

Lesson 1 - Introduction to Flet

Flet is a Python library that allows you to quickly build interactive web apps with very little code. It is similar to tools like Streamlit and Dash, but it focuses on simplicity and real-time interaction.

Here are the basics of using Flet to create web apps:

1. Installing Flet

To start, you need to install Flet in your Python environment:

```
pip install flet
```

2. Creating a Simple App

Once Flet is installed, you can start creating a simple interactive app. Here's a basic example to get started:

```
import flet as ft

def main(page):
    page.add(ft.Text("Hello, Flet!"))

ft.app(target=main)
```

Key points:

- `flet.app()`: This is the main function to start your Flet application. You pass it a function (like `main()`) that defines the layout and logic of your app.
- `page.add()`: This adds widgets (UI elements) to the page. In this example, we are adding a simple `Text` widget that says "Hello, Flet!".
- `ft.Text()`: This creates a text widget. You can customize the text with different properties like size, color, and alignment.

3. Widgets in Flet

Flet provides a variety of UI widgets for building interactive apps. Some of the common ones are:

- **Text**: Displays text.
- **Button**: A clickable button.
- **TextField**: An input box where users can type text.
- **Column/Row**: Used to arrange widgets vertically or horizontally.

Here's a more complex example that shows how to use some of these widgets:

```
import flet as ft

def main(page):
    # Create a button and text input
    text_box = ft.TextField(label="Enter your name")
    button = ft.ElevatedButton(text="Say Hello",
                               on_click=lambda e: page.add(ft.Text(f"Hello, {text_box.value}!")))

    # Add widgets to the page
    page.add(text_box, button)

ft.app(target=main)
```

Key points:

- **TextField**: The `TextField` widget allows users to type something in it. You can access the text using the `.value` attribute.
- **Button with on_click**: The `Button` widget triggers an event (like clicking) and calls a function, where you can define the behavior you want (in this case, showing a personalized greeting).

4. Layout and Styling

You can use `Column`, `Row`, `Stack`, and `Container` widgets to organize the layout and control how the widgets are arranged.

For example, a `Column` widget arranges elements vertically:

```

import flet as ft

def main(page):
    page.add(
        ft.Column(
            [
                ft.Text("Welcome to Flet!"),
                ft.ElevatedButton(text="Click Me", on_click=lambda e:
page.add(ft.Text("You clicked the button!"))),
            ]
        )
    )

ft.app(target=main)

```

5. Interactive Updates

One of the best features of Flet is its ability to interact in real-time. Any changes you make to widgets can instantly be reflected in the UI.

For example, you can update the text dynamically after a button click:

```

import flet as ft

def main(page):
    text = ft.Text("Click the button to change me!")

    def change_text(e):
        text.value = "You changed the text!"
        page.update() # Update the page to reflect the change

    button = ft.RaisedButton(text="Change Text",
                             on_click=change_text)

    page.add(text, button)

```

```
ft.app(target=main)
```

6. Running the App

To run the app, you simply execute your Python script in the terminal:

```
python my_flet_app.py
```

This will open a local server (usually on `localhost:8550`), and you can view your app in your web browser.

Summary of Key Concepts:

1. **Widgets:** Use `Text`, `Button`, `TextBox`, `Column`, `Row`, etc., to build your UI.
2. **Events:** Attach event handlers (like `on_click`) to buttons or other widgets for interactivity.
3. **Layouts:** Use `Column`, `Row`, or `Container` to organize the widgets.
4. **Real-time updates:** Use `page.update()` to refresh the page when something changes.
5. **Running:** `ft.app(target=main)` launches your app in the browser.

Files created in this lesson:

```
hello_flet.py
interactive_updates.py
layout_and_styling_column.py
layout_and_styling_column_02.py
layout_and_styling_container.py
layout_and_styling_row.py
layout_and_styling_stack.py
widgets_in_flet.py
```

Next Steps:

- Exploring widgets
- Exploring layout options.
- Diving deep into themes
- Diving deep into data tables

- Diving deep into user authentication.
- Deploy your app on the web using Flet's hosting services.