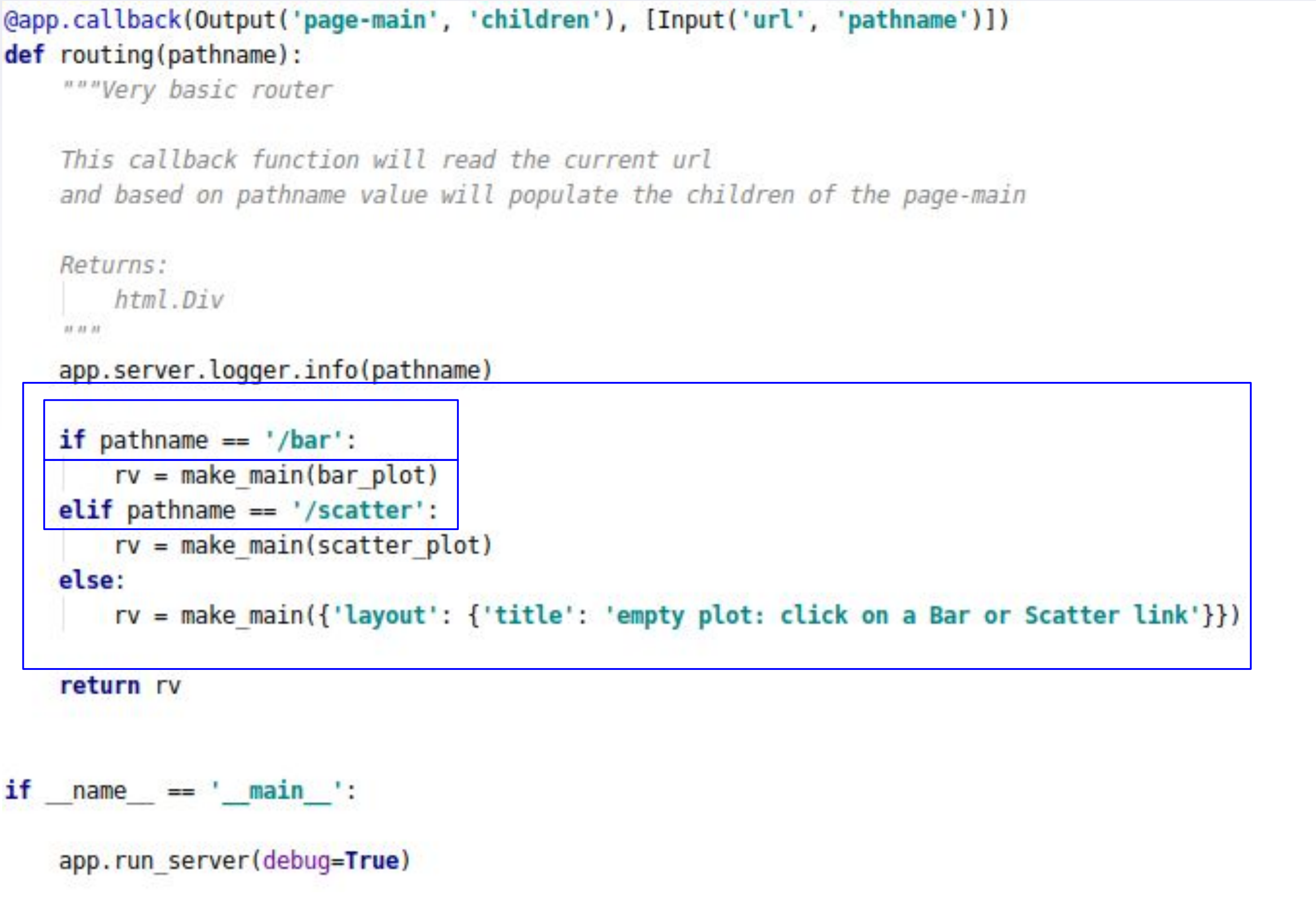
plotly | Dash

gunicorn

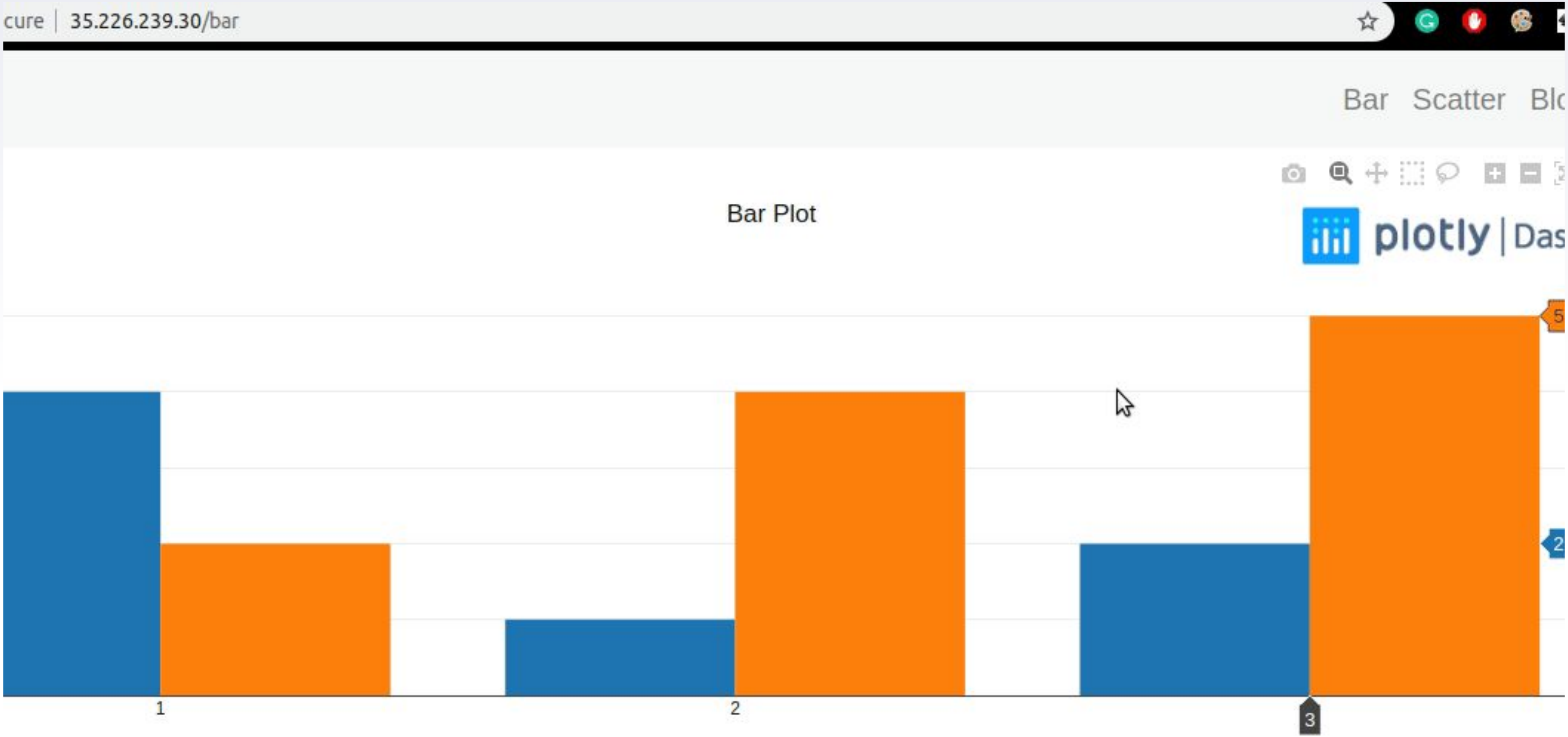
Flask



https://github.com/sladkovm/docker-flask-gunicorn-nginx



```
@app.callback(Output('page-main', 'children'), [Input('url', 'pathname')])
def routing(pathname):
    """Very basic router

    This callback function will read the current url
    and based on pathname value will populate the children of the page-main

    Returns:
        html.Div
    """
    app.server.logger.info(pathname)

    if pathname == '/bar':
        rv = make_main(bar_plot)
    elif pathname == '/scatter':
        rv = make_main(scatter_plot)
    else:
        rv = make_main({'layout': {'title': 'empty plot: click on a Bar or Scatter link'}})

    return rv


if __name__ == '__main__':

    app.run_server(debug=True)
```


cure | 35.226.239.30/bar
Bar
Scatter
Blo
Bar Plot
plotly | Das
5
2
1
2
3

Dziękuję

Wojciech Łaguna