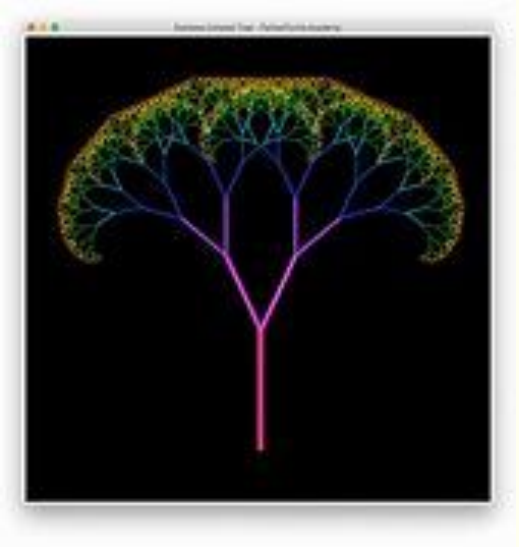
Unit:3; Class:6

Introduction to Turtle & Streamlit

Instructor: Musfique Ahmed

Turtle Graphics in Python

- Turtle is a Python module used for drawing graphics.
- It provides an easy way to create shapes, patterns, and animations.
- Inspired by the LOGO programming language.



Turtle

Setting Up Turtle:

```
import turtle

screen = turtle.Screen()  # Create a screen
t = turtle.Turtle()  # Create a turtle object
t.forward(100)  # Move forward 100 pixels
t.right(90)  # Turn right by 90 degrees
t.forward(100)  # Move forward again
turtle.done()  # End the turtle graphics
```

💡 **Practice:** Modify the code to draw a square.

Basic Turtle Commands

Command	Description
`forward(x)`	Move forward by `x` pixels
`backward(x)`	Move backward by `x` pixels
`right(angle)`	Rotate right by given degrees
`left(angle)`	Rotate left by given degrees
`penup()`	Lift pen (move without drawing)
`pendown()`	Put pen down (start drawing)

💡 **Practice:** Try drawing a triangle using the `left()` or `right()` functions.

Drawing Shapes with Loops

- We can use loops to draw repeated patterns efficiently.

```
import turtle

t = turtle.Turtle()
for _ in range(4):  # Loop 4 times
    t.forward(100)
    t.right(90)  # Turn 90 degrees

turtle.done()
```

💡 **Practice:** Modify the loop to draw a hexagon.

Color & Speed in Turtle

- `turtle.color("red")` → Changes pen color.
- `turtle.speed(3)` → Adjusts drawing speed (1: slow, 10: fast).

```
import turtle

t = turtle.Turtle()
t.speed(0)  # Max speed

for i in range(100):
    t.forward(i * 2)
    t.right(91)  # Experiment with different angles

turtle.done()
```

💡 **Practice:** Change the color and speed.

Fun Example - Drawing a Spiral


```
import turtle

t = turtle.Turtle()
t.speed(0)  # Max speed

for i in range(100):
    t.forward(i * 2)
    t.right(91)  # Experiment with different angles

turtle.done()
```

💡 **Practice:** Change the angle to see different spiral designs.

Where to Use Turtle?

- **Education:** Teaching kids and beginners programming concepts.
- **Game Development:** Simple graphical games and puzzles.
- **Mathematics & Geometry:** Drawing complex patterns, fractals, and visualizing algorithms.
- **Simulation:** Creating animations for different real-world scenarios.

What is Streamlit?

- Streamlit is an open-source Python library for creating web apps.
- No need for HTML/CSS—just Python!
- Great for **data science dashboards, ML models, and interactive apps.**

Setting Up Streamlit:

Installation:

```
pip install streamlit
```

Running a Streamlit App:

```
streamlit run my_app.py
```

Where to Use Streamlit?

- **Data Science Dashboards:** Displaying interactive charts, tables, and ML results.
- **Machine Learning Apps:** Creating interactive ML models for predictions.
- **Automation Tools:** Building apps for data input, report generation, and automation tasks.
- **Prototyping Web Apps:** Quickly building and sharing Python-based web apps without complex frontend coding.

💡 **Practice:** Install Streamlit and run a sample app.

Writing Your First Streamlit App


```
import streamlit as st

st.title("Welcome to Streamlit!")
st.write("This is a simple web app using Streamlit.")
```

💡 **Practice:** Modify the text and title.

Adding User Input in Streamlit:

```
import streamlit as st

name = st.text_input("Enter your name:")
st.write("Hello", name)
```

💡 **Practice:** Ask users for their favorite color and display it.

Adding Interactive Widgets

Widget	Description
`st.text_input("Enter text")`	Input field for text
`st.number_input("Enter number")`	Numeric input field
`st.text_area("Write here")`	Multiline text input
`st.selectbox("Choose", ["A", "B", "C"])`	Dropdown menu
`st.checkbox("Check this")`	Checkbox for selection
`st.radio("Select one", ["Option 1", "Option 2"])`	Radio buttons
`st.button("Click Me")`	Clickable button
`st.slider("Pick a number", 1, 100)`	Slider for number selection
`st.file_uploader("Upload file")`	Allows file upload
`st.date_input("Pick a date")`	Select a date
`st.progress(50)`	Displays a progress bar
`st.image("image.png")`	Displays an image
`st.video("video.mp4")`	Embeds a video

💡 **Practice:** Try using multiple widgets in a simple app.

Displaying Charts with Streamlit


```python
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

data = {"Category": ["A", "B", "C"], "Values": [10, 20, 30]}
df = pd.DataFrame(data)

st.bar_chart(df)
```

💡 **Practice:** Modify the data to display different values.

How to Deploy a Streamlit App

Install Streamlit & Create Your App

```
pip install streamlit
```

1. Save your script as `app.py`.
2. **Run Locally for Testing**
   ```
   streamlit run app.py
   ```
3. **Deploy on Streamlit Cloud (Free)**
   - Go to **Streamlit Community Cloud**
   - **Sign in with GitHub** and create a repository.
   - **Upload your app script (`app.py`)**.
   - Click **New App** → **Select your repo** → **Deploy**.
4. **Deploy on Other Platforms** (Optional)
   - **Heroku:** Use `Procfile` with `streamlit run app.py`.
   - **AWS, Google Cloud:** Deploy using Docker containers.

💡 **Practice:** Deploy a basic app on Streamlit Cloud.

Project Idea - Build a Simple Calculator

Requirements:

1. Use Streamlit for UI.
2. Take two numbers as input.
3. Perform **addition, subtraction, multiplication, division.**

💡 **Practice:** Implement this project using `st.number_input()` and `st.button()`.

Summary & Next Steps

✅ **Turtle Graphics:**

- Used for drawing and animation.
- Functions: `forward()`, `right()`, `color()`, `speed()`.

✅ **Streamlit Web Apps:**

- Allows Python-based UI without HTML.
- Widgets: `st.text_input()`, `st.button()`, `st.slider()`.

Next Steps:

- Try making an interactive drawing app using Turtle.
- Build a data dashboard using Streamlit.

Thank You

Do the Quiz Please, you have 10 minutes to do that!