# Dash Components

### Objectives

After completing the lab you will be able to:

- Create a dash application layout
- Add HTML H1, P, and Div components
- Add core graph component
- Add multiple charts

**Estimated time needed:** 30 minutes

### Dataset Used

Airline Reporting Carrier On-Time Performance dataset from Data Asset eXchange

# About Skills Network Cloud IDE

This Skills Network Labs Cloud IDE (Integrated Development Environment) provides a hands-on environment in your web browser for completing course and project related labs. It utilizes Theia, an open-source IDE platform, that can be run on desktop or on the cloud.
So far in the course you have been using Jupyter notebooks to run your python code. This IDE provides an alternative for editing and running your Python code. In this lab you will be using this alternative Python runtime to create and launch your Dash applications.

### Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persisted. When you launch the Cloud IDE, you are presented with a 'dedicated computer on the cloud' exclusively for you. This is available to you as long as you are actively working on the labs.

Once you close your session or it is timed out due to inactivity, you are logged off, and this 'dedicated computer on the cloud' is deleted along with any files you may have created, dowloaded or installed. The next time you launch this lab, a new environment is created for you.

*If you finish only part of the lab and return later, you may have to start from the beginning. So, it is a good idea to plan to your time accordingly and finish your labs in a single session.*

# Let's start creating dash application

### Goal

Create a dashboard that displays the percentage of flights running under specific distance group. Distance group is the distance intervals, every 250 miles, for flight segment. If the flight covers to 500 miles, it will be under distance group 2 (250 miles + 250 miles).

### Expected Output

Below is the expected result from the lab. Our dashboard application consists of three components:

- Title of the application
- Description of the application
- Chart conveying the proportion of distance group by month

### To do:

1. Import required libraries and read the dataset
2. Create an application layout
3. Add title to the dashboard using HTML H1 component
4. Add a paragraph about the chart using HTML P component
5. Add the pie chart above using core graph component
6. Run the app

# Get the tool ready

- Install python packages required to run the application. Copy and paste the below command to the terminal.

```
1. 1
1. python3 -m pip install packaging
```

Copied!

```
1. 1
1. python3 -m pip install pandas dash
```

Copied!

```
1. 1
1. pip3 install httpx==0.20 dash plotly
```

Copied!

- Create a new python script, by clicking on the menu bar and selecting **File**->**New File**, as in the image below.
- Provide the file name as `dash_basics.py`
- Open a new terminal, by clicking on the menu bar and selecting **Terminal**->**New Terminal**, as in the image below.
- Now, you have script and terminal ready to start the lab.

# TASK 1 - Data Preparation

Let's start with

- Importing necessary libraries
- Reading and sampling 500 random data points
- Get the chart ready

Copy the below code to the `dash_basics.py` script and review the code.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

```
# Import required packages
import pandas as pd
import plotly.express as px
import dash
import dash_html_components as html
import dash_core_components as dcc

# Read the airline data into pandas dataframe
airline_data =  pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DV0101EN-S
                            encoding = "ISO-8859-1",
                            dtype={'Div1Airport': str, 'Div1TailNum': str,
                                   'Div2Airport': str, 'Div2TailNum': str})

# Randomly sample 500 data points. Setting the random state to be 42 so that we get same result.
data = airline_data.sample(n=500, random_state=42)

# Pie Chart Creation
fig = px.pie(data, values='Flights', names='DistanceGroup', title='Distance group proportion by flights')
```

Copied!

# TASK 2 - Create dash application and get the layout skeleton

Next, we create a skeleton for our dash application. Our dashboard application has three components as seen before:

- Title of the application
- Description of the application
- Chart conveying the proportion of distance group by month

Mapping to the respective Dash HTML tags:

- Title added using `html.H1()` tag
- Description added using `html.P()` tag
- Chart added using `dcc.Graph()` tag

Copy the below code to the `dash_basics.py` script and review the structure.

*NOTE*: Copy below the current code

```
1
2
3
4
5
6
7
8
9
10
11
12
13
```

```
14. 14
15. 15
16. 16
```

```
 1. # Create a dash application
 2. app = dash.Dash(__name__)
 3. 
 4. # Get the layout of the application and adjust it.
 5. # Create an outer division using html.Div and add title to the dashboard using html.H1 component
 6. # Add description about the graph using HTML P (paragraph) component
 7. # Finally, add graph component.
 8. app.layout = html.Div(children=[html.H1(),
 9.                                 html.P(),
10.                                 dcc.Graph(),
11. 
12.                     ])
13. 
14. # Run the application
15. if __name__ == '__main__':
16.     app.run_server()
```

Copied!

# TASK 3 - Add the application title

Update the `html.H1()` tag to hold the application title.

- Application title is `Airline Dashboard`
- Use style parameter provided below to make the title `center` aligned, with color code `#503D36`, and font-size as `40`

```
1. 1
```

```
1. 'Airline Dashboard',style={'textAlign': 'center', 'color': '#503D36', 'font-size': 40}
```

Copied!

After updating the `html.H1()` with the application title, the `app.layout` will look like:

# TASK 4 - Add the application description

Update the `html.P()` tag to hold the description of the application.

- Description is `Proportion of distance group (250 mile distance interval group) by flights.`
- Use style parameter to make the description `center` aligned and with color `#F57241`.

```
1. 1
```

```
1. 'Proportion of distance group (250 mile distance interval group) by flights.', style={'textAlign':'center', 'color': '#F57241'}
```

Copied!

After updating the `html.H1()` with the application title, the `app.layout` will look like:

# TASK 5 - Update the graph

Update `figure` parameter of `dcc.Graph()` component to add the pie chart. We have created pie chart and assigned it to `fig`. Let's use that to update the `figure` parameter.

```
1. 1
```

```
1. figure=fig
```

Copied!

After updating the `dcc.Graph()` with the application title, the `app.layout` will look like:

Before running the application, save the file by clicking on **File -> Save** from the menu bar.

You can Refer to the entire python code here

```
 1. 1
 2. 2
 3. 3
 4. 4
 5. 5
 6. 6
 7. 7
 8. 8
 9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
```

```
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
```

```
 1.
 2. # Import required packages
 3. import pandas as pd
 4. import plotly.express as px
 5. import dash
 6. import dash_html_components as html
 7. import dash_core_components as dcc
 8.
 9. # Read the airline data into pandas dataframe
10. airline_data =  pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DV0101EN-S
11.                             encoding = "ISO-8859-1",
12.                             dtype={'Div1Airport': str, 'Div1TailNum': str,
13.                                    'Div2Airport': str, 'Div2TailNum': str})
14.
15. # Randomly sample 500 data points. Setting the random state to be 42 so that we get same result.
16. data = airline_data.sample(n=500, random_state=42)
17.
18. # Pie Chart Creation
19. fig = px.pie(data, values='Flights', names='DistanceGroup', title='Distance group proportion by flights')
20.
21. # Create a dash application
22. app = dash.Dash(__name__)
23.
24. # Get the layout of the application and adjust it.
25. # Create an outer division using html.Div and add title to the dashboard using html.H1 component
26. # Add description about the graph using HTML P (paragraph) component
27. # Finally, add graph component.
28. app.layout = html.Div(children=[html.H1('Airline Dashboard', style={'textAlign': 'center', 'color': '#503D36', 'font-size': 40}),
29.                                 html.P('Proportion of distance group (250 mile distance interval group) by flights.', style={'textA
30.                                 dcc.Graph(figure=fig),
31.
32.                     ])
33.
34. # Run the application
35. if __name__ == '__main__':
36.     app.run_server()
```

Copied!

# TASK 6 - Run the application

- Run the python file using the following command in the terminal

```
1. 1
```

```
1. python3 dash_basics.py
```

Copied!

- Observe the port number shown in the terminal.

- Click on the `Launch Application` option from the side menu bar.Provide the port number and click `OK`

The app will open in a new browser tab like below:

**Congratulations, you have successfully created your first dash application!**

# Exercise : Practice Tasks

You will practice some tasks to update the dashboard.

1. Change the title to the dashboard from “Airline Dashboard” to “Airline On-time Performance Dashboard” using HTML H1 component and font-size as 50.

   ▼ Answer

   html.H1(‘Airline On-time Performance Dashboard’,style={‘textAlign’: ‘center’, ‘color’: ‘#503D36’, ‘font-size’: 50}),

2. Save the above changes and relaunch the dashboard application to see the updated dashboard title.

   ▼ Answer

   Click on file –> save file.Then go to terminal and Run the command python3 dash_basics.py to open the updated file again and relaunch the application by entering the port number.The updated dashboard title will be seen.

3. Write a command to stop the running app in the terminal

▼ Answer

Press `ctrl+c` inside the terminal to stop the dash application.

## Author

Saishruthi Swaminathan

## Changelog

| Date | Version | Changed by | Change Description |
|---|---|---|---|
| 05-07-2021 | 1.1 | Saishruthi | Initial version created |
| 24-08-2022 | 1.2 | Pratiksha | Updated instructions |
| 29-08-2022 | 1.3 | Pratiksha Verma | Updated Screenshot |
| 06-07-2023 | 1.4 | Dr. Pooja | Code update |