Introduction to dash

A simple example

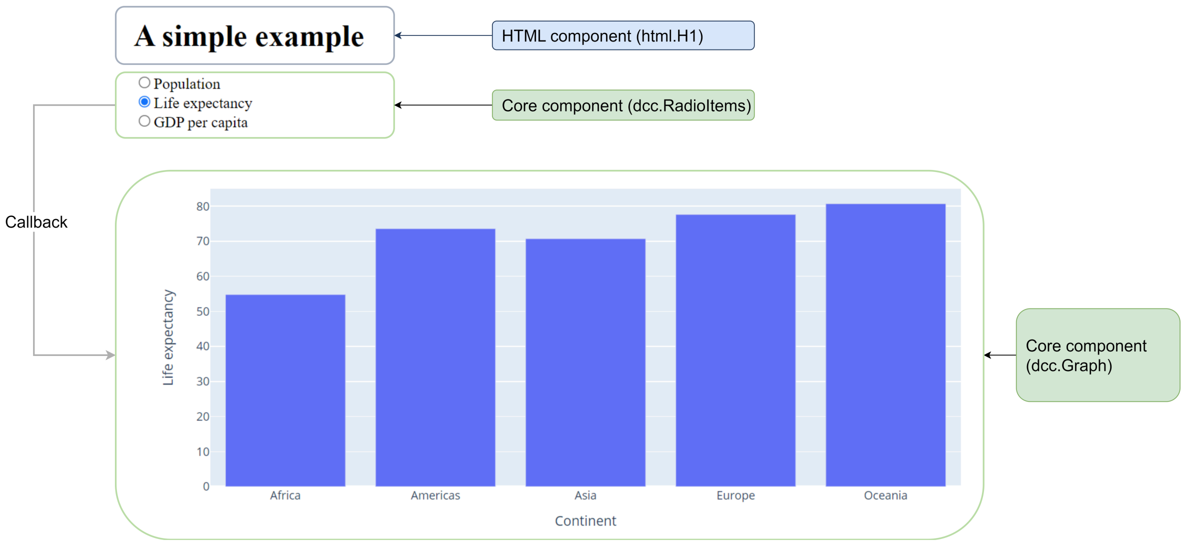
In [ ]:
```
import pandas as pd
from dash import Dash, html, dcc, callback, Input, Output
import plotly.express as px
```

In [ ]:
```
# Load data
info = pd.read_csv('countries.csv')

# Compute average features per continent
df = info.groupby('Continent')[['Population', 'Life expectancy', 'GDP per capita']].mean()
```

In [ ]:
```
# Initialize the app
app = Dash()

# App layout
app.layout = html.Div([

    # Add title
    html.H1('A simple example'),

    # Add radio items
    dcc.RadioItems(options=['Population', 'Life expectancy', 'GDP per capita'],
                   value='Life expectancy', id='radio'),

    # Add graph
    dcc.Graph(figure={}, id='graph')
])

# Add a callback to implement the user interaction
@callback(
    Output(component_id='graph', component_property='figure'),
    Input(component_id='radio', component_property='value')
)
def update_graph(selected_option):

    # Create bar plot
    fig = px.bar(df, x=df.index, y=selected_option)

    return fig

# Run the app
app.run(debug=True, use_reloader=False)
```

Components of a Dash app

- HTML Components: headings, text, etc.
- Dash Core components: controls, graphs, tables
- Callback functions: functions to implement user interaction



1) How to add components to your Dash app

Adding HTML components

- Headings (html.H1)
- Sections (html.Div)
- Text (html.P)
- Links (html.A)
- ...

In [ ]:
```
# Initialize the app
app = Dash()

# Define app layout
app.layout = html.Div([

    # Add a title
    html.H1('Hello Dash'),

    # Add a section (div=division)
    html.Div([

        # Add a paragraph
        html.P('Dash converts Python classes into HTML'),

        # Add another paragraph
        html.P("This conversion happens behind the scenes by Dash's JavaScript front-end")
    ]),

    # Add a link
    html.A('All dash HTML components', href='https://dash.plotly.com/dash-html-components')
])

# Run the app
app.run(debug=True, use_reloader=False)
```

Adding Dash core components

- Dropdown menus (dcc.Dropdown)
- Sliders (dcc.Slider)
- Checkboxes (dcc.Checkboxes)
- Radio Items (dcc.RadioItems)
- ...

In [ ]:
```
# Initialize the app
app = Dash()

# Define app layout
```

```
app.layout = html.Div([

    # Add a dropdown menu
    dcc.Dropdown(['New York', 'Bern', 'Thun']),

    # Add a line break
    html.Br(),

    # Add a slider
    dcc.Slider(0, 10, 1, value=5),

    # Add a line break
    html.Br(),

    # Add checkboxes
    dcc.Checklist(['Contract A', 'Contract B', 'Contract C']),

    # Add a line break
    html.Br(),

    # Add radio items
    dcc.RadioItems(['Option A', 'Option B', 'Option C'])


])

# Run the app
app.run(debug=True, use_reloader=False)
```

2) How to add interactivity using callbacks

A callback function consists of

- One or multiple Inputs
- One or multiple Outputs
- A Python function

In [ ]:
```
# Initialize the app
app = Dash()

# Define app layout
app.layout = html.Div([

    # Add a title
    html.H2('Select an option to see callbacks in action.'),

    # Add checkboxes
    dcc.Checklist(['Bern', 'Lugano', 'Zurich'], value=['Bern'], id='checkboxes'),

    # Add a line break
    html.Br(),

    # Add a paragraph that will display the selected options
    html.P(id='output_field'),

])

# Add a callback to implement the user interaction
@callback(
    Output(component_id='output_field', component_property='children'),
    Input(component_id='checkboxes', component_property='value')
)
def update_output_div(selected_options):

    # Initialize the output string
    output_string = 'You have selected the following options: '

    # Add the selected options to the output string
    for option in selected_options:
        output_string += option + ' '

    return output_string


# Run the app
app.run(debug=True, use_reloader=False)
```

3) How to style your app

You can use CSS to style your app

In [ ]:
```
# Initialize the app
app = Dash()

# Define app layout
app.layout = html.Div([

    # Add title
    html.H1('This is my title',
            style={'textAlign': 'center',
                   'color': 'blue',
                   'fontFamily': 'Arial, sans-serif',
                   'padding': 40})

])

# Run the app
app.run(debug=True, use_reloader=False)
```