Matplotlib - Line Plots

Basic Line Plot

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

x = np.linspace(0, 2 * np.pi, 100)
y1 = np.sin(x)
y2 = np.cos(x)

plt.figure(figsize=(10, 6))
plt.plot(x, y1, label='sin(x)', linewidth=2)
plt.plot(x, y2, label='cos(x)', linewidth=2, linestyle='--')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Trigonometric Functions')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```

Multiple Lines with Different Styles

```
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.sin(x) * np.exp(-x/10)
y3 = np.cos(x)

plt.figure(figsize=(12, 6))
plt.plot(x, y1, 'b-', label='sin(x)', linewidth=2)
plt.plot(x, y2, 'r--', label='damped sin(x)', linewidth=2)
plt.plot(x, y3, 'g:', label='cos(x)', linewidth=3)

plt.xlabel('Time (s)', fontsize=12)
plt.ylabel('Amplitude', fontsize=12)
plt.title('Oscillating Functions', fontsize=14, fontweight='bold')
plt.legend(loc='upper right', fontsize=10)
plt.grid(True, alpha=0.3, linestyle='--')
plt.axhline(y=0, color='k', linestyle='-', linewidth=0.5)
plt.show()
```

Customized Plot with Markers

```
x = np.linspace(0, 5, 20)
y = x**2

plt.figure(figsize=(10, 6))
plt.plot(x, y, marker='o', markersize=8, linestyle='-',
         linewidth=2, color='purple', markerfacecolor='yellow',
         markeredgecolor='black', markeredgewidth=2,
         label='y = x²')

plt.xlabel('x', fontsize=12)
plt.ylabel('y', fontsize=12)
plt.title('Quadratic Function with Markers', fontsize=14)
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```

Subplot with Multiple Plots

```
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

x = np.linspace(0, 10, 100)

# Plot 1: Linear
axes[0, 0].plot(x, x, 'b-')
axes[0, 0].set_title('Linear: y = x')
axes[0, 0].grid(True, alpha=0.3)

# Plot 2: Quadratic
axes[0, 1].plot(x, x**2, 'r-')
axes[0, 1].set_title('Quadratic: y = x²')
axes[0, 1].grid(True, alpha=0.3)

# Plot 3: Cubic
axes[1, 0].plot(x, x**3, 'g-')
axes[1, 0].set_title('Cubic: y = x³')
axes[1, 0].grid(True, alpha=0.3)

# Plot 4: Exponential
axes[1, 1].plot(x, np.exp(x/5), 'm-')
axes[1, 1].set_title('Exponential: y = e^(x/5)')
axes[1, 1].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

Advanced Styling and Annotations

```
x = np.linspace(0, 10, 100)
y = np.sin(x) * np.exp(-x/10)

plt.figure(figsize=(12, 6))
plt.plot(x, y, 'b-', linewidth=2, label='Damped sine wave')

# Add vertical lines at peaks
peaks = x[np.where((y[1:-1] > y[0:-2]) & (y[1:-1] > y[2:]))[0] + 1]
for peak in peaks[:3]:
    plt.axvline(x=peak, color='r', linestyle='--', alpha=0.5)

# Add annotation
max_idx = np.argmax(y)
plt.annotate('Maximum', xy=(x[max_idx], y[max_idx]),
             xytext=(x[max_idx]+1, y[max_idx]+0.2),
             arrowprops=dict(arrowstyle='->', color='red'),
             fontsize=12, color='red')

# Customize appearance
plt.xlabel('x', fontsize=14, fontweight='bold')
plt.ylabel('y', fontsize=14, fontweight='bold')
plt.title('Damped Sine Wave with Annotations', fontsize=16)
plt.legend(loc='upper right', fontsize=12)
plt.grid(True, alpha=0.3, linestyle=':')
plt.xlim(0, 10)
plt.show()
```