Module 02

Dash Plotly

Data Science Developer

Purwadhika
Startup and Coding School

Plotly

Plotly is a library that allows you to create interactive plots that you can use in dashboards or websites (you can save them as html files or static images).

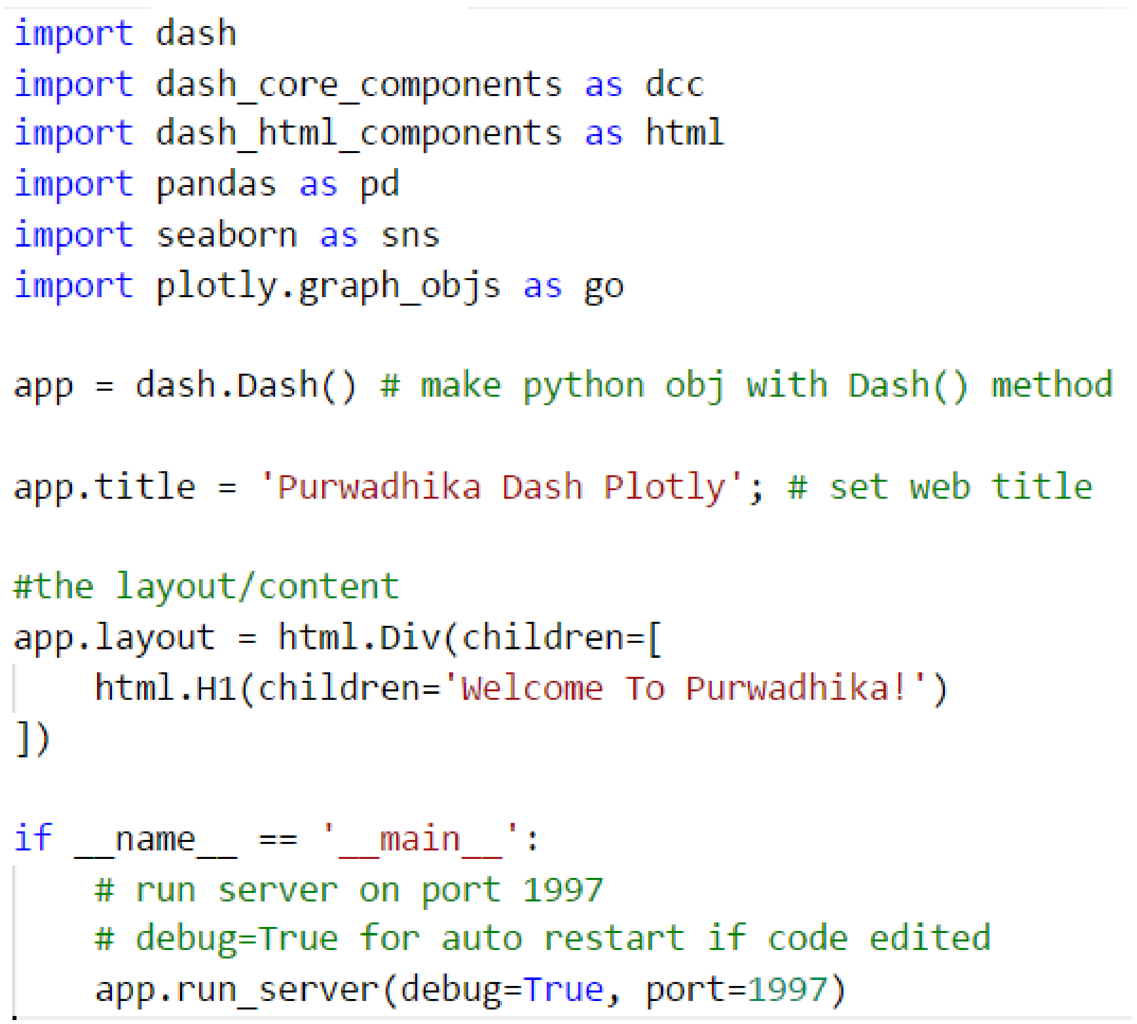
Dash

- Dash is a productive Python framework for building web applications.
- Written on top of Flask, Plotly.js, and React.js, Dash is ideal for building data visualization apps with highly custom user interfaces in pure Python. It's particularly suited for anyone who works with data in Python.

Installation

- pip install dash==0.26.5 # The core dash backend
- pip install dash-html-components==0.12.0 # HTML components
- pip install dash-core-components==0.26.0 # Supercharged components

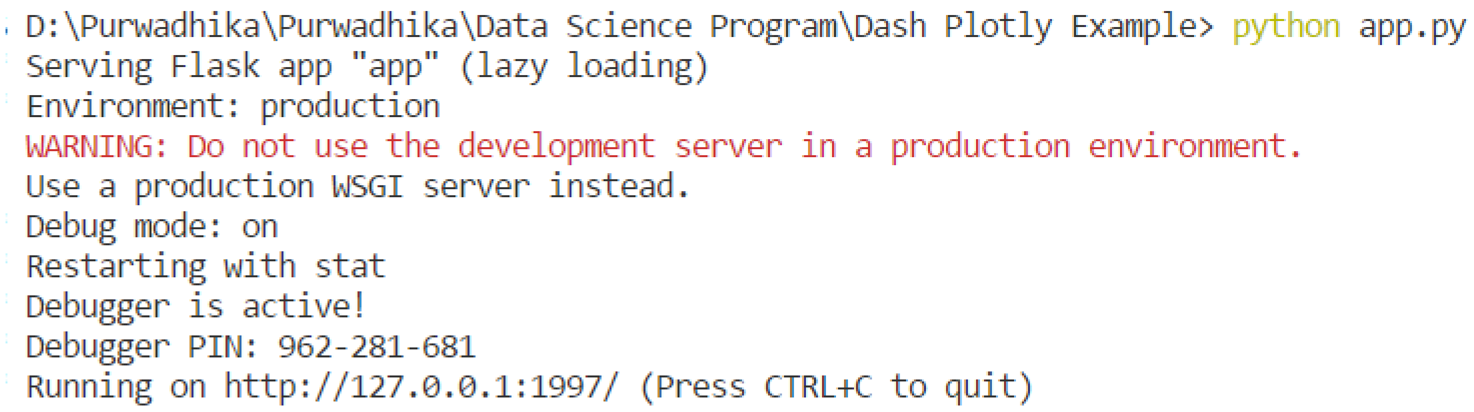
Imports and Setup (First Code)



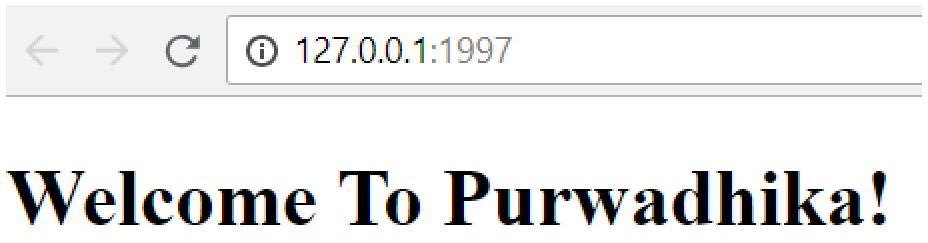
```python
import dash
import dash_core_components as dcc
import dash_html_components as html
import pandas as pd
import seaborn as sns
import plotly.graph_objs as go

app = dash.Dash() # make python obj with Dash() method

app.title = 'Purwadhika Dash Plotly'; # set web title

#the layout/content
app.layout = html.Div(children=[
    html.H1(children='Welcome To Purwadhika!')
])

if __name__ == '__main__':
    # run server on port 1997
    # debug=True for auto restart if code edited
    app.run_server(debug=True, port=1997)
```

Purwadhika
Startup and Coding School

Run The Program



```
D:\Purwadhika\Purwadhika\Data Science Program\Dash Plotly Example> python app.py
Serving Flask app "app" (lazy loading)
Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
Debug mode: on
Restarting with stat
Debugger is active!
Debugger PIN: 962-281-681
Running on http://127.0.0.1:1997/ (Press CTRL+C to quit)
```



127.0.0.1:1997

Welcome To Purwadhika!

Dash HTML Components

- Dash is a web application framework that provides pure Python abstraction around HTML, CSS, and JavaScript.
- Instead of writing HTML or using an HTML templating engine, you compose your layout using Python structures with the dash-html-components library.

Open this link for details :
https://dash.plot.ly/dash-html-components

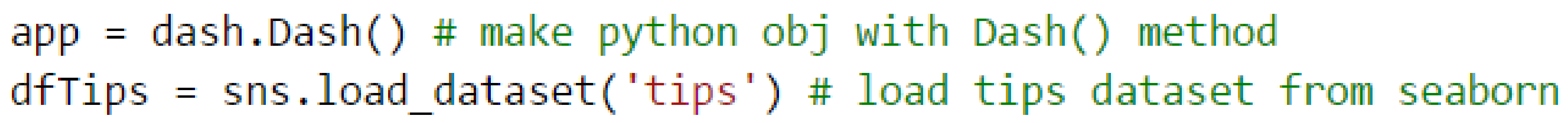
Purwadhika
Startup and Coding School

Dash Core Components

- Dash ships with supercharged components for interactive user interfaces. A core set of components, written and maintained by the Dash team, is available in the dash-core-components library.

Open this link for details :
https://dash.plot.ly/dash-core-components

Load Dataset

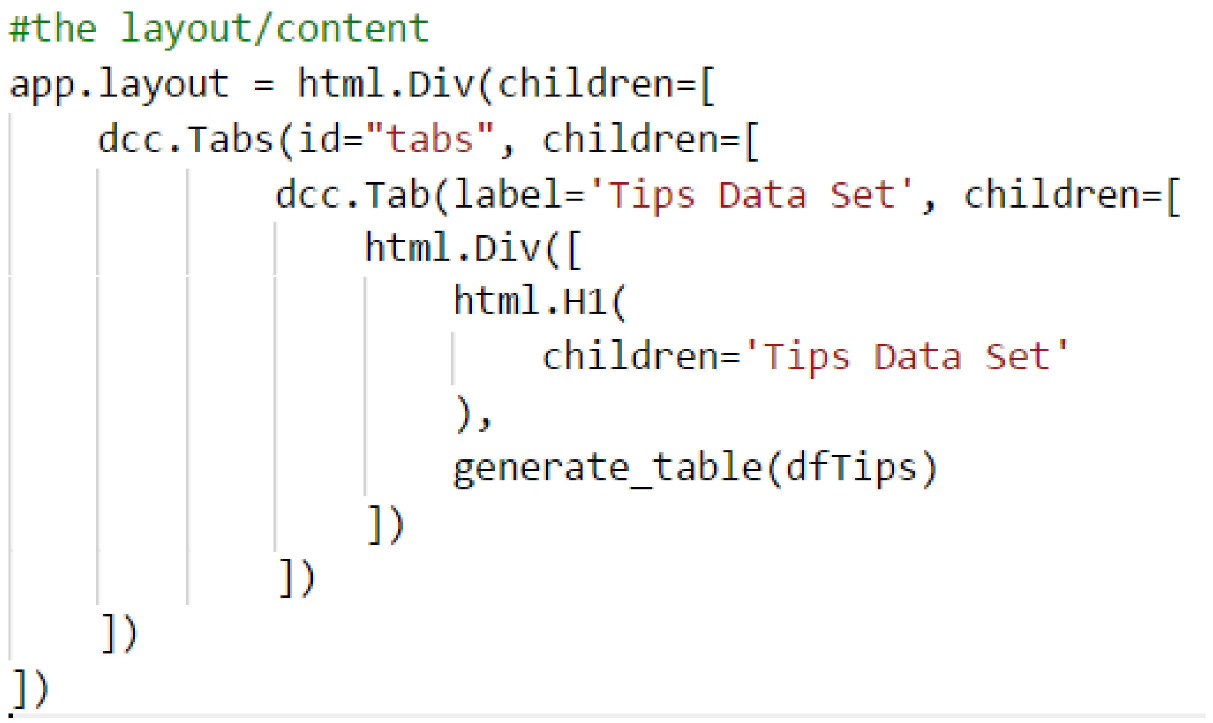


```python
app = dash.Dash() # make python obj with Dash() method
dfTips = sns.load_dataset('tips') # load tips dataset from seaborn
```

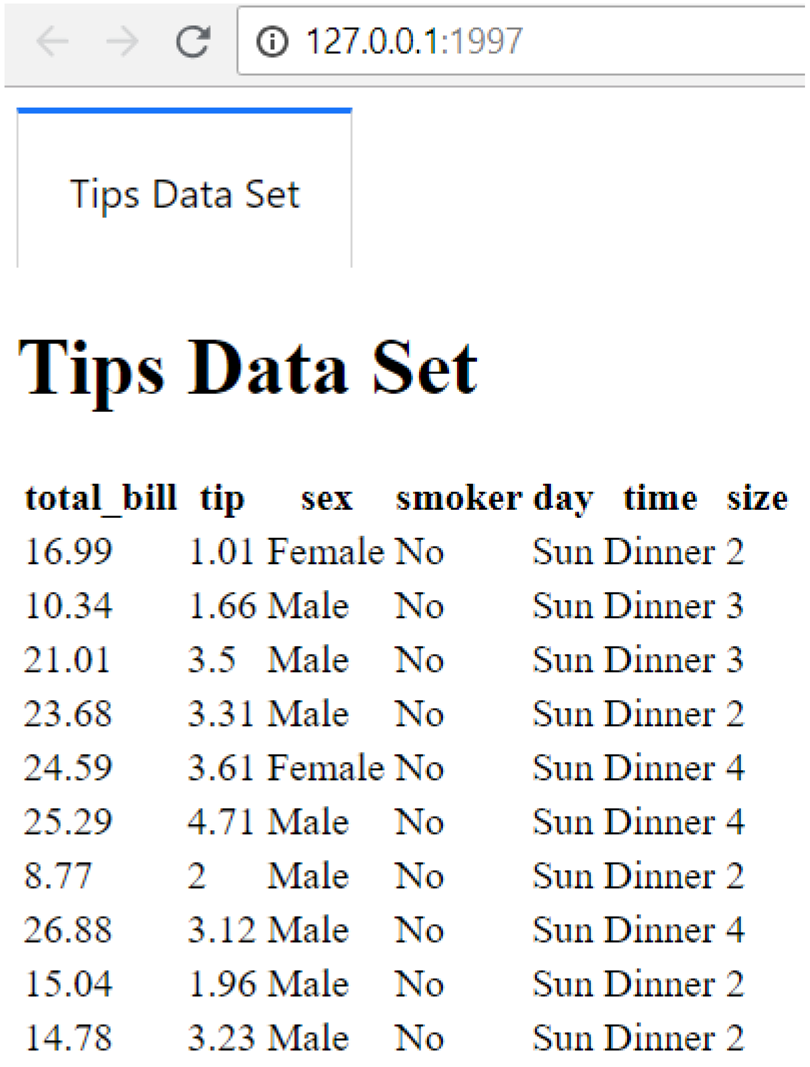
Create Function generate_table

```python
# function to generate HTML Table
def generate_table(dataframe, max_rows=10):
    return html.Table(
        # Header
        [html.Tr([html.Th(col) for col in dataframe.columns])] +

        # Body
        [html.Tr([
            html.Td(dataframe.iloc[i][col]) for col in dataframe.columns
        ]) for i in range(min(len(dataframe), max_rows))]
    )
```

Change The Layout



```
#the layout/content
app.layout = html.Div(children=[
    dcc.Tabs(id="tabs", children=[
            dcc.Tab(label='Tips Data Set', children=[
                html.Div([
                    html.H1(
                        children='Tips Data Set'
                    ),
                    generate_table(dfTips)
                ])
            ])
    ])
])
```

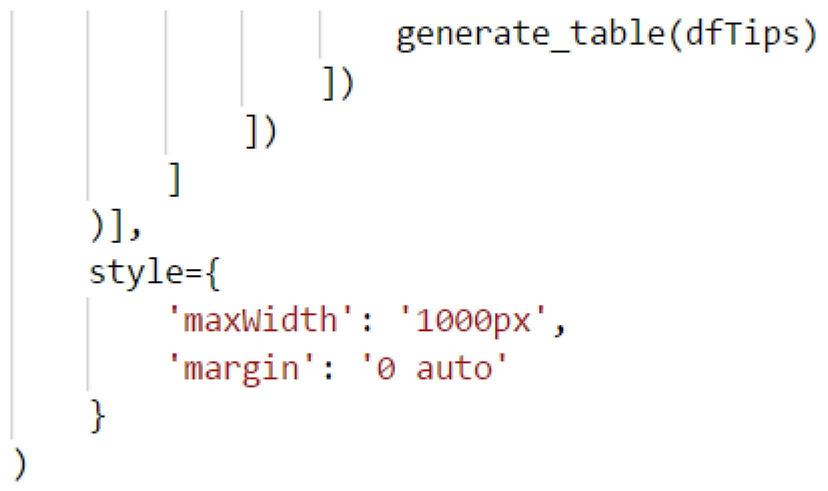


Tips Data Set

total_bill	tip	sex	smoker	day	time	size
16.99	1.01	Female	No	Sun	Dinner	2
10.34	1.66	Male	No	Sun	Dinner	3
21.01	3.5	Male	No	Sun	Dinner	3
23.68	3.31	Male	No	Sun	Dinner	2
24.59	3.61	Female	No	Sun	Dinner	4
25.29	4.71	Male	No	Sun	Dinner	4
8.77	2	Male	No	Sun	Dinner	2
26.88	3.12	Male	No	Sun	Dinner	4
15.04	1.96	Male	No	Sun	Dinner	2
14.78	3.23	Male	No	Sun	Dinner	2

Purwadhika
Startup and Coding School

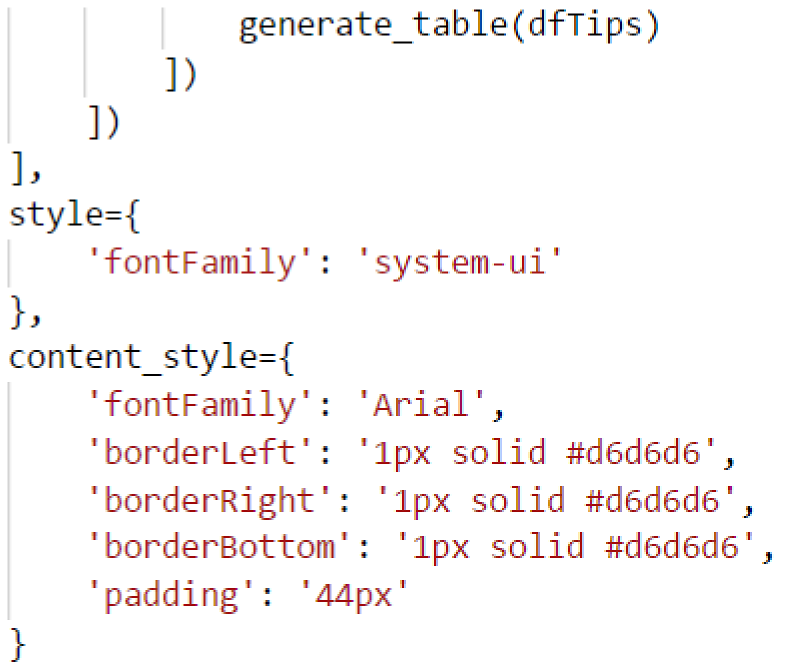
Give Style(CSS) to Parent Div



```
                    generate_table(dfTips)
                ])
            ])
        ]
    )],
    style={
        'maxWidth': '1000px',
        'margin': '0 auto'
    }
)
```

- Now the Div will be on the center of the screen because of the margin : 0 auto, and the Div max width will be = 1000px.

Give Style(CSS) to dcc.Tabs

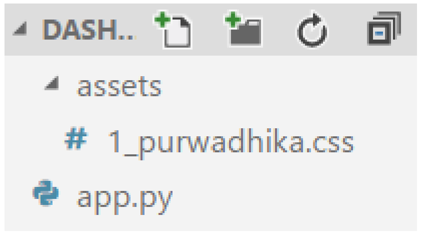


```
                generate_table(dfTips)
            ])
        ])
],
style={
    'fontFamily': 'system-ui'
},
content_style={
    'fontFamily': 'Arial',
    'borderLeft': '1px solid #d6d6d6',
    'borderRight': '1px solid #d6d6d6',
    'borderBottom': '1px solid #d6d6d6',
    'padding': '44px'
}
```

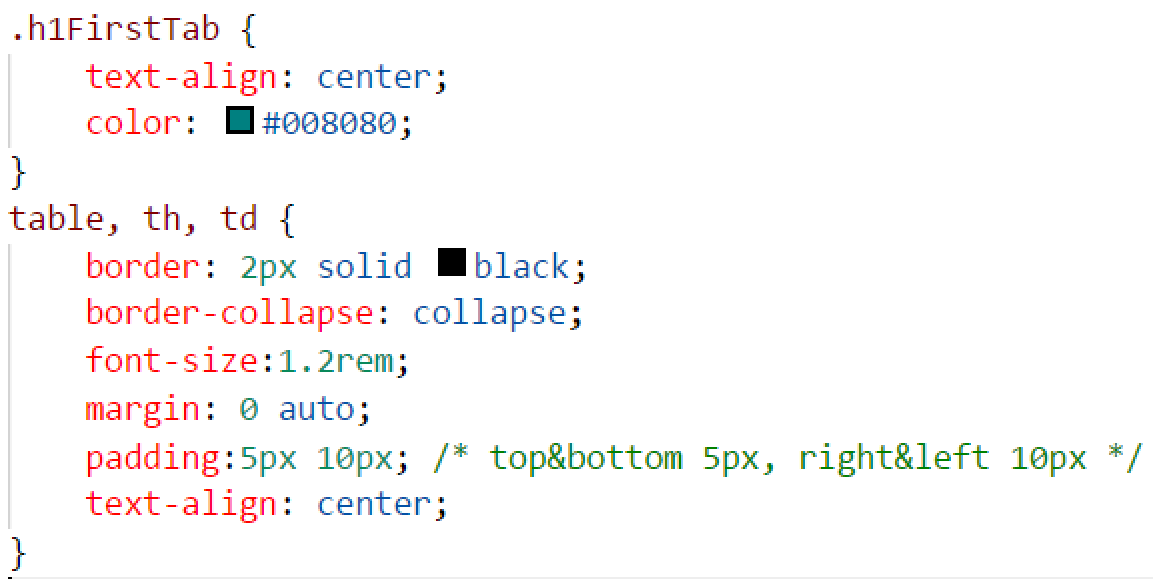
- Now the tabs fontFamily = system-ui
- and the content of dcc.Tabs will have a border on all sides except the top, and the fontFamily changed to Arial

Adding & Using CSS Files

- Create css file 1_purwadhika.css inside new assets folder
- The CSS files on the assets folder will automatically imported and used by the app.py because of Dash
- Dash will include the files in alphanumerical order by filename. So, we recommend prefixing your filenames with numbers if you need to ensure their order (e.g. 10_typography.css, 20_header.css)

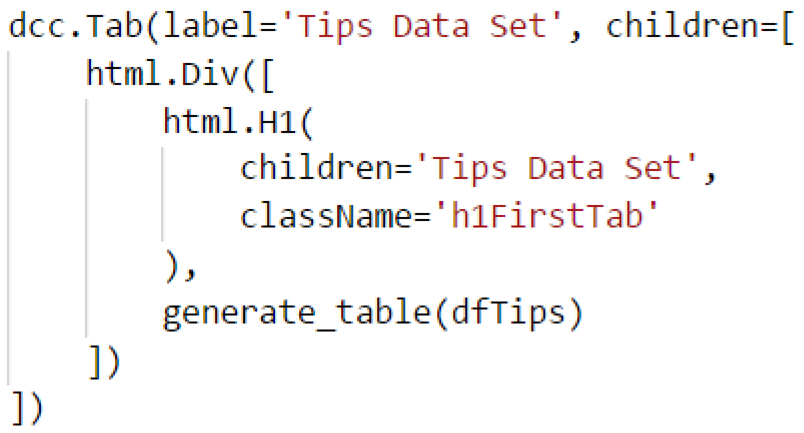


1_purwadhika.css



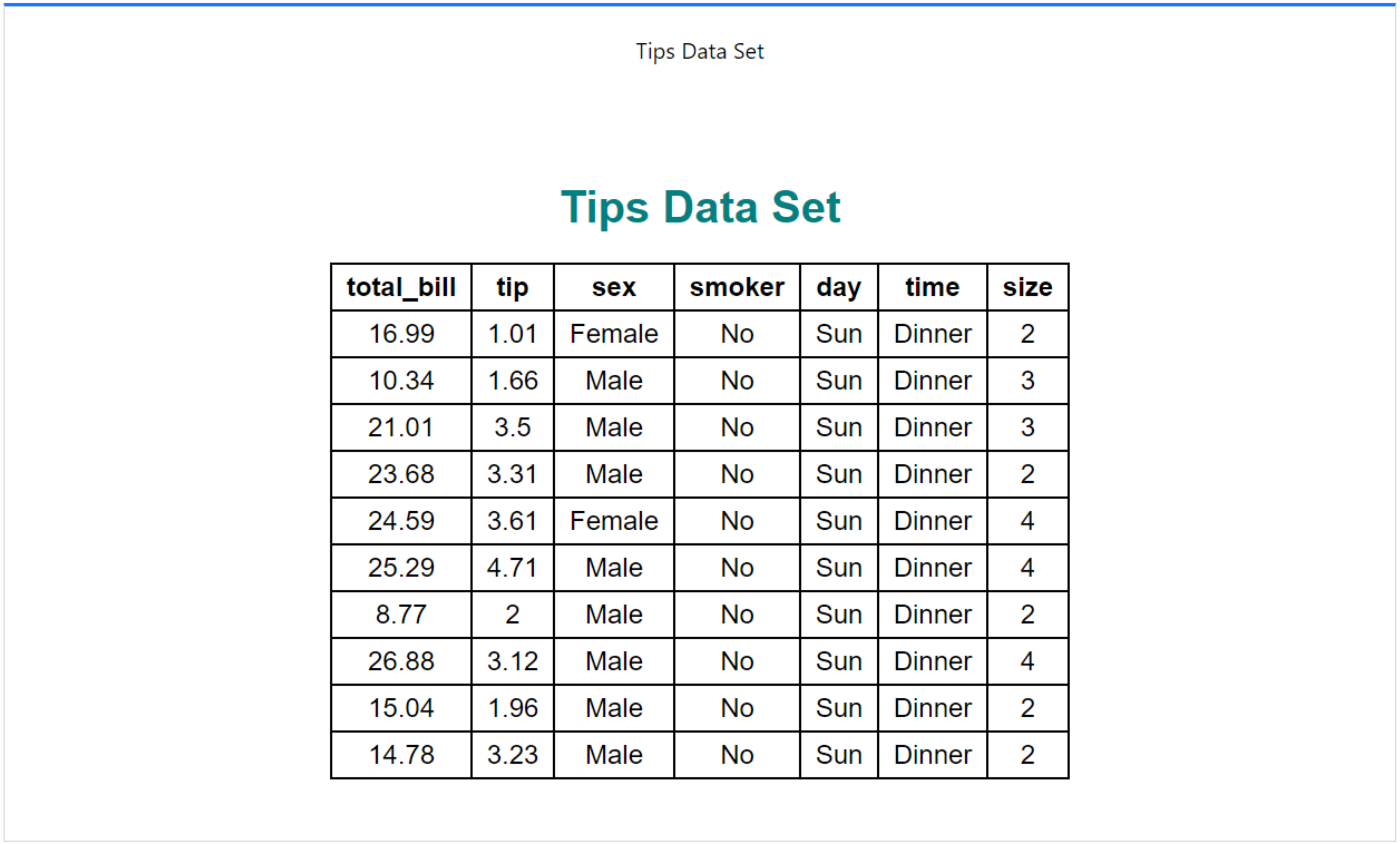
```css
.h1FirstTab {
    text-align: center;
    color: #008080;
}
table, th, td {
    border: 2px solid black;
    border-collapse: collapse;
    font-size:1.2rem;
    margin: 0 auto;
    padding:5px 10px; /* top&bottom 5px, right&left 10px */
    text-align: center;
}
```

App.py



```
dcc.Tab(label='Tips Data Set', children=[
    html.Div([
        html.H1(
            children='Tips Data Set',
            className='h1FirstTab'
        ),
        generate_table(dfTips)
    ])
])
```

- Give className property to H1 Element inside first Tab, and the value = ‘h1FirstTab’



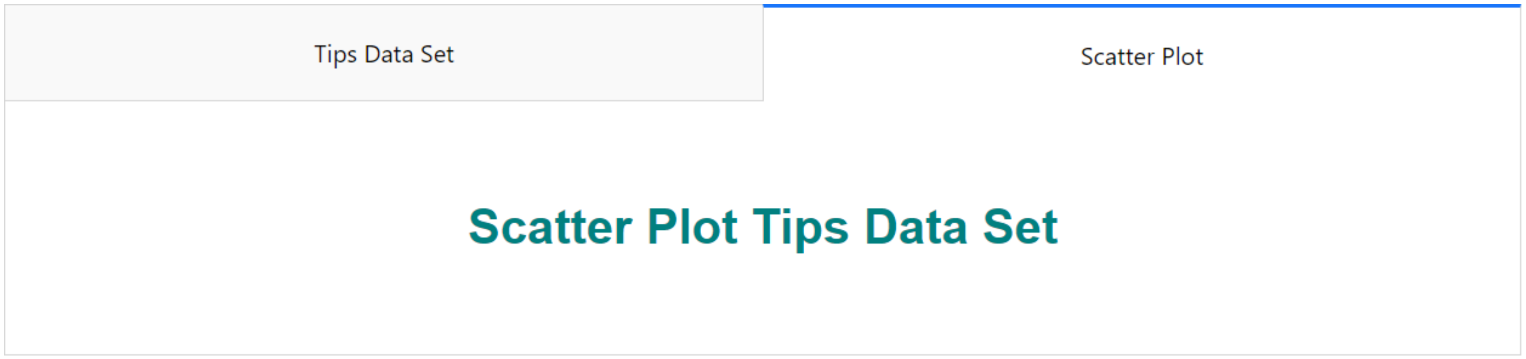
Tips Data Set

Tips Data Set

total_bill	tip	sex	smoker	day	time	size
16.99	1.01	Female	No	Sun	Dinner	2
10.34	1.66	Male	No	Sun	Dinner	3
21.01	3.5	Male	No	Sun	Dinner	3
23.68	3.31	Male	No	Sun	Dinner	2
24.59	3.61	Female	No	Sun	Dinner	4
25.29	4.71	Male	No	Sun	Dinner	4
8.77	2	Male	No	Sun	Dinner	2
26.88	3.12	Male	No	Sun	Dinner	4
15.04	1.96	Male	No	Sun	Dinner	2
14.78	3.23	Male	No	Sun	Dinner	2

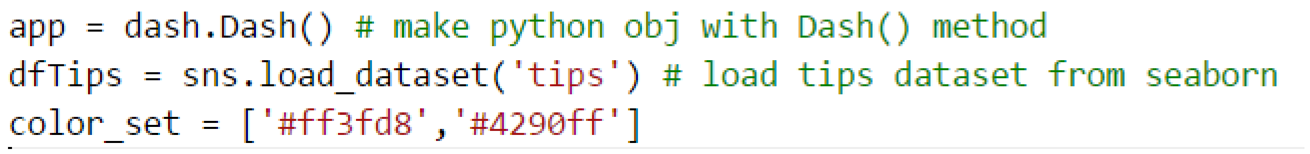
Add New Tab

```
dcc.Tab(label='Scatter Plot', children=[
    html.Div([
        html.H1(
            children='Scatter Plot Tips Data Set',
            className='h1FirstTab'
        )
    ])
])
```



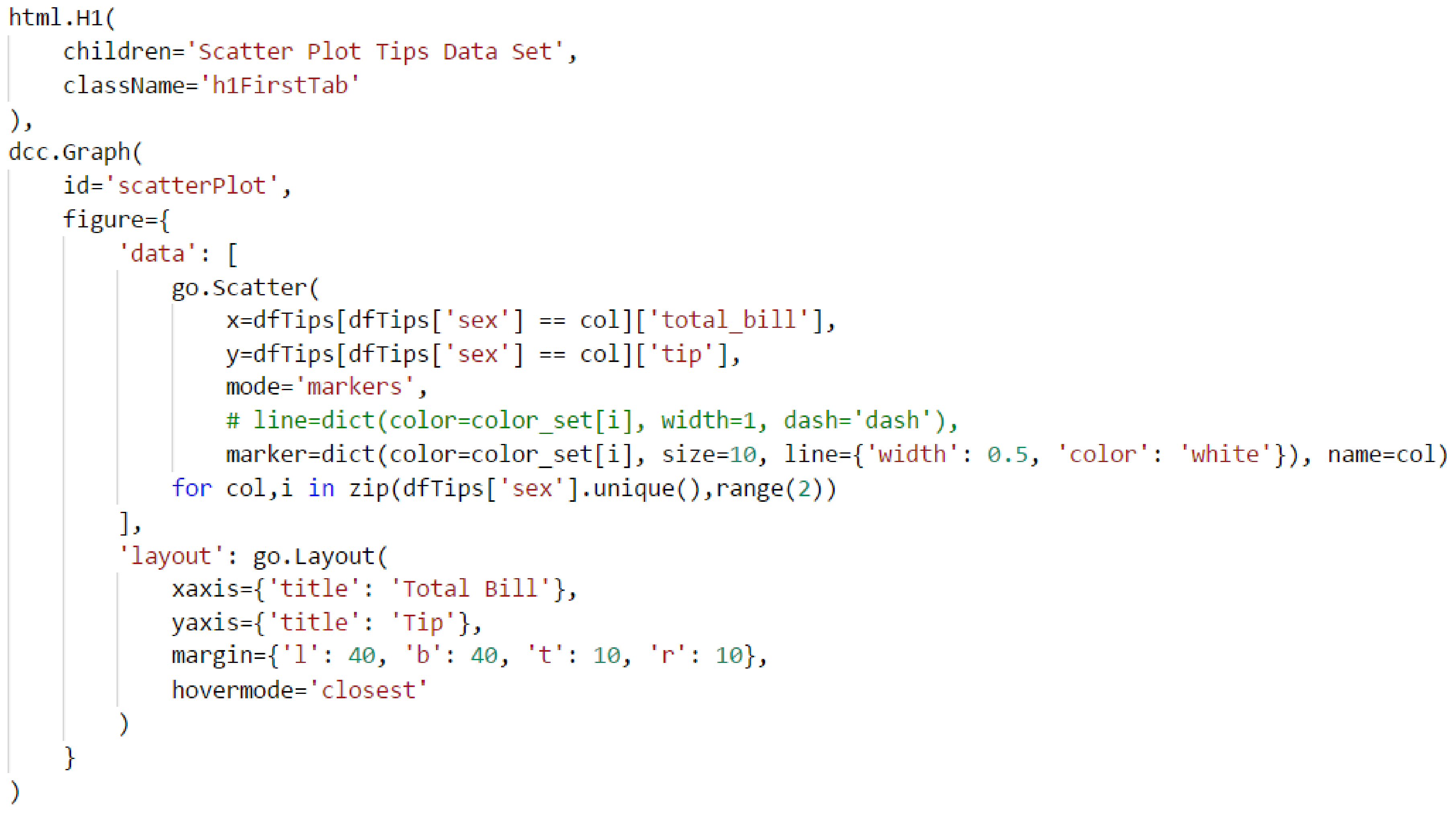
Scatter Plot Tips Data Set

Add Python List color_set

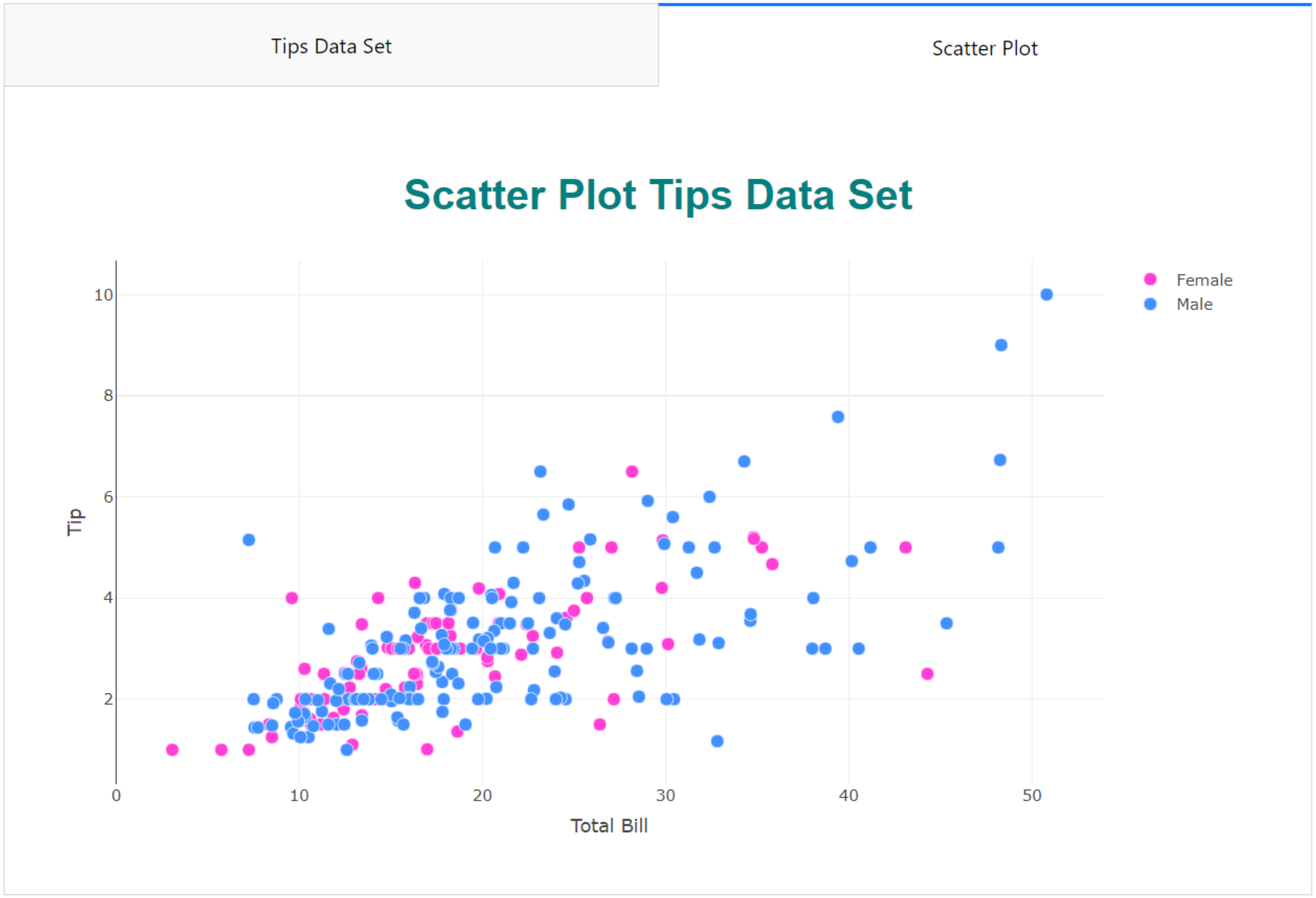


```
app = dash.Dash() # make python obj with Dash() method
dfTips = sns.load_dataset('tips') # load tips dataset from seaborn
color_set = ['#ff3fd8','#4290ff']
```

Add dcc.Graph below H1 Tab 2



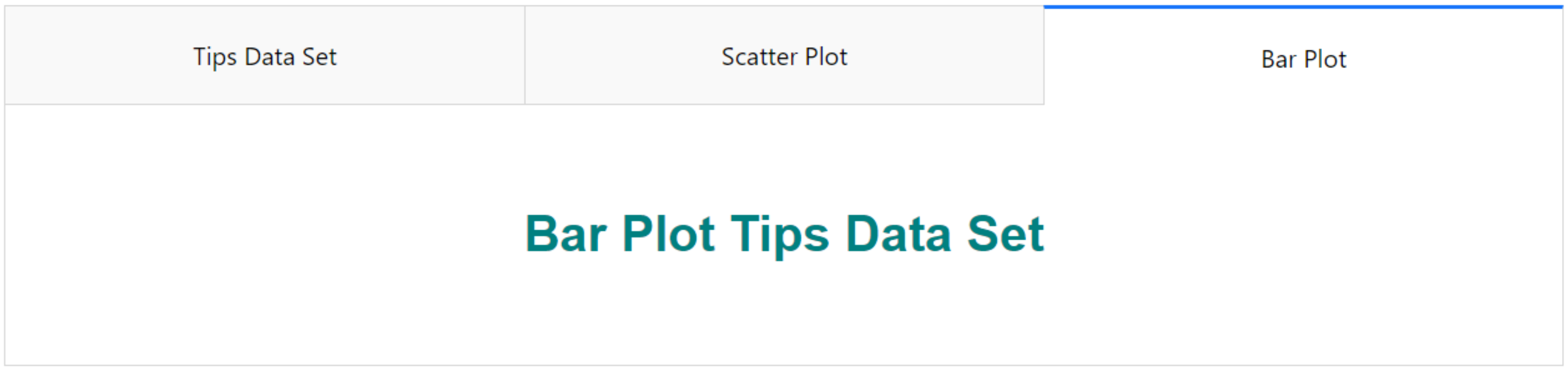
```
html.H1(
    children='Scatter Plot Tips Data Set',
    className='h1FirstTab'
),
dcc.Graph(
    id='scatterPlot',
    figure={
        'data': [
            go.Scatter(
                x=dfTips[dfTips['sex'] == col]['total_bill'],
                y=dfTips[dfTips['sex'] == col]['tip'],
                mode='markers',
                # line=dict(color=color_set[i], width=1, dash='dash'),
                marker=dict(color=color_set[i], size=10, line={'width': 0.5, 'color': 'white'}), name=col)
            for col,i in zip(dfTips['sex'].unique(),range(2))
        ],
        'layout': go.Layout(
            xaxis={'title': 'Total Bill'},
            yaxis={'title': 'Tip'},
            margin={'l': 40, 'b': 40, 't': 10, 'r': 10},
            hovermode='closest'
        )
    }
)
```



Purwadhika
Startup and Coding School

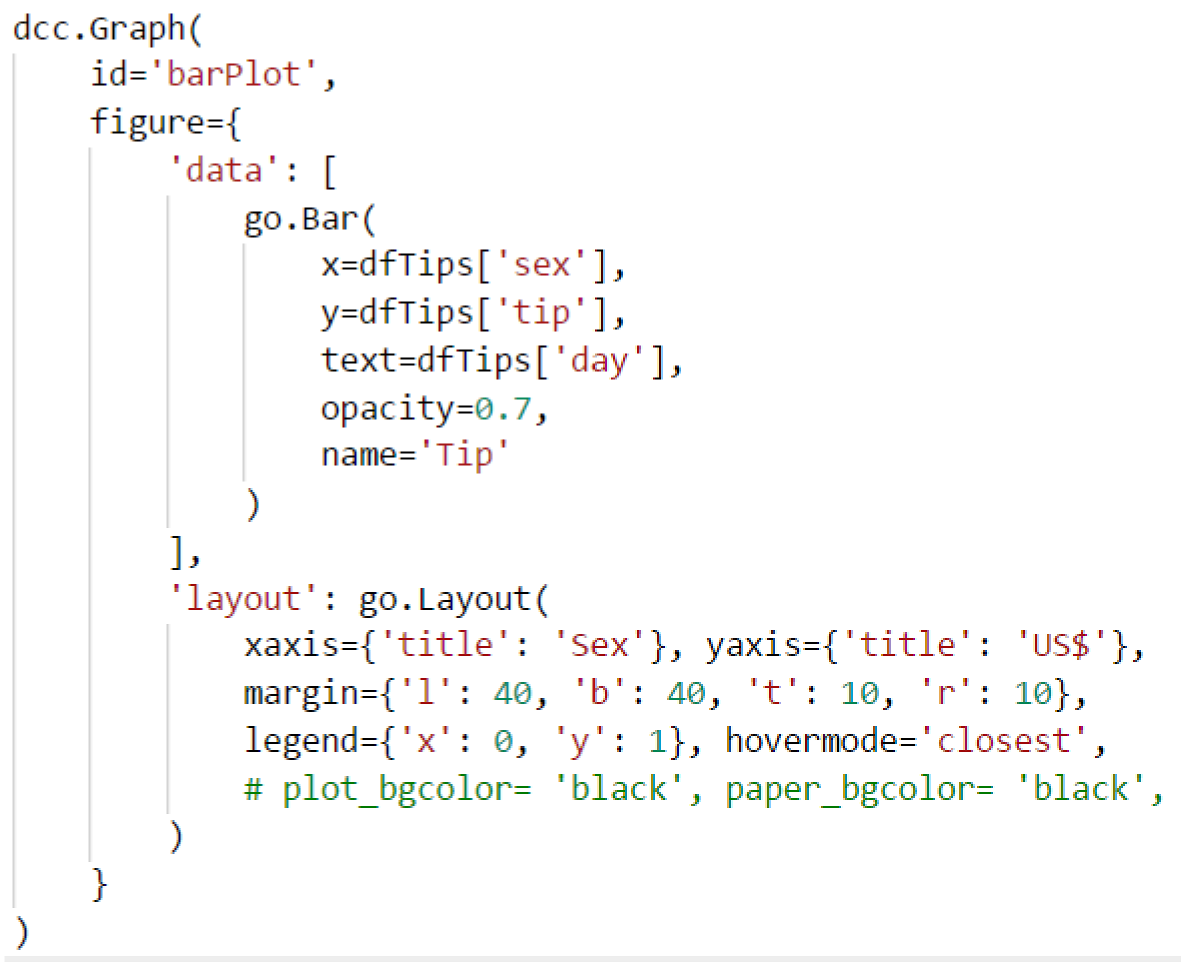
Add New Tab

```
dcc.Tab(label='Bar Plot', children=[
    html.Div([
        html.H1(
            children='Bar Plot Tips Data Set',
            className='h1FirstTab'
        )
    ])
])
```

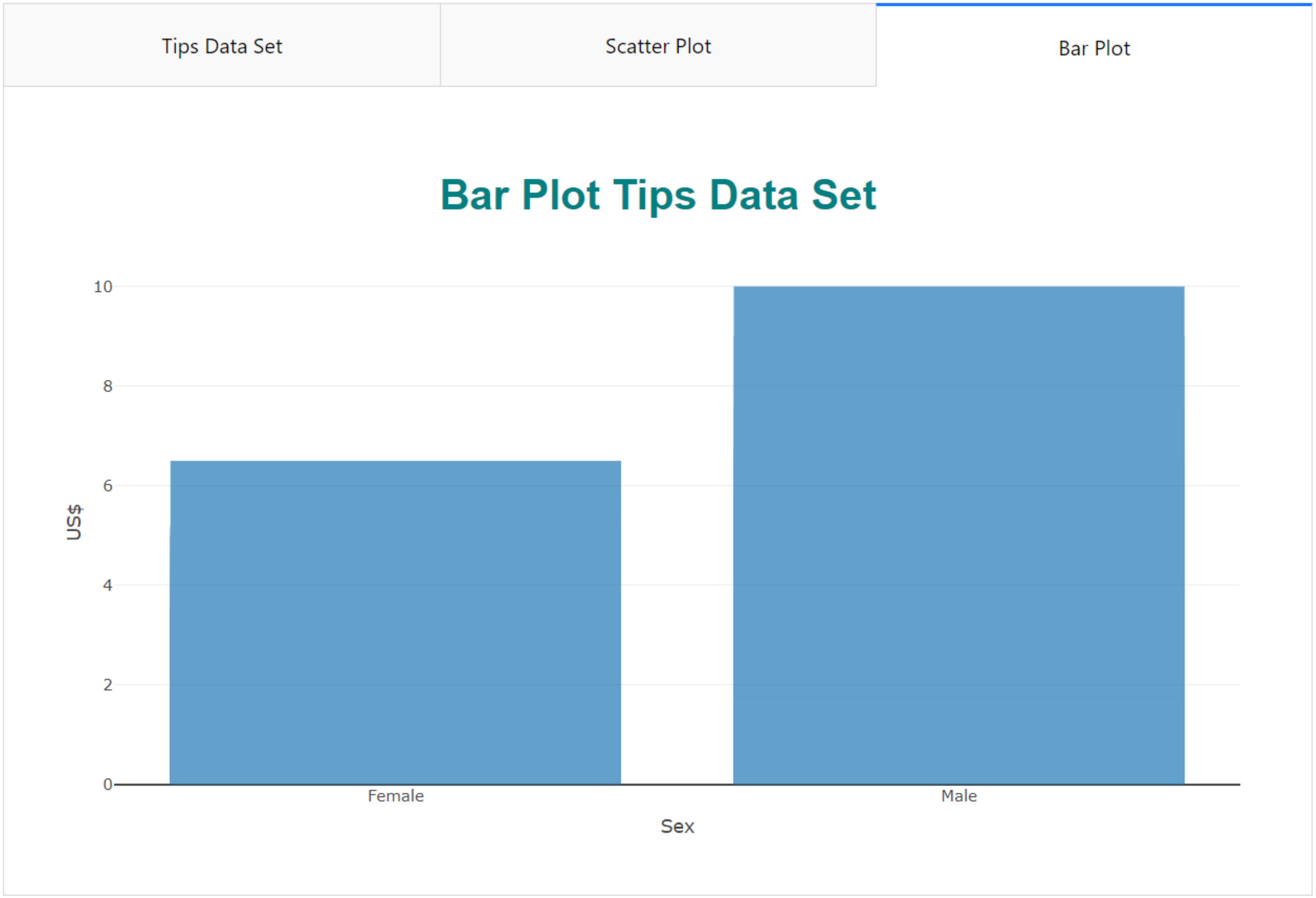


Bar Plot Tips Data Set

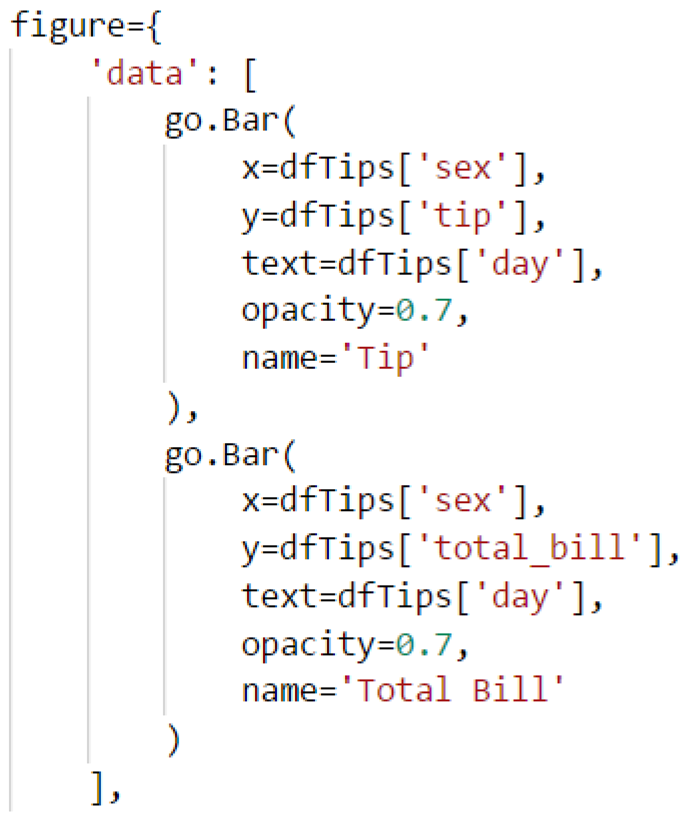
Add dcc.Graph below H1 Tab 3



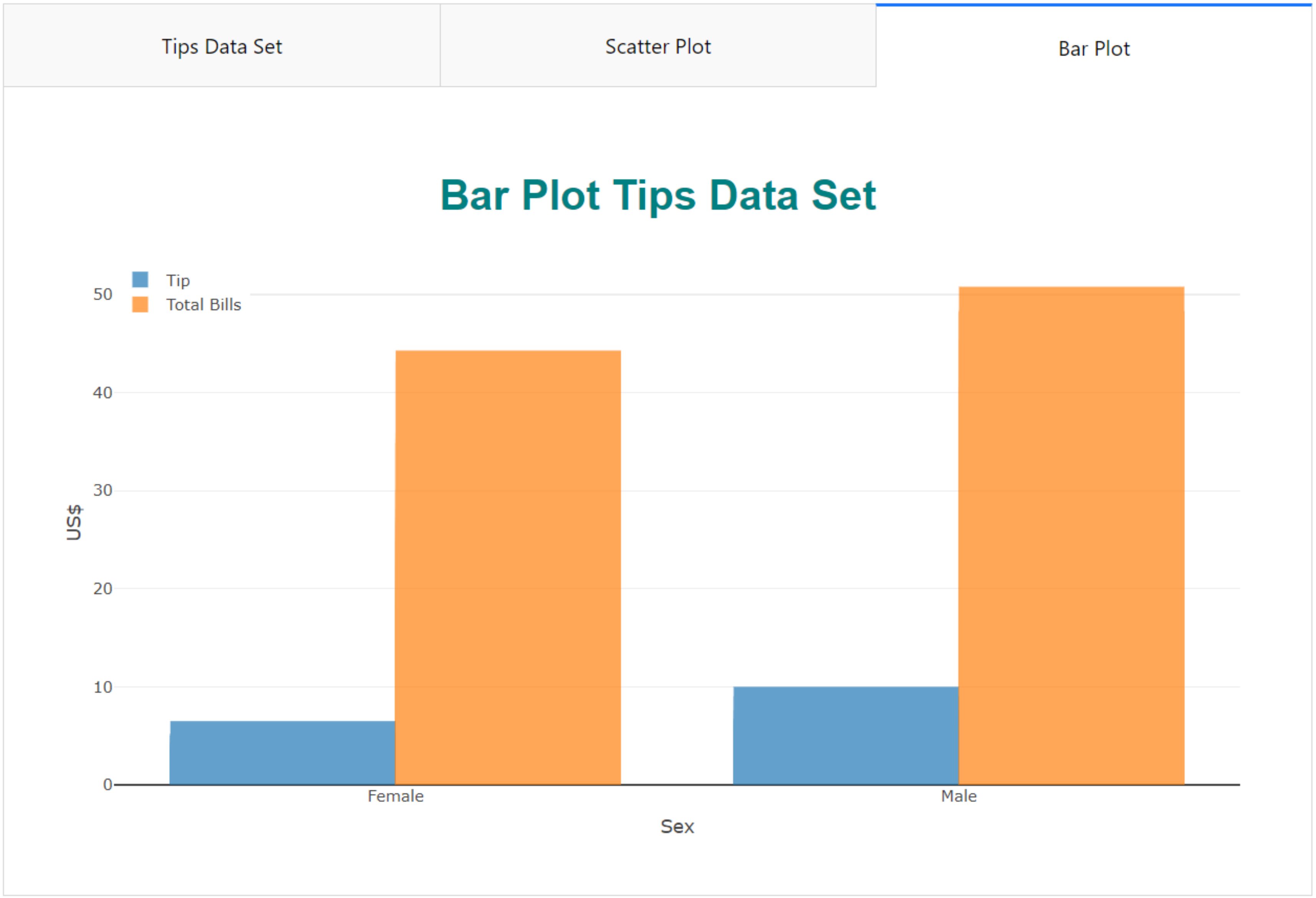
```
dcc.Graph(
    id='barPlot',
    figure={
        'data': [
            go.Bar(
                x=dfTips['sex'],
                y=dfTips['tip'],
                text=dfTips['day'],
                opacity=0.7,
                name='Tip'
            )
        ],
        'layout': go.Layout(
            xaxis={'title': 'Sex'}, yaxis={'title': 'US$'},
            margin={'l': 40, 'b': 40, 't': 10, 'r': 10},
            legend={'x': 0, 'y': 1}, hovermode='closest',
            # plot_bgcolor= 'black', paper_bgcolor= 'black',
        )
    }
)
```


Tips Data Set
Scatter Plot
Bar Plot
Bar Plot Tips Data Set
10
8
6
4
2
0
US$
Female
Male
Sex

Add more go.Bar to figure data List



```
figure={
    'data': [
        go.Bar(
            x=dfTips['sex'],
            y=dfTips['tip'],
            text=dfTips['day'],
            opacity=0.7,
            name='Tip'
        ),
        go.Bar(
            x=dfTips['sex'],
            y=dfTips['total_bill'],
            text=dfTips['day'],
            opacity=0.7,
            name='Total Bill'
        )
    ],
```


Tips Data Set
Scatter Plot
Bar Plot
Bar Plot Tips Data Set
Tip
Total Bills
US$
50
40
30
20
10
0
Female
Male
Sex

Basic Dash Callbacks

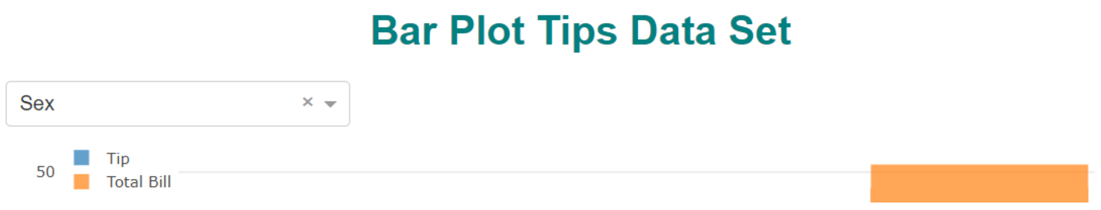
- Basic Dash Callbacks make your Dash apps interactive.
- Lets get started!

- For more details : https://dash.plot.ly/getting-started-part-2

Purwadhika
Startup and Coding School

Add Dropdown below H1 Tab 3

```
html.Div([
    html.H1(
        children='Bar Plot Tips Data Set',
        className='h1FirstTab'
    )
]),
html.Div([
        dcc.Dropdown(
            id='ddl-x-bar-plot',
            options=[{'label': 'Sex', 'value': 'sex'},
                    {'label': 'Smoker', 'value': 'smoker'},
                    {'label': 'Day', 'value': 'day'},
                    {'label': 'Time', 'value': 'time'}],
            value='sex'
        )],
        style={'width': '30%'}
),
dcc.Graph(
```

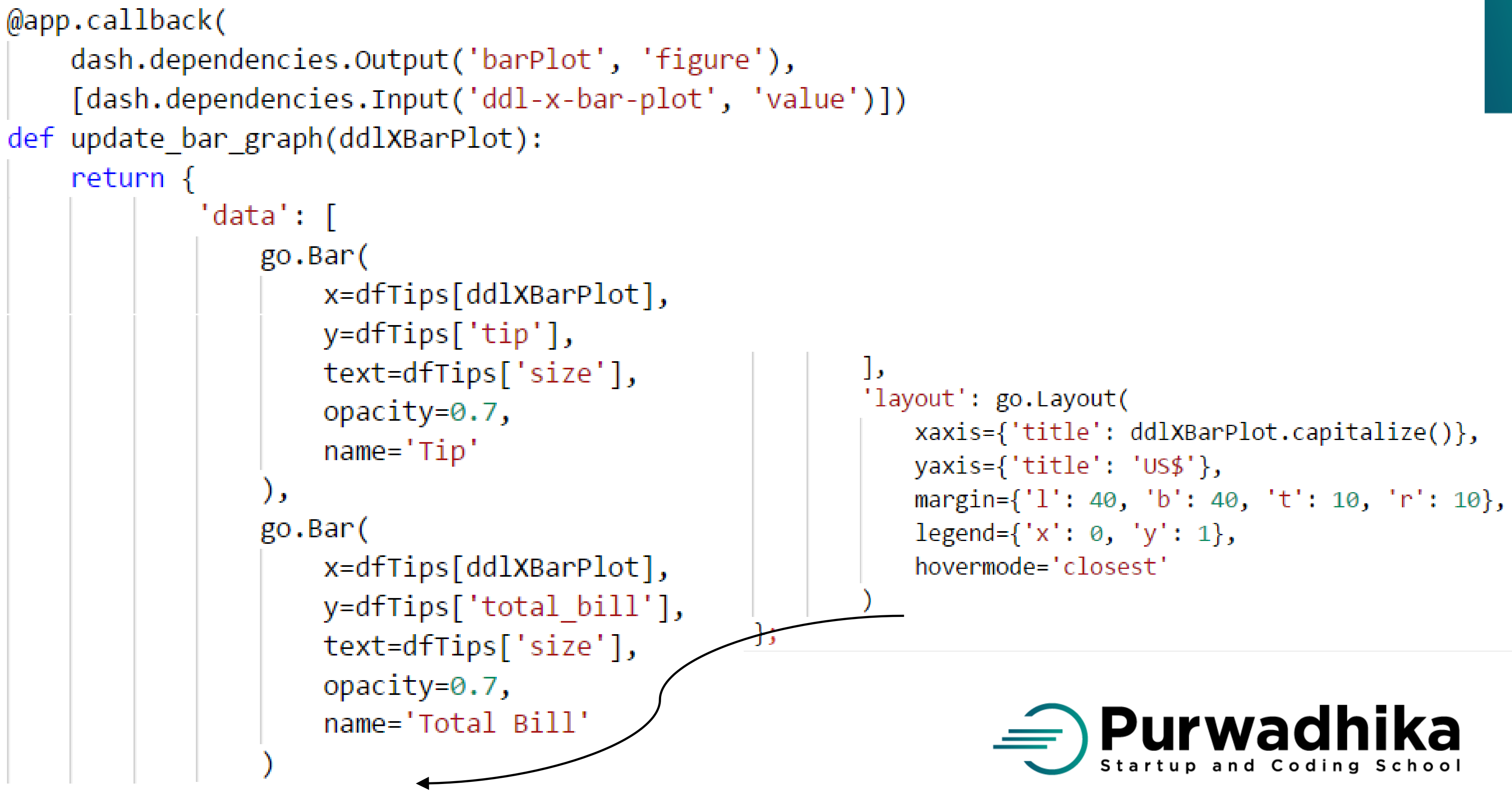


Bar Plot Tips Data Set

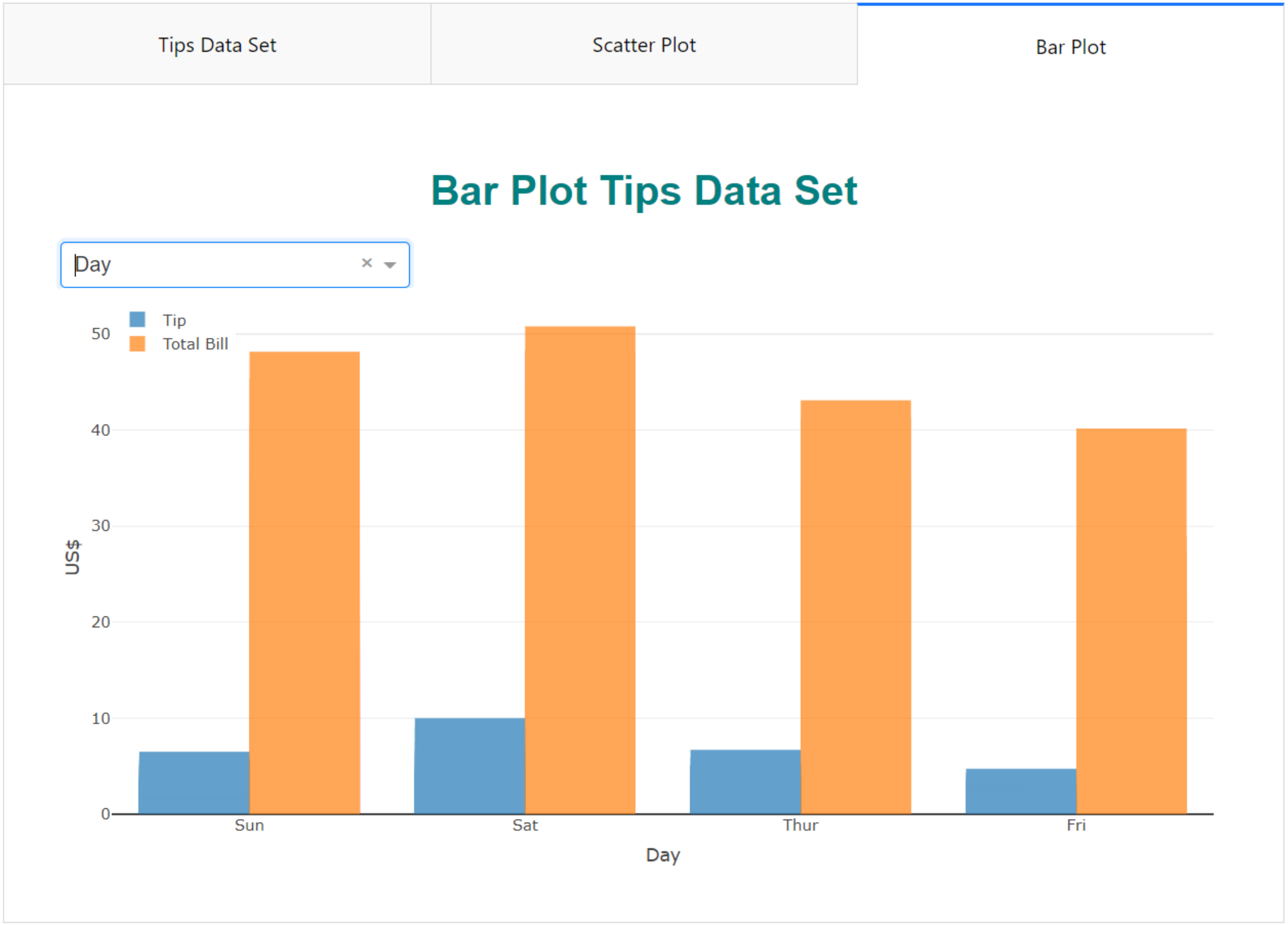
Sex

Add @app.callback and new function

- Put it after app.layout. ddlXBarPlot parameter will be filled with ddl-x-bar-plot value, and the value that returned by the update_bar_graph function will fill the figure property of the barPlot



```
@app.callback(
    dash.dependencies.Output('barPlot', 'figure'),
    [dash.dependencies.Input('ddl-x-bar-plot', 'value')])
def update_bar_graph(ddlXBarPlot):
    return {
                'data': [
                    go.Bar(
                        x=dfTips[ddlXBarPlot],
                        y=dfTips['tip'],
                        text=dfTips['size'],
                        opacity=0.7,
                        name='Tip'
                    ),
                    go.Bar(
                        x=dfTips[ddlXBarPlot],
                        y=dfTips['total_bill'],
                        text=dfTips['size'],
                        opacity=0.7,
                        name='Total Bill'
                    )
                ],
                'layout': go.Layout(
                    xaxis={'title': ddlXBarPlot.capitalize()},
                    yaxis={'title': 'US$'},
                    margin={'l': 40, 'b': 40, 't': 10, 'r': 10},
                    legend={'x': 0, 'y': 1},
                    hovermode='closest'
                )
    };
```

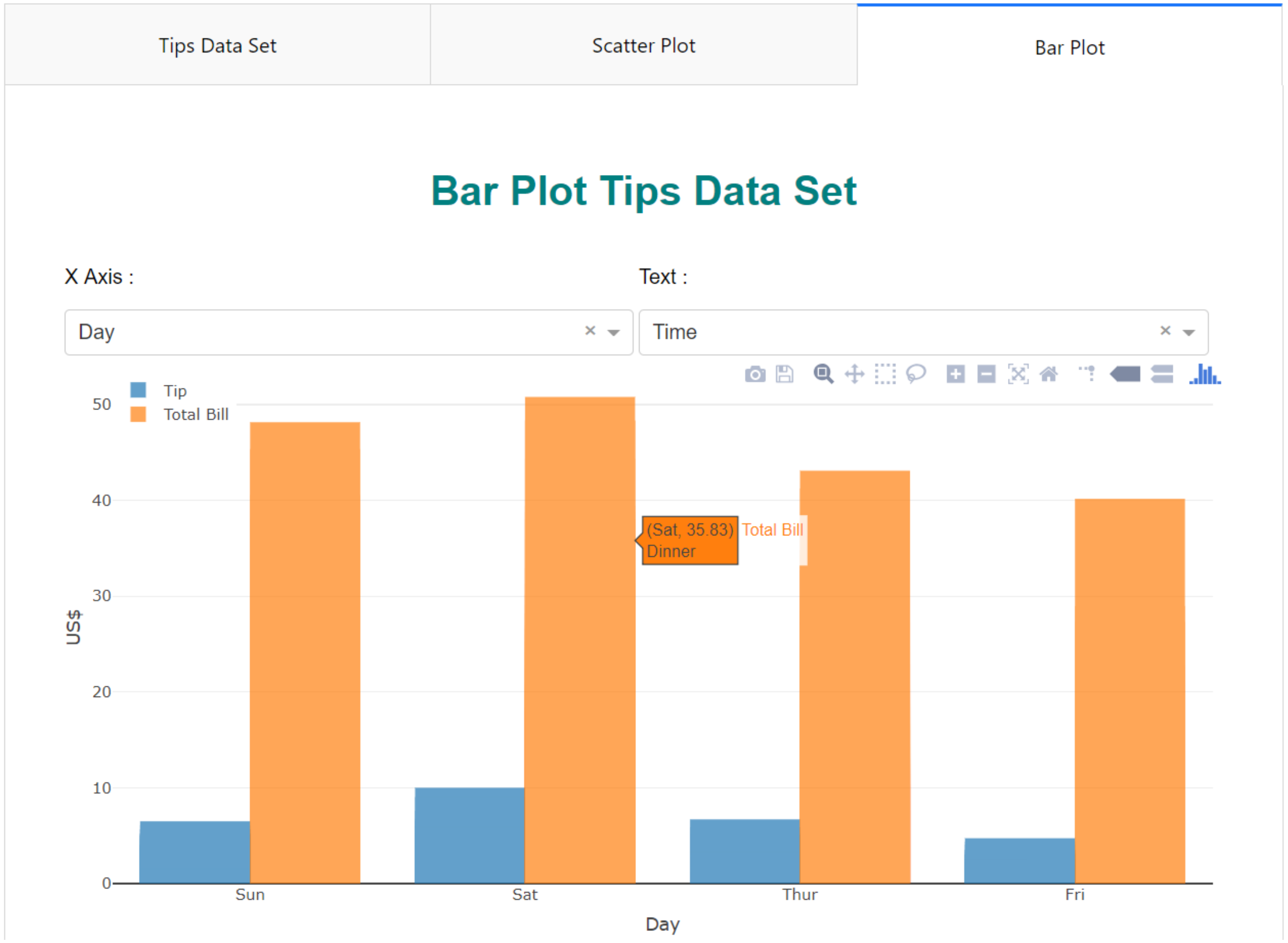


Bar Plot Tips Data Set

Purwadhika
Startup and Coding School

Multi Input Callback

- Challenge! make it like this!



Bar Plot Tips Data Set

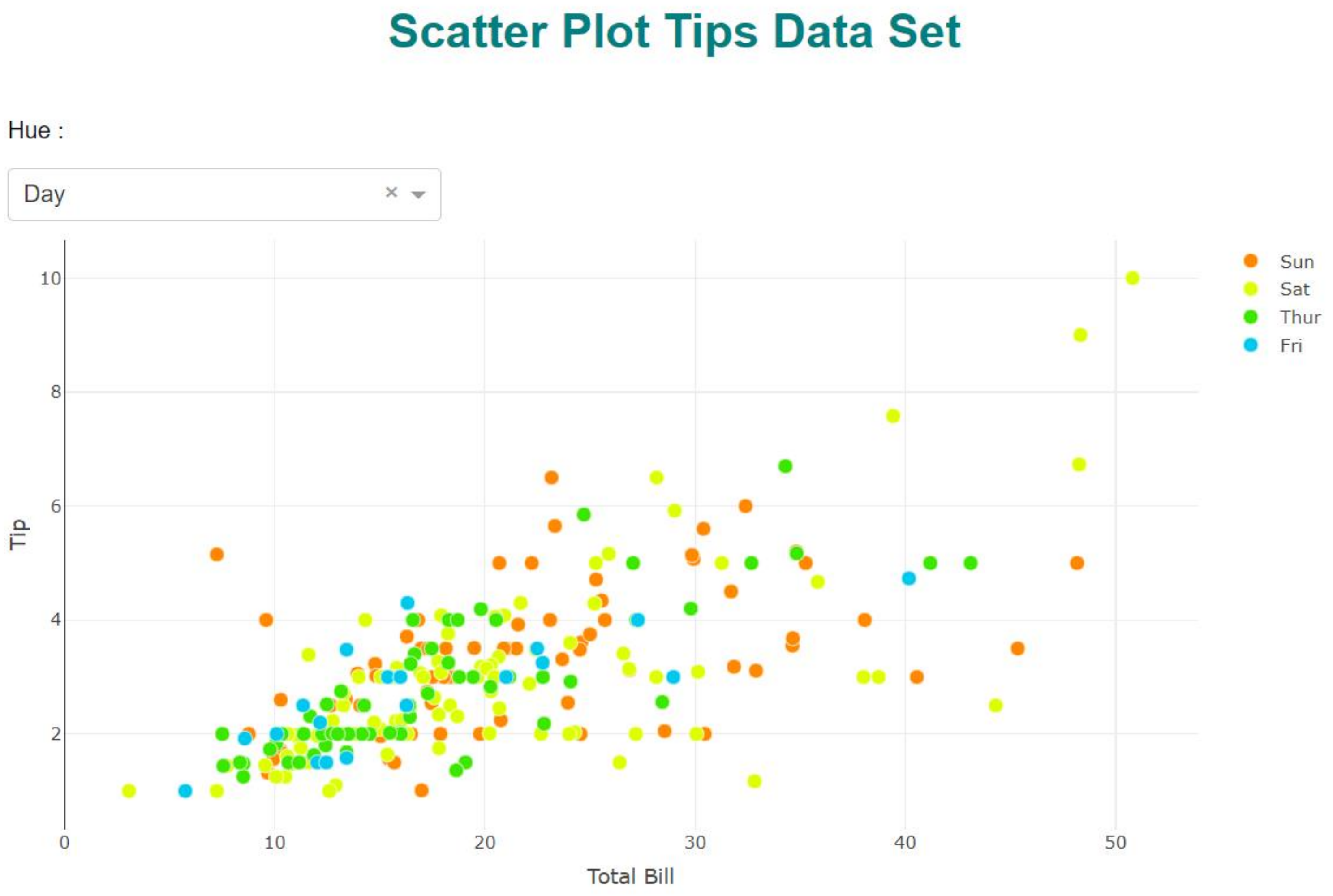
X Axis : Day

Text : Time

Purwadhika
Startup and Coding School

Multi Output Callback

- Challenge! make it like this!



Scatter Plot Tips Data Set
Hue :
Day
Sun
Sat
Thur
Fri
Tip
Total Bill
0
10
20
30
40
50
2
4
6
8
10