

Basic Structure and Workflow

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

- Import: `import matplotlib.pyplot as plt`
- Simple Plotting:

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
```

```
plt.plot(x, y) # Create a line plot
plt.title("Sample Line Plot") # Add title
plt.xlabel("X-axis") # Label x-axis
plt.ylabel("Y-axis") # Label y-axis
plt.show() # Display the figure
```

Customization (Styles and Aesthetics)

Enhance readability with colors, markers, line styles, and legends.

Color, Marker, and LineStyle

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
```

```
# color: color, linestyle: type of line,
marker: point shape, markersize: size
plt.plot(x, y, color='red',
linestyle='--', marker='o',
markersize=8)
plt.title("Styled Plot")
plt.grid(True) # Show grid
plt.show()
```

Utilization of Legends

Identify multiple data series in a single plot.

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 3, 5, 7, 9]
```

```
plt.plot(x, y1, label='Group A')
plt.plot(x, y2, label='Group B')
plt.title("Graph with Legends")
plt.legend(loc='upper left') # Specify
legend position
plt.show()
```

Subplots (Multi-plot Layouts)

Arrange multiple graphs in one figure for comparative analysis.

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(0, 2 * np.pi, 100)
y_sin = np.sin(x)
y_cos = np.cos(x)
```

```
# Create subplots in 1 row and 2 columns
fig, axes = plt.subplots(1, 2,
figsize=(12, 4))
```

```
# First plot
axes[0].plot(x, y_sin, color='blue')
axes[0].set_title("Sine Function")
axes[0].set_xlabel("x")
axes[0].set_ylabel("sin(x)")
```

```
# Second plot
axes[1].plot(x, y_cos, color='orange')
axes[1].set_title("Cosine Function")
axes[1].set_xlabel("x")
axes[1].set_ylabel("cos(x)")
```

```
plt.tight_layout() # Automatically
adjust spacing
plt.show()
```

Saving Figures

Save generated plots as high-resolution image files.

```
import matplotlib.pyplot as plt
```

```
plt.plot([1, 2, 3], [4, 5, 6])
plt.title("File Saving Example")
```

```
# dpi: resolution setting, bbox_inches:
minimize margins
```

```
plt.savefig("my_visualization.png",
dpi=300, bbox_inches='tight')
print("Visualization saved to
my_visualization.png")
```