learnwithbhawana

Indian Flag 🟠⚪🟢 with Rotating Ashoka Chakra (Code Explanation)

1. Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import time
import sys
```

- **numpy (np)** – helps in doing math calculations easily, especially with angles and circles.
- **matplotlib.pyplot (plt)** – used to draw shapes, colours, and the flag on the screen.
- **matplotlib.animation** – lets us create animations (moving images).
- **time** – lets us control timing, for example, to make typing effects.
- **sys** – used for printing smoothly in the terminal.

2. Chakra Parameters

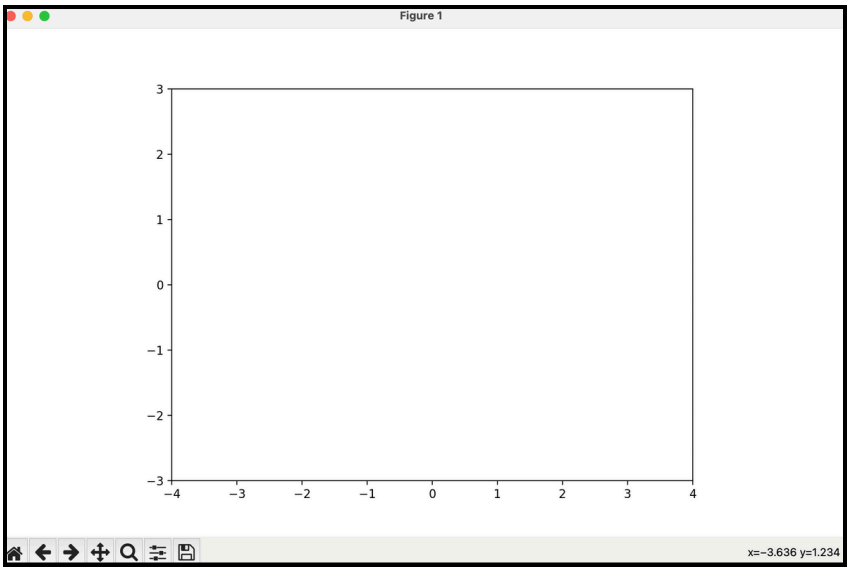
```
# Chakra parameters
num_spokes = 24
chakra_radius = 1.0
spoke_angles = np.linspace(0, 2 * np.pi, num_spokes, endpoint=False)
```

- **num_spokes** – The Ashoka Chakra has 24 spokes.
- **chakra_radius** – Size of the Chakra circle (1.0 units).
- **spoke_angles** – Creates 24 equally spaced angles from **0** to **360 degrees** (in radians) so each spoke is at the correct position.

3. Create Figure and Axis

```
# Create figure and axis
fig, ax = plt.subplots(figsize=(10, 6))
ax.set_xlim(-4, 4)
ax.set_ylim(-3, 3)
ax.set_aspect('equal')
ax.axis('off')
```

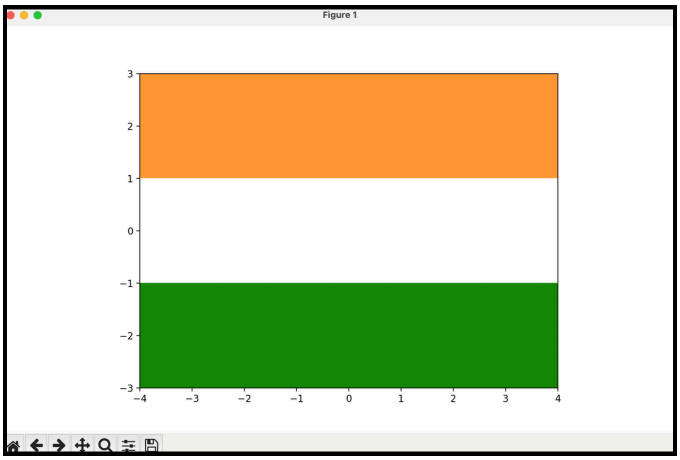
- **plt.subplots** – Creates the canvas (figure) and a drawing area (axis).
- **figsize** – Size of the figure in inches (10 wide, 6 tall).
- **set_xlim / set_ylim** – Sets the visible area of the drawing.
- **set_aspect('equal')** – Ensures circles don't look squished.
- **axis('off')** – Hides x-axis and y-axis lines for a clean flag look.



4. Draw the Flag Stripes

```python
# Draw the Indian flag stripes
def draw_flag():
    ax.fill_between([-4, 4], 1, 3, color='#FF9933')     # Saffron
    ax.fill_between([-4, 4], -1, 1, color='white')      # White
    ax.fill_between([-4, 4], -3, -1, color='#138808')   # Green
```

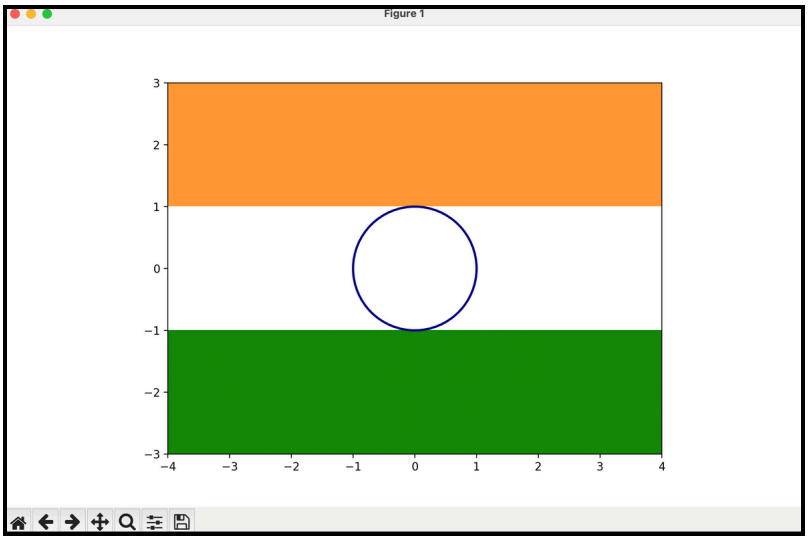
- **fill_between** – Colors a horizontal stripe between two y-values.
- First stripe: from y=1 to y=3 → **Saffron**.
- Second stripe: from y=-1 to y=1 → **White**.
- Third stripe: from y=-3 to y=-1 → **Green**.



5. Draw the Chakra Outline

```python
# Draw Chakra outline
chakra_lines = []
def init_chakra():
    circle = plt.Circle((0, 0), chakra_radius, color='#000080', fill=False, linewidth=2)
    ax.add_patch(circle)
```

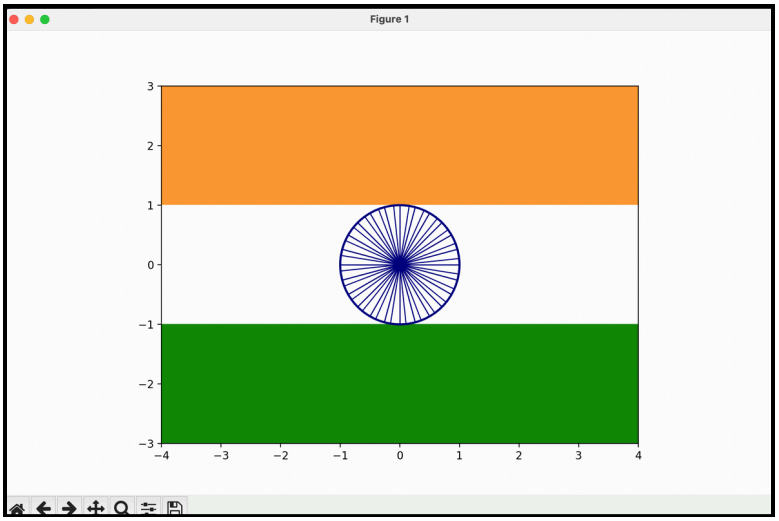
- **chakra_lines** – Will store the spokes so we can move them later.
- **plt.Circle** – Draws a circle at center (0,0) with given radius and navy blue color.
- **add_patch** – Adds the circle shape to our figure.

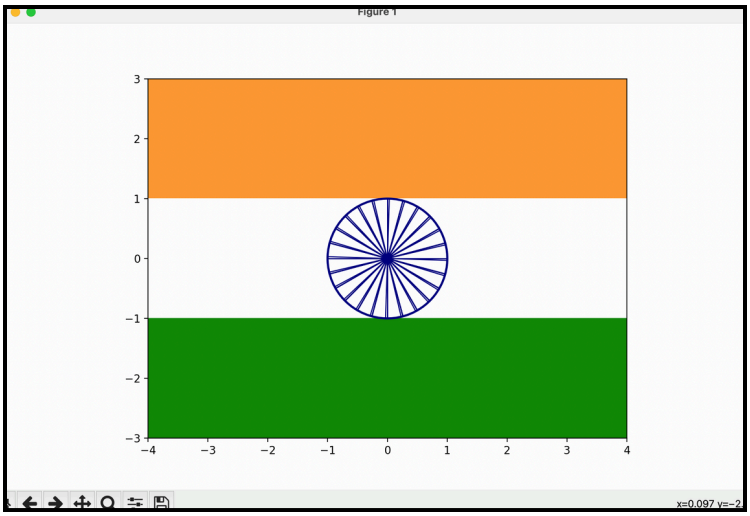


6. Animation Update Function

```python
# Animation update function
def update(frame):
    rotation = frame * (2 * np.pi / 100)
    for i, angle in enumerate(spoke_angles):
        rotated_angle = angle + rotation
        x = chakra_radius * np.cos(rotated_angle)
        y = chakra_radius * np.sin(rotated_angle)
        if frame == 0:
            line, = ax.plot([0, x], [0, y], color='#000080', lw=1)
            chakra_lines.append(line)
        else:
            chakra_lines[i].set_data([0, x], [0, y])
    return chakra_lines
```

- **frame** – In animation, each step is called a “frame”. The value increases as animation plays.
- **rotation** – Decides how much the Chakra turns each frame.
- **enumerate(spoke_angles)** – Loops through all spoke starting angles.
- **rotated_angle** – Adds rotation to the spoke’s original position.
- **x, y** – Endpoints of the spoke line after rotation (using cosine and sine for coordinates).
- **if frame == 0** – In the first frame, we create and draw spokes.
- **else** – For later frames, just update spoke positions.
- **return chakra_lines** – Returns updated lines for smooth animation.





7. Putting it Together

```
# Draw flag and chakra
draw_flag()
init_chakra()
```

- **Draws the stripes and adds the Chakra outline before starting the animation.**

8. Animate the Chakra

```
# Animate Chakra
ani = animation.FuncAnimation(fig, update, frames=200, interval=50, blit=True)
```

- **FuncAnimation** – Runs update again and again to make it look like the spokes are spinning.
- **frames=200** – Number of animation steps.
- **interval=50** – Time between frames in milliseconds (50 ms).
- **blit=True** – Optimizes animation speed.

9. Typing Effect Message

```
# Typing effect in terminal
message = "🟠⚪🟢 Proud to be Indian 🟠⚪🟢 \n🟠⚪🟢 Happy Independence Day 🟠⚪🟢\n"

for char in message:
    print(char, end='', flush=True)
    time.sleep(0.1)
```

- **message** – Text to show in terminal with emojis.
- **for char in message** – Loops through each character one by one.
- **print(..., flush=True)** – Shows each character immediately instead of waiting.
- **time.sleep(0.1)** – Adds delay between each character for typing effect.

10. Show the Flag

```
plt.show()
```

- Displays the complete flag with spinning Chakra in a window.

```
Python_project/Visual_Basic /indian_flag.py"
🟠⚪🟢 Proud to be Indian 🟠⚪🟢
🟠⚪🟢 Happy Independence Day 🟠⚪🟢
```