Matplotlib

In [1]:

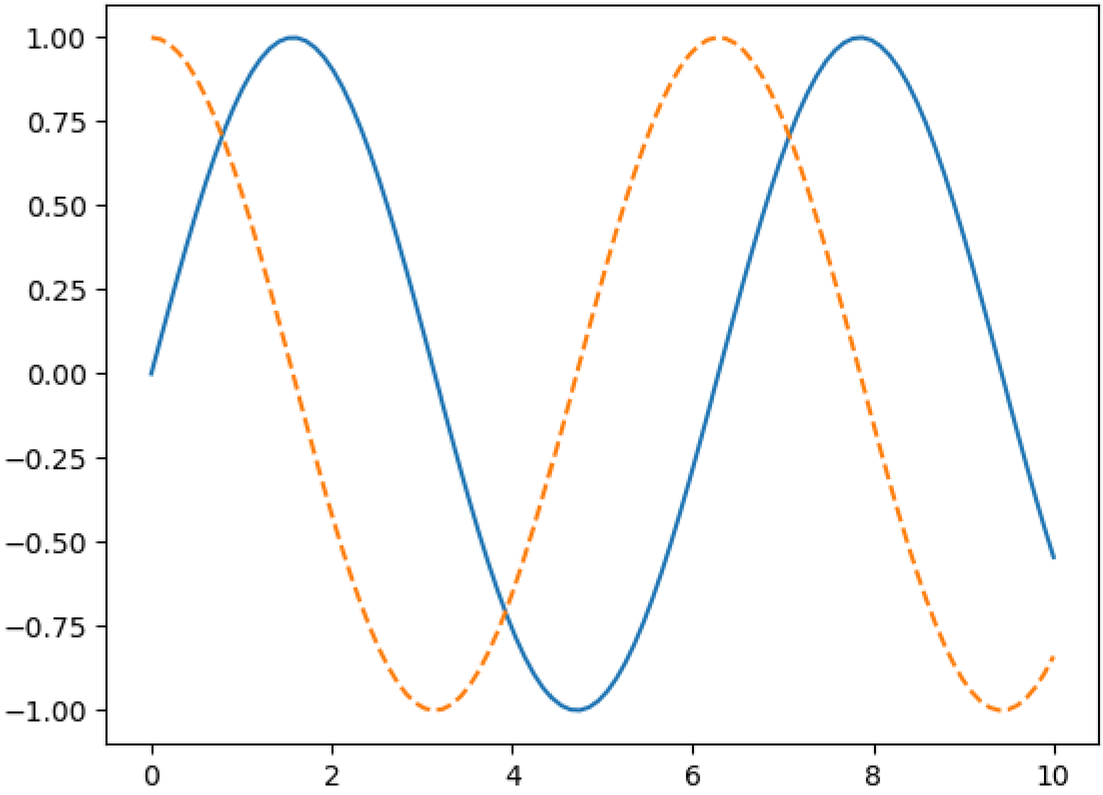
```
import numpy as np
import pandas as pd
```

In [2]:

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

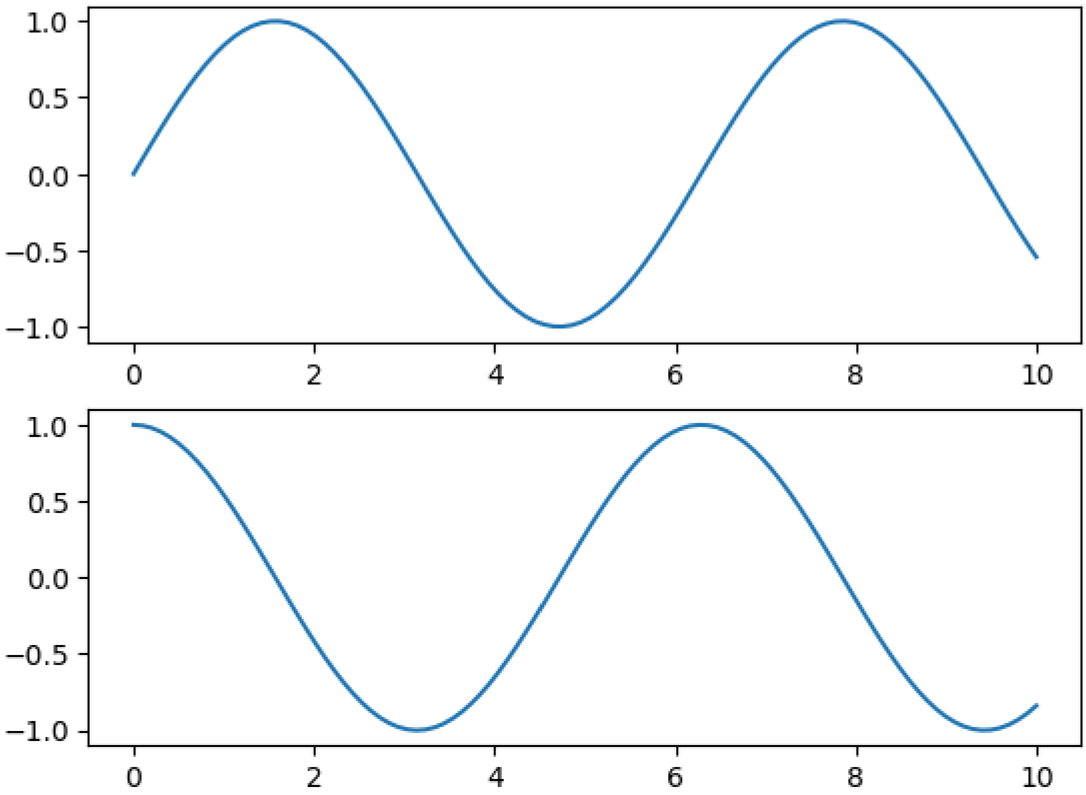
In [3]:

```
%matplotlib inline
x1 = np.linspace(0, 10, 100)

fig = plt.figure()
plt.plot(x1, np.sin(x1), '-')
plt.plot(x1, np.cos(x1), '--');
```



In [4]:

```
plt.figure()
plt.subplot(2, 1, 1)
plt.plot(x1, np.sin(x1))
plt.subplot(2, 1, 2)
plt.plot(x1, np.cos(x1));
```

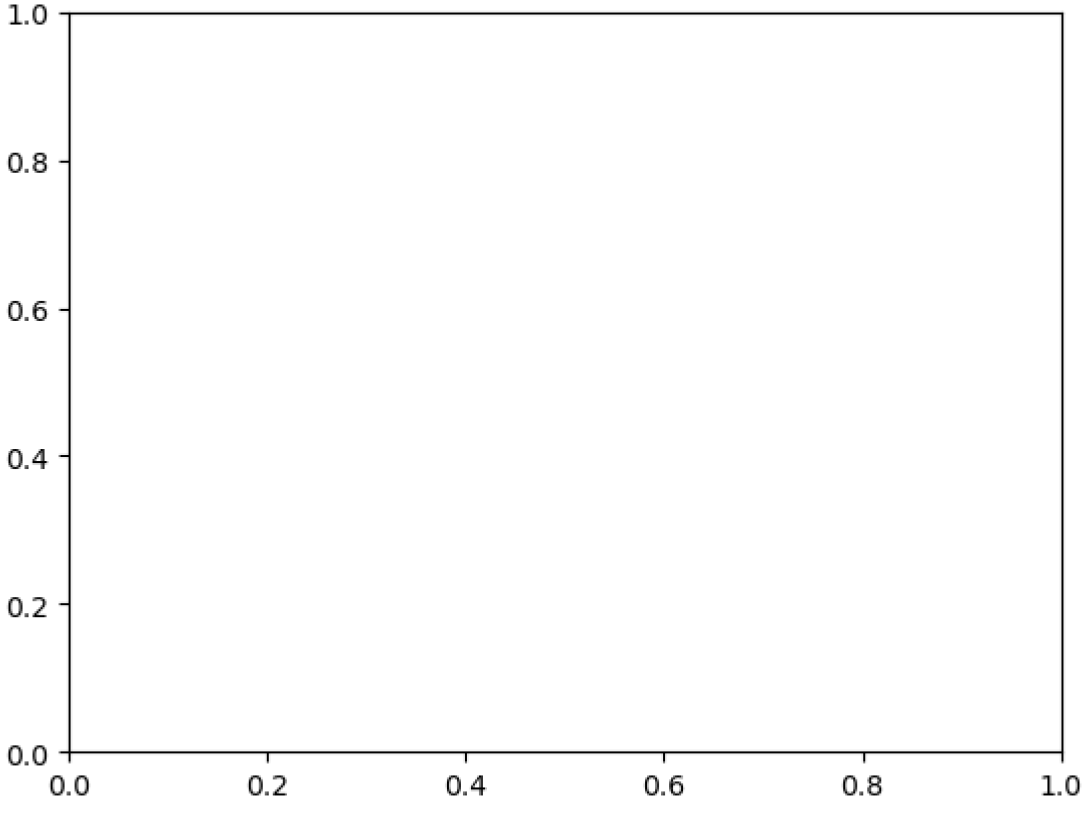


In [5]:
```
print(plt.gcf())
```

```
Figure(640x480)
<Figure size 640x480 with 0 Axes>
```

In [6]:
```
print(plt.gca())
```

```
Axes(0.125,0.11;0.775x0.77)
```

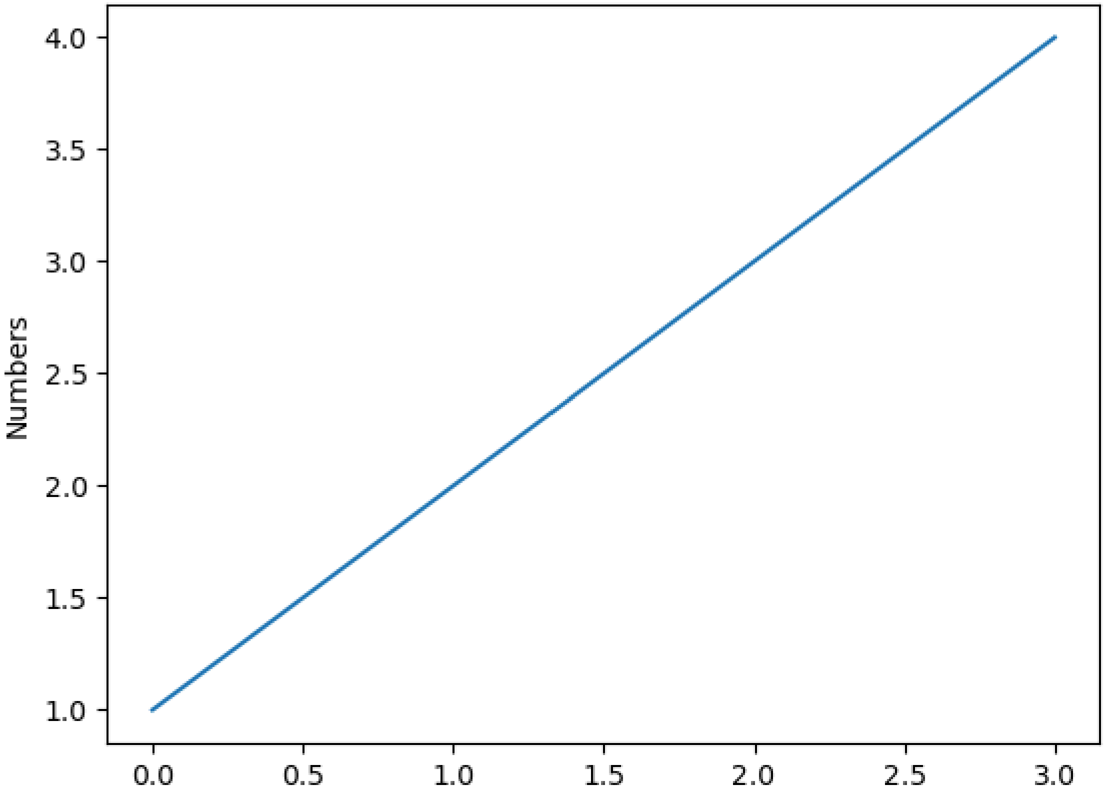


In [7]:
```
print(plt.gci())
```

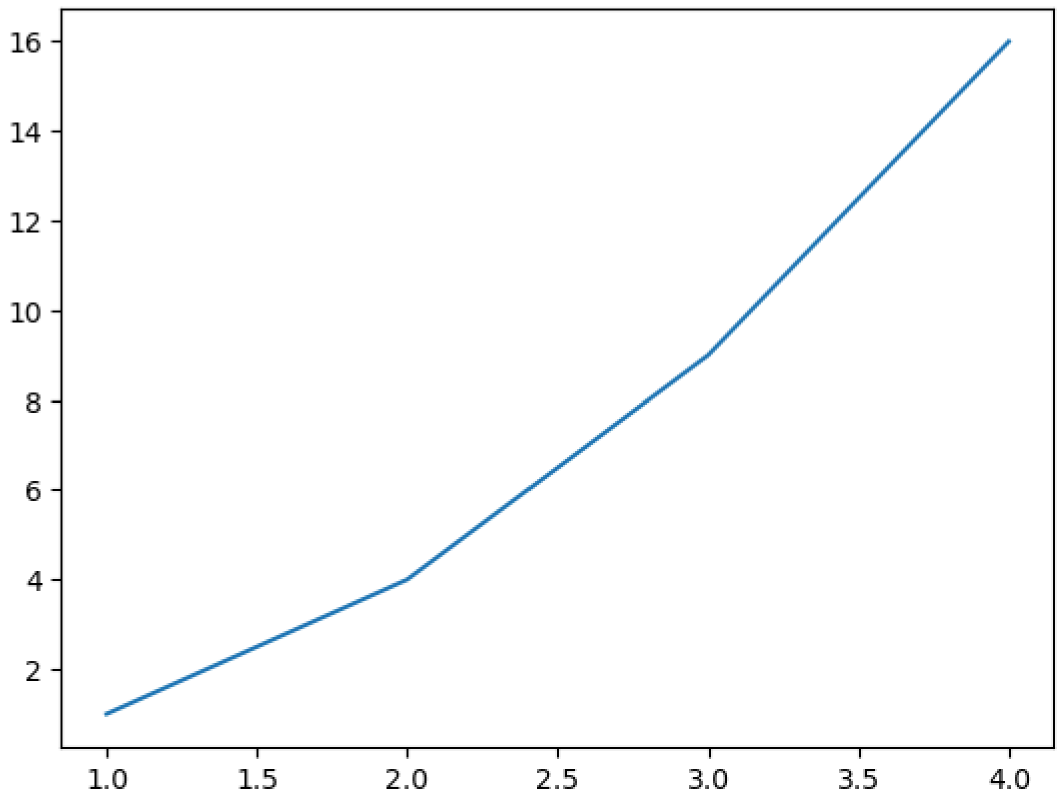
```
None
<Figure size 640x480 with 0 Axes>
```

Visualization with Pyplot

In [8]:
```
plt.plot([1, 2, 3, 4])
plt.ylabel('Numbers')
plt.show()
```



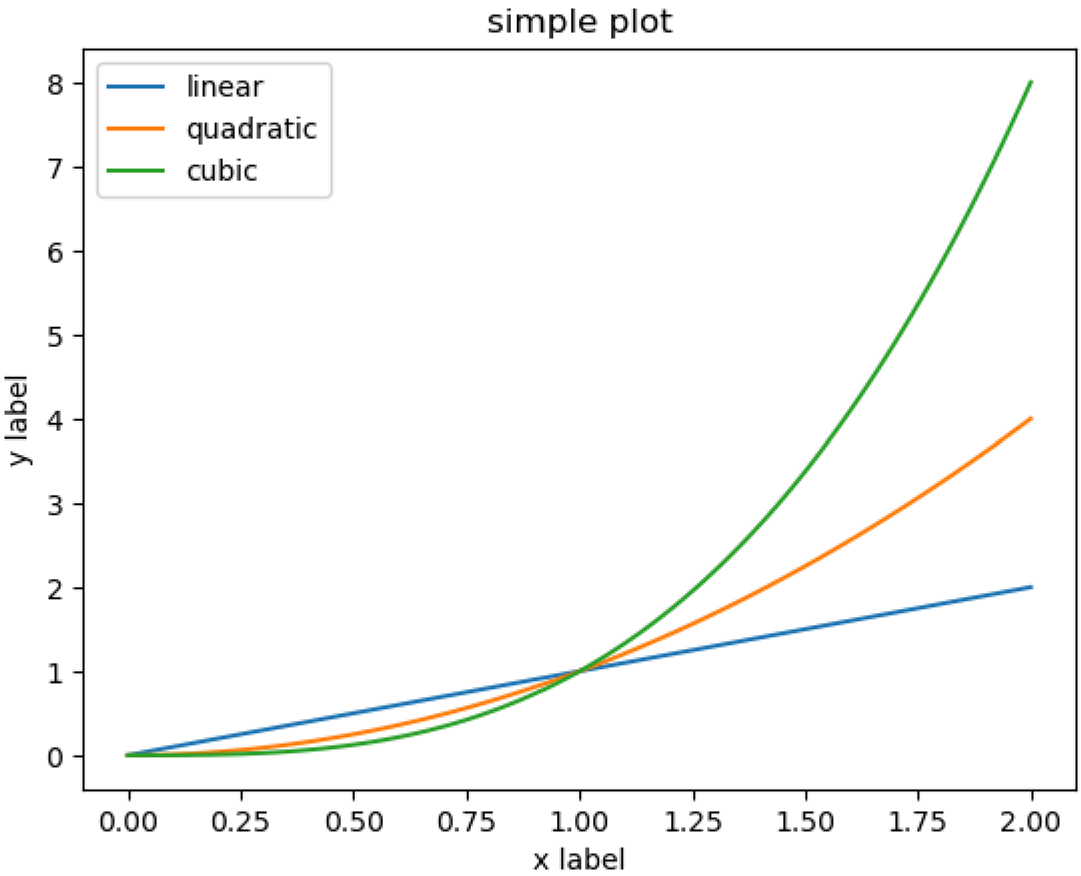
In [9]:
```
plt.plot([1, 2, 3, 4],[1, 4, 9, 16])
plt.show()
```



State-machine interface

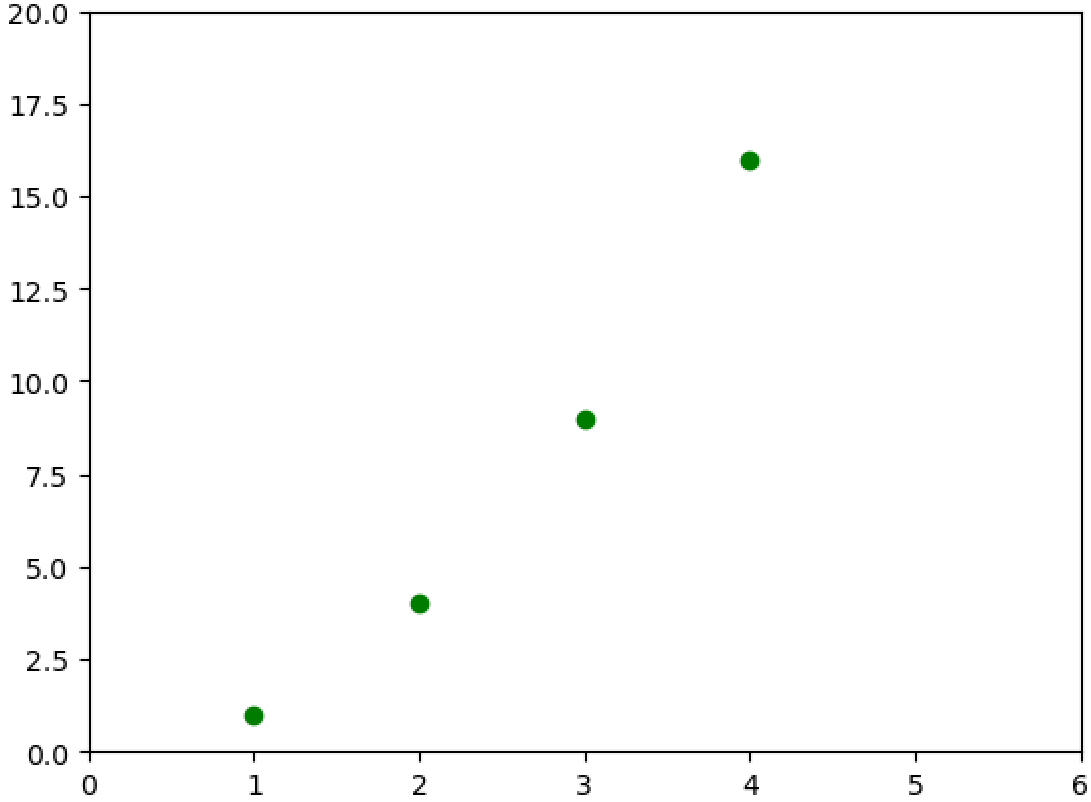
In [10]:

```
x = np.linspace(0, 2, 100)
plt.plot(x, x, label='linear')
plt.plot(x, x**2, label='quadratic')
plt.plot(x, x**3, label='cubic')

plt.xlabel('x label')
plt.ylabel('y label')
plt.title("simple plot")

plt.legend()
plt.show()
```



In [11]:

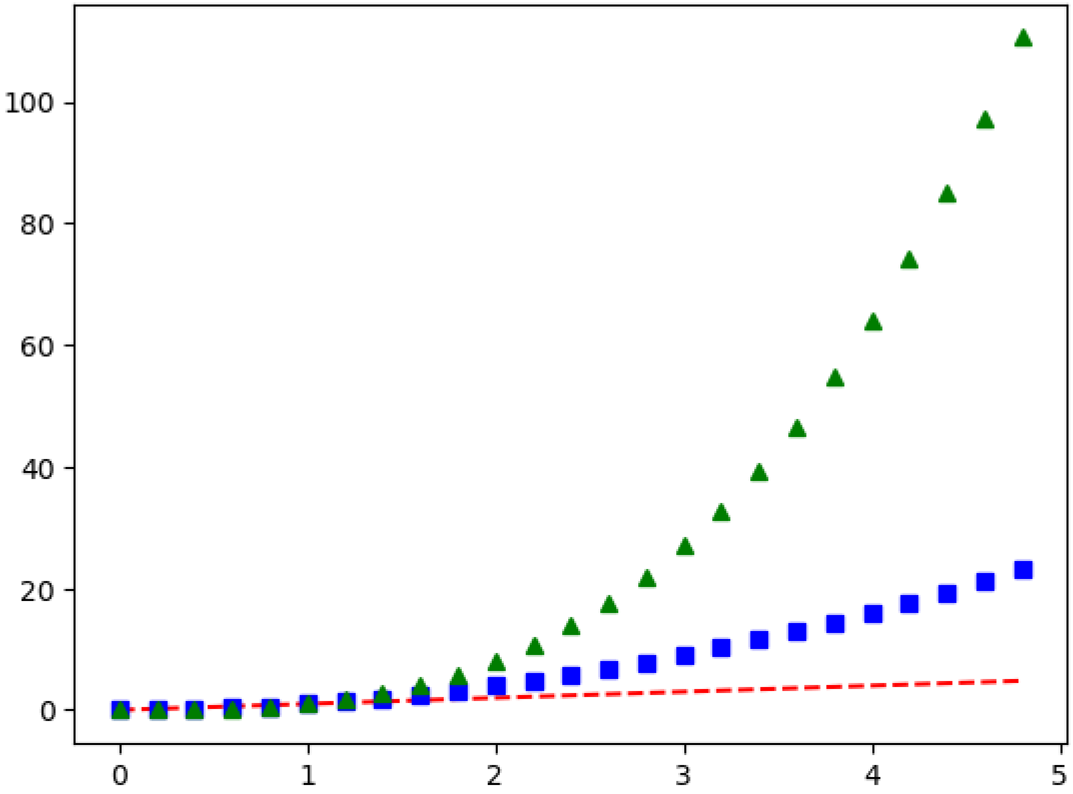
```python
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'go')
plt.axis([0, 6, 0, 20])
plt.show()
```



Working with Numpy arrays

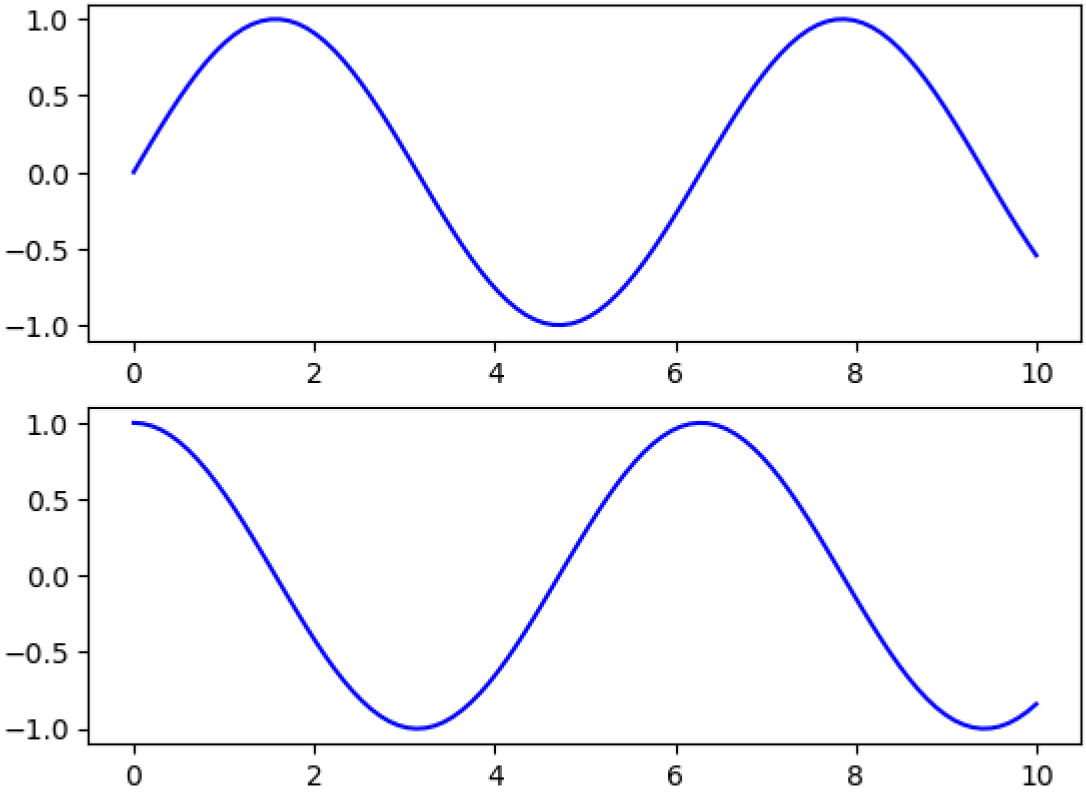
In [12]:

```
t = np.arange(0., 5., 0.2)

plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```

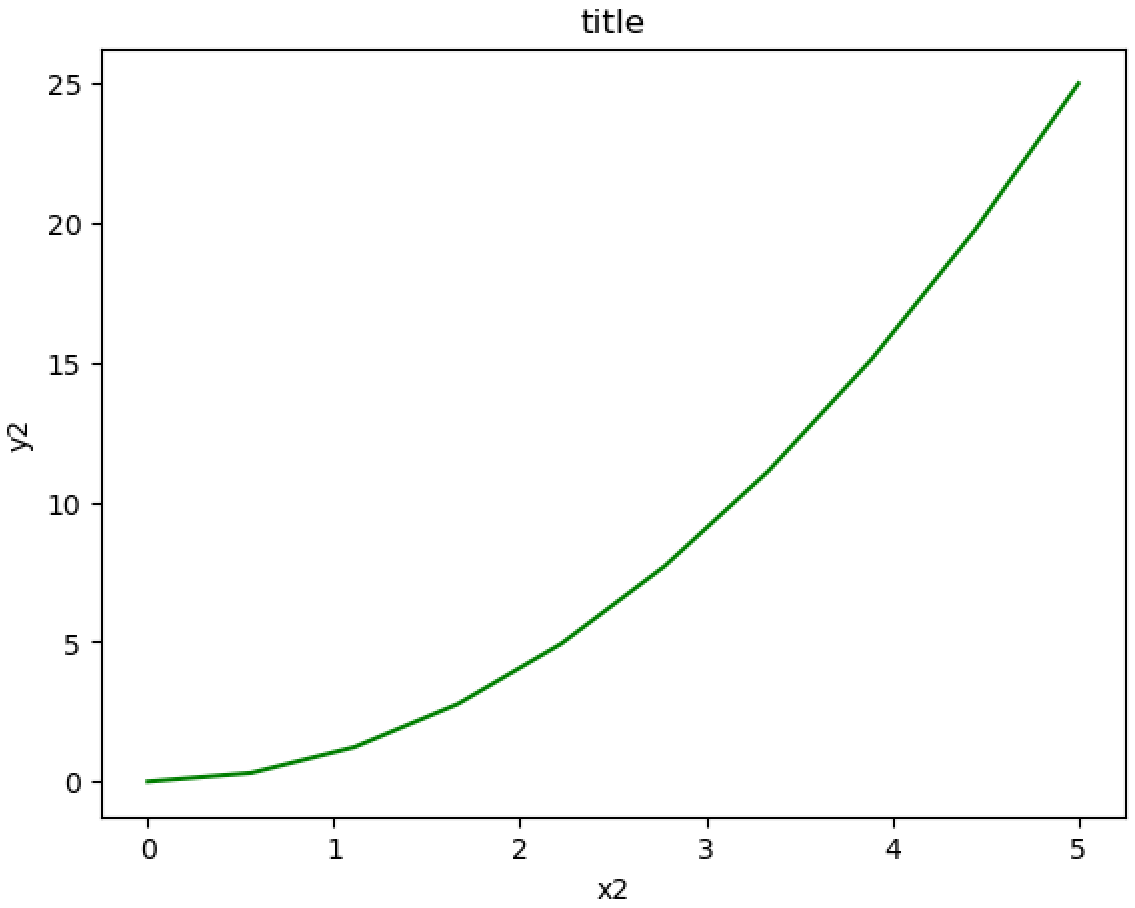


In [13]:

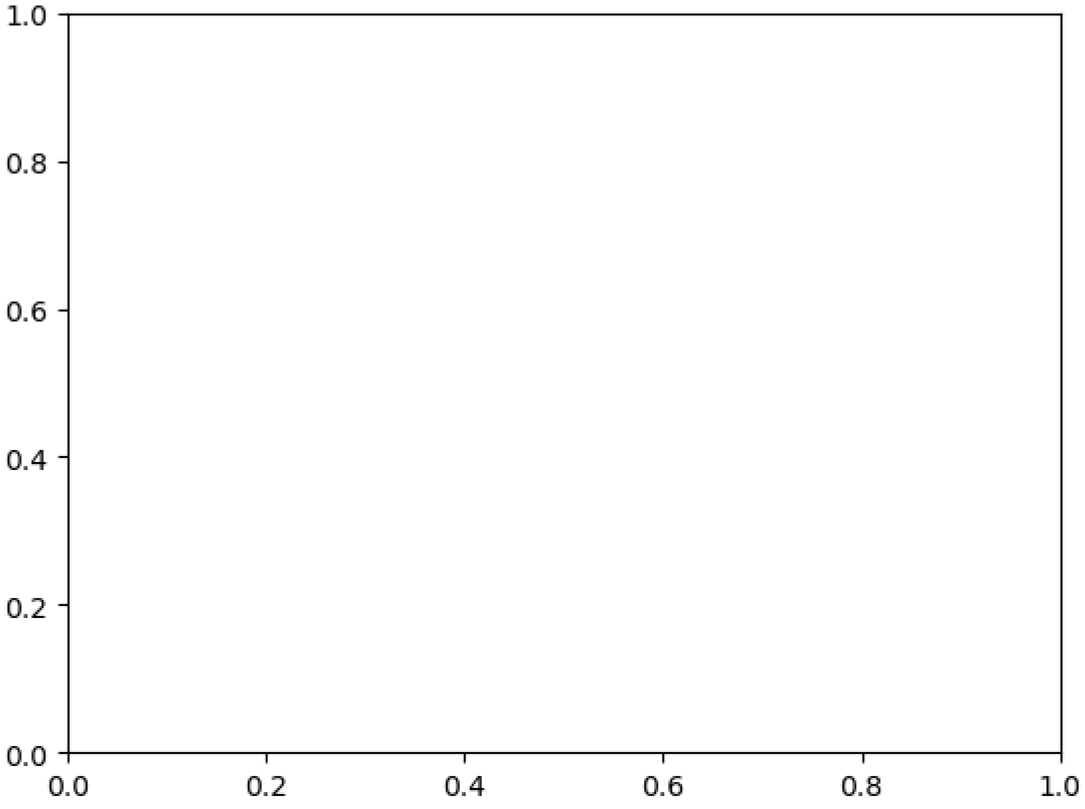
```
fig, ax = plt.subplots(2)

ax[0].plot(x1, np.sin(x1), 'b-')
ax[1].plot(x1, np.cos(x1), 'b-');
```



In [14]:
```
fig = plt.figure()

x2 = np.linspace(0, 5, 10)
y2 = x2 ** 2

axes = fig.add_axes([0.1, 0.1, 0.8, 0.8])

axes.plot(x2, y2, 'g')

axes.set_xlabel('x2')
axes.set_ylabel('y2')
axes.set_title('title');
```

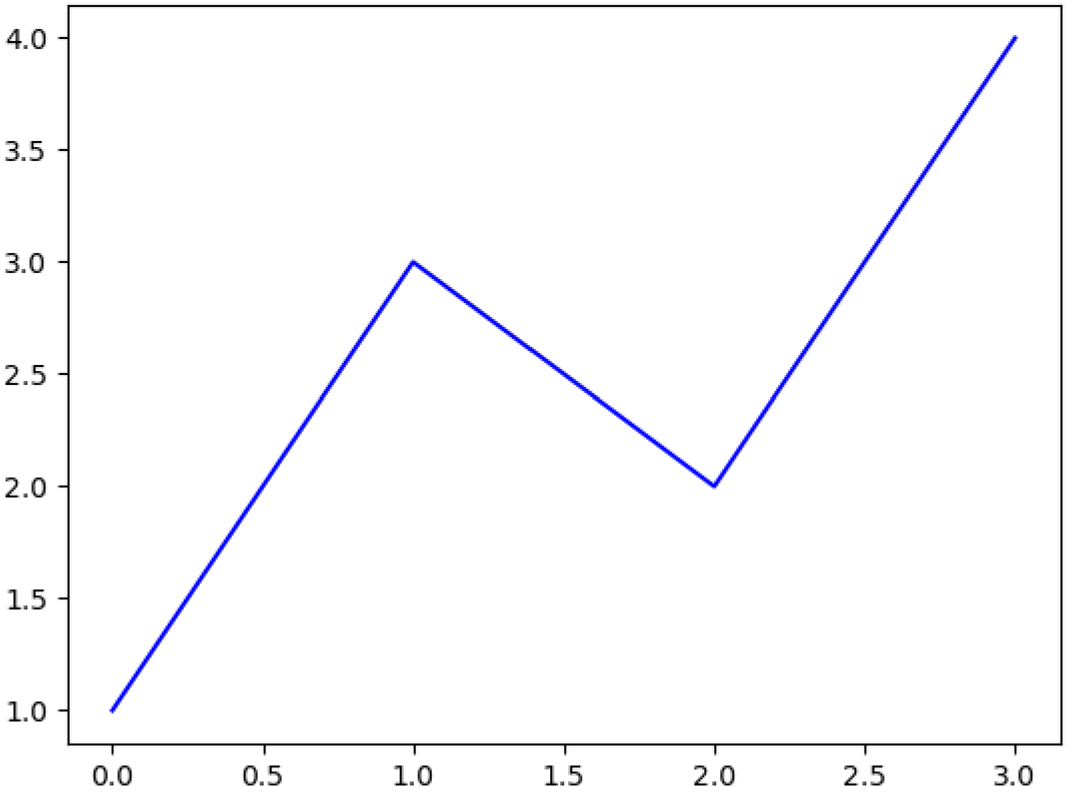


In [15]:
```
fig = plt.figure()
ax = plt.axes()
```

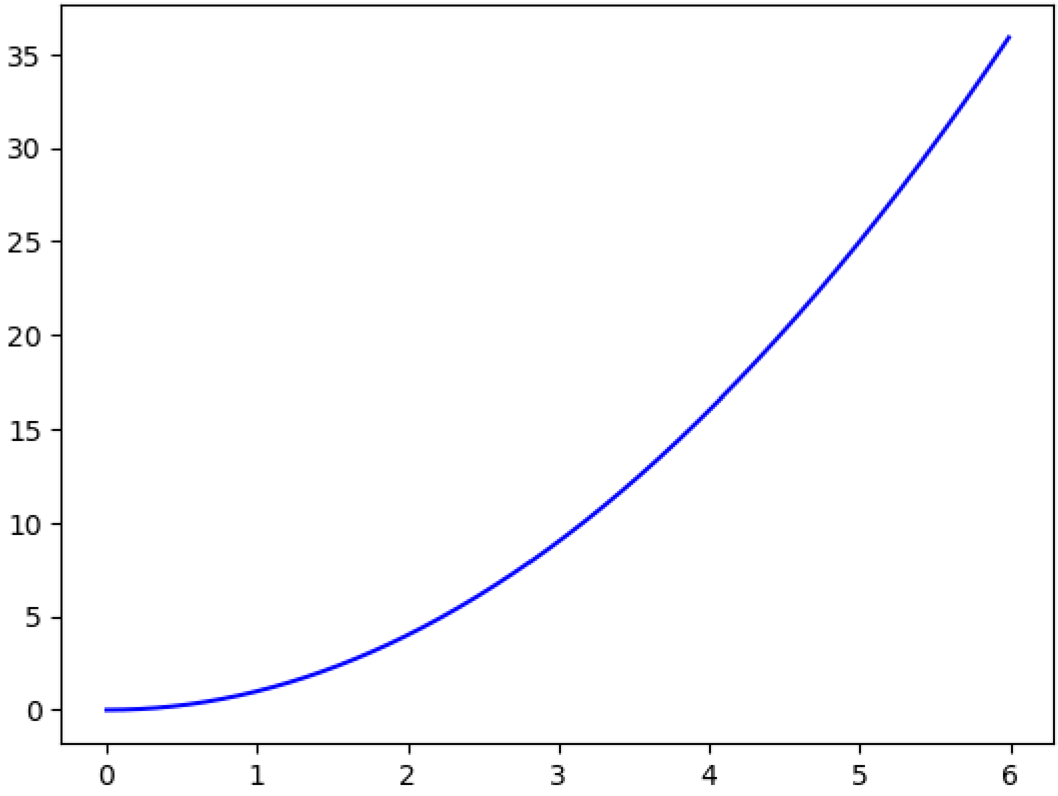


plots with matplotlib

In [16]:
```
plt.plot([1, 3, 2, 4], 'b-')
plt.show()
```

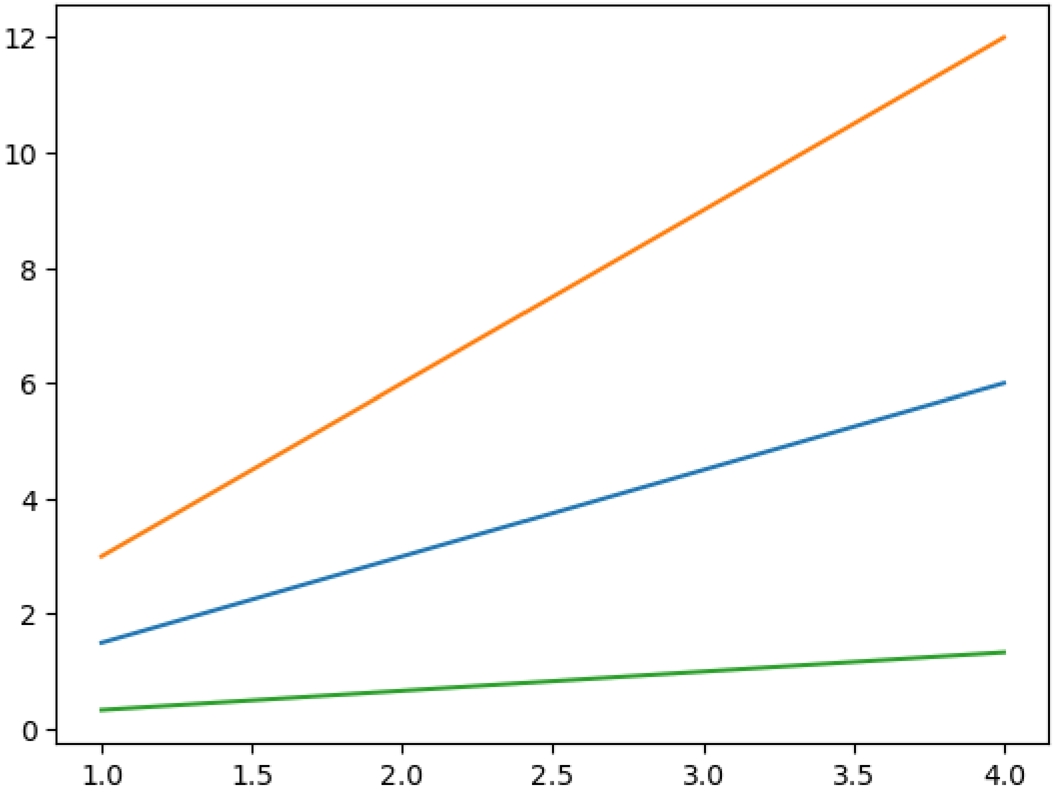


In [17]:
```
x3 = np.arange(0.0, 6.0,0.01)
plt.plot(x3, [xi**2 for xi in x3], 'b-')
plt.show()
```



Multiline plots

In [18]:
```
x4 = range(1, 5)

plt.plot(x4, [xi*1.5 for xi in x4])
plt.plot(x4, [xi*3 for xi in x4])
plt.plot (x4, [xi/3.0 for xi in x4])
plt.show()
```



In [19]:
```
fig.savefig('plot1.png')
```

In [20]:
```
pip install Ipython
```

```
Requirement already satisfied: Ipython in c:\users\admin\anaconda3\lib\site-packa
ges (8.25.0)
Requirement already satisfied: decorator in c:\users\admin\anaconda3\lib\site-pac
kages (from Ipython) (5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\admin\anaconda3\lib\site-pa
ckages (from Ipython) (0.18.1)
Requirement already satisfied: matplotlib-inline in c:\users\admin\anaconda3\lib
\site-packages (from Ipython) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in c:\users\admin\an
aconda3\lib\site-packages (from Ipython) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in c:\users\admin\anaconda3\lib\si
te-packages (from Ipython) (2.15.1)
Requirement already satisfied: stack-data in c:\users\admin\anaconda3\lib\site-pa
ckages (from Ipython) (0.2.0)
Requirement already satisfied: traitlets>=5.13.0 in c:\users\admin\anaconda3\lib
\site-packages (from Ipython) (5.14.3)
Requirement already satisfied: colorama in c:\users\admin\anaconda3\lib\site-pack
ages (from Ipython) (0.4.6)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\admin\anaconda3\li
b\site-packages (from jedi>=0.16->Ipython) (0.8.3)
Requirement already satisfied: wcwidth in c:\users\admin\anaconda3\lib\site-packa
ges (from prompt-toolkit<3.1.0,>=3.0.41->Ipython) (0.2.5)
Requirement already satisfied: executing in c:\users\admin\anaconda3\lib\site-pac
kages (from stack-data->Ipython) (0.8.3)
Requirement already satisfied: asttokens in c:\users\admin\anaconda3\lib\site-pac
kages (from stack-data->Ipython) (2.0.5)
Requirement already satisfied: pure-eval in c:\users\admin\anaconda3\lib\site-pac
kages (from stack-data->Ipython) (0.2.2)
Requirement already satisfied: six in c:\users\admin\anaconda3\lib\site-packages
(from asttokens->stack-data->Ipython) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [21]:

```python
fig.canvas.get_supported_filetypes()
```

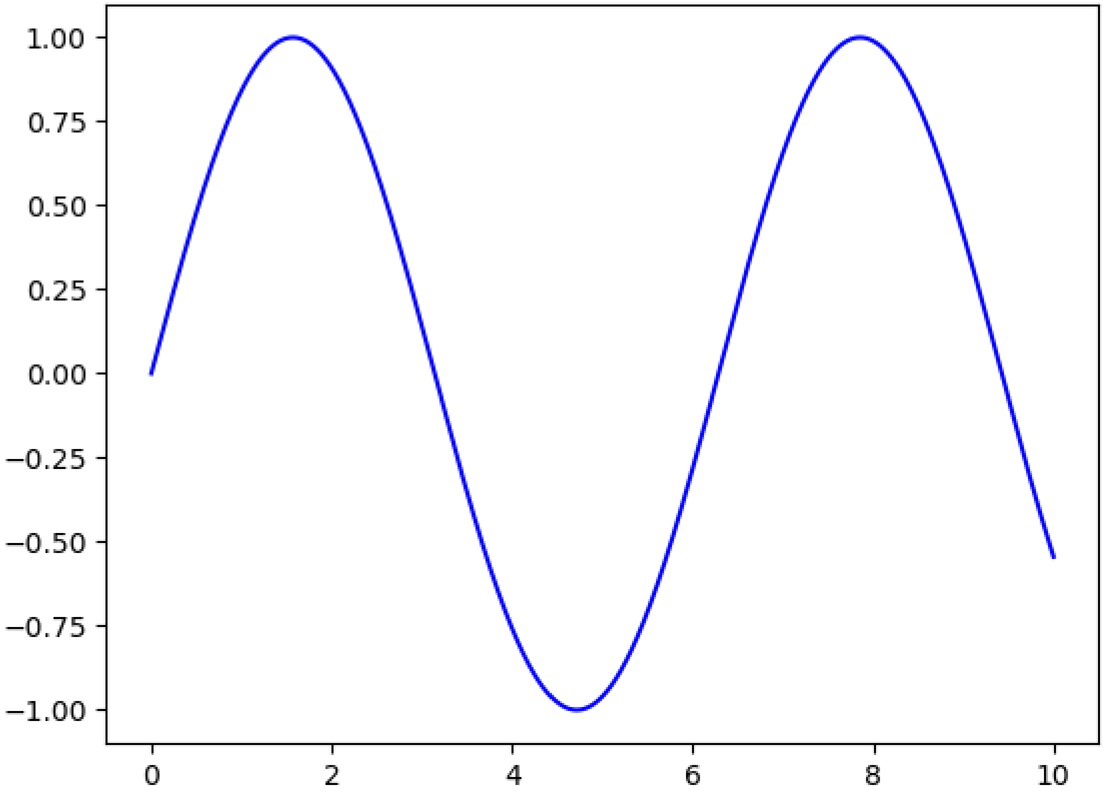
Out[21]:

```
{'eps': 'Encapsulated Postscript',
 'jpg': 'Joint Photographic Experts Group',
 'jpeg': 'Joint Photographic Experts Group',
 'pdf': 'Portable Document Format',
 'pgf': 'PGF code for LaTeX',
 'png': 'Portable Network Graphics',
 'ps': 'Postscript',
 'raw': 'Raw RGBA bitmap',
 'rgba': 'Raw RGBA bitmap',
 'svg': 'Scalable Vector Graphics',
 'svgz': 'Scalable Vector Graphics',
 'tif': 'Tagged Image File Format',
 'tiff': 'Tagged Image File Format',
 'webp': 'WebP Image Format'}
```

Line plot

In [22]:

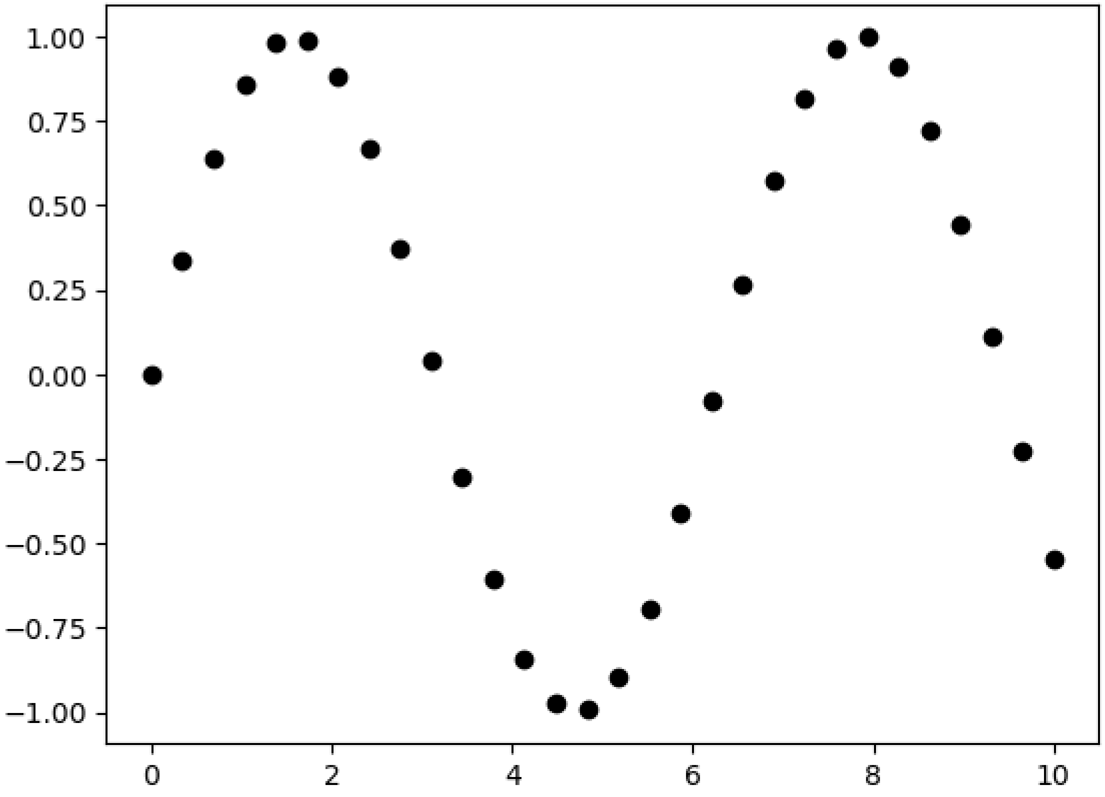
```python
fig = plt.figure()

ax = plt.axes()

x5 = np.linspace(0, 10, 1000)

ax.plot(x5, np.sin(x5), 'b-');
```



Scatter Plot

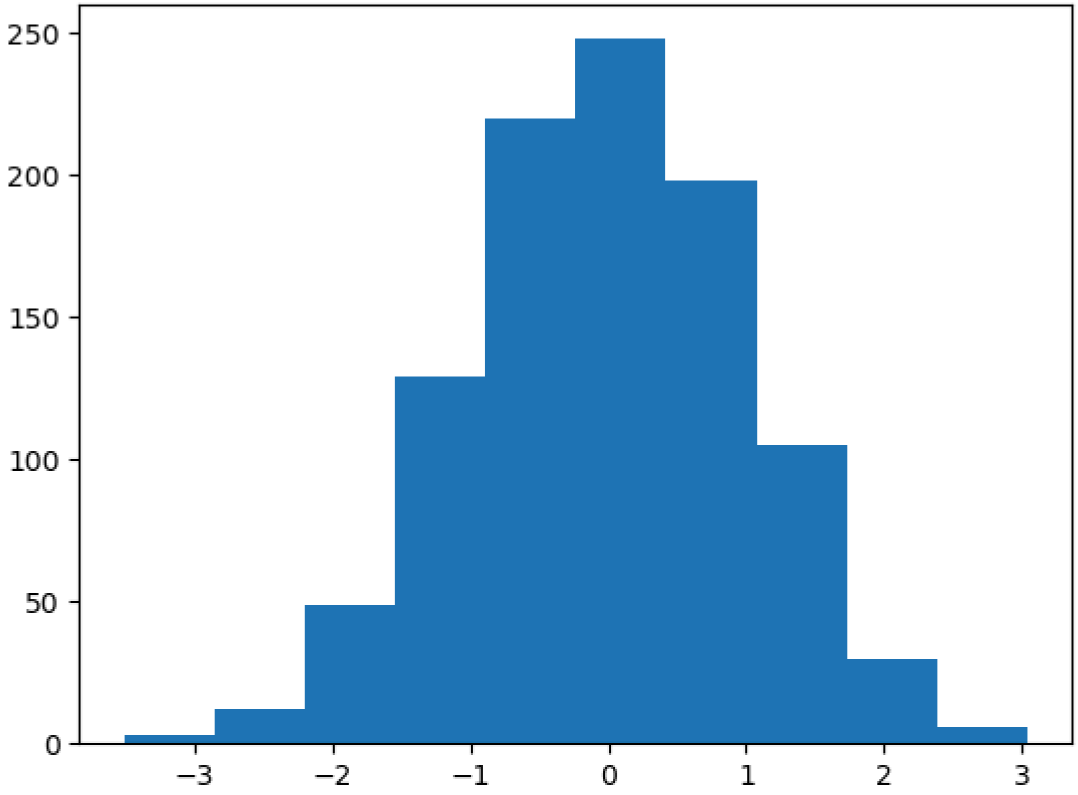
In [23]:

```
x7 = np.linspace(0, 10, 30)

y7 = np.sin(x7)

plt.plot(x7, y7, 'o', color = 'black');
```



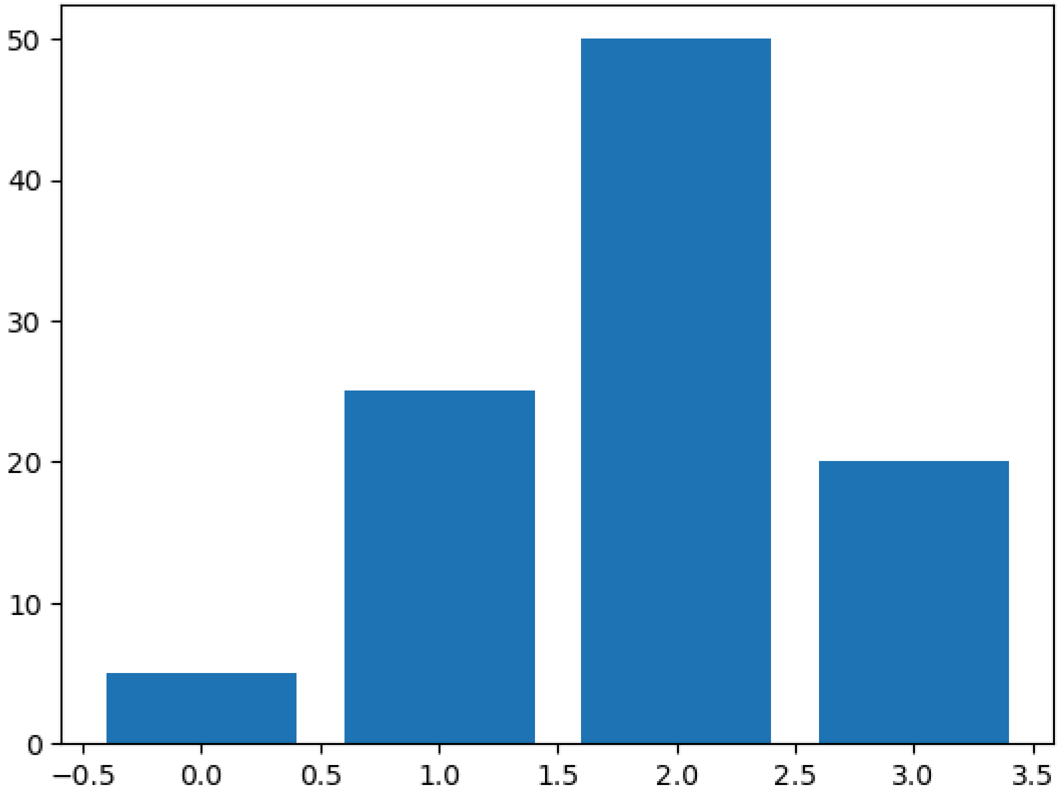
Histogram

In [24]:
```
data1 = np.random.randn(1000)

plt.hist(data1);
```



Bar chart

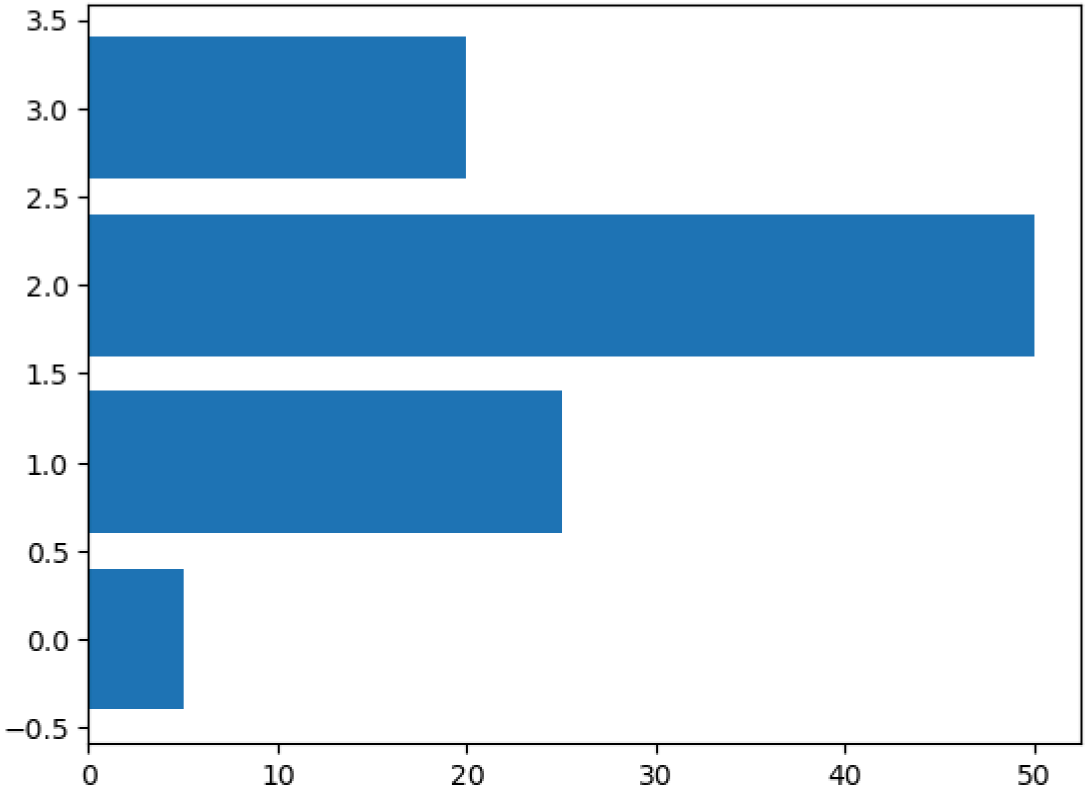
In [25]:
```
data2 = [5. , 25. , 50. , 20.]

plt.bar(range(len(data2)), data2)

plt.show()
```



Horizontal Bar Chart

In [26]:

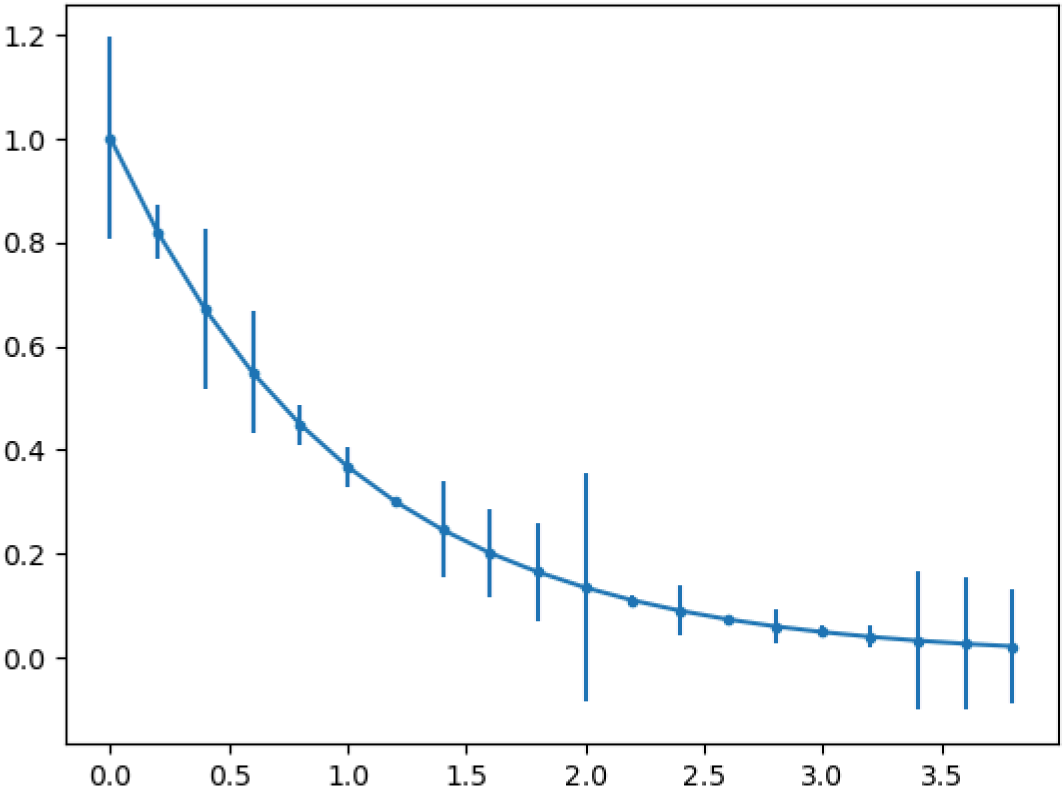
```
data2 = [5. , 25. , 50. , 20.]

plt.barh(range(len(data2)), data2)

plt.show()
```



Error of Bar Chart

In [27]:

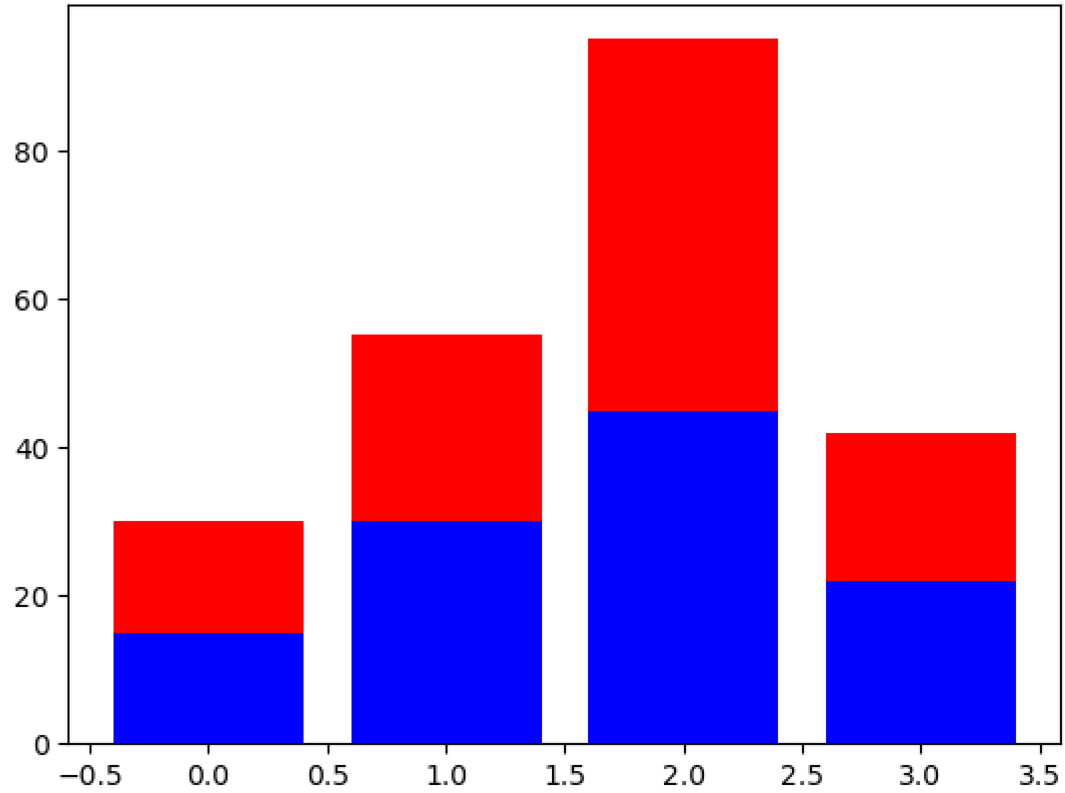
```
x9 = np.arange(0, 4, 0.2)

y9 = np.exp(-x9)

e1 = 0.1 * np.abs(np.random.randn(len(y9)))

plt.errorbar(x9, y9, yerr = e1, fmt = '.-')

plt.show();
```



Stacked Bar Chart

In [28]:

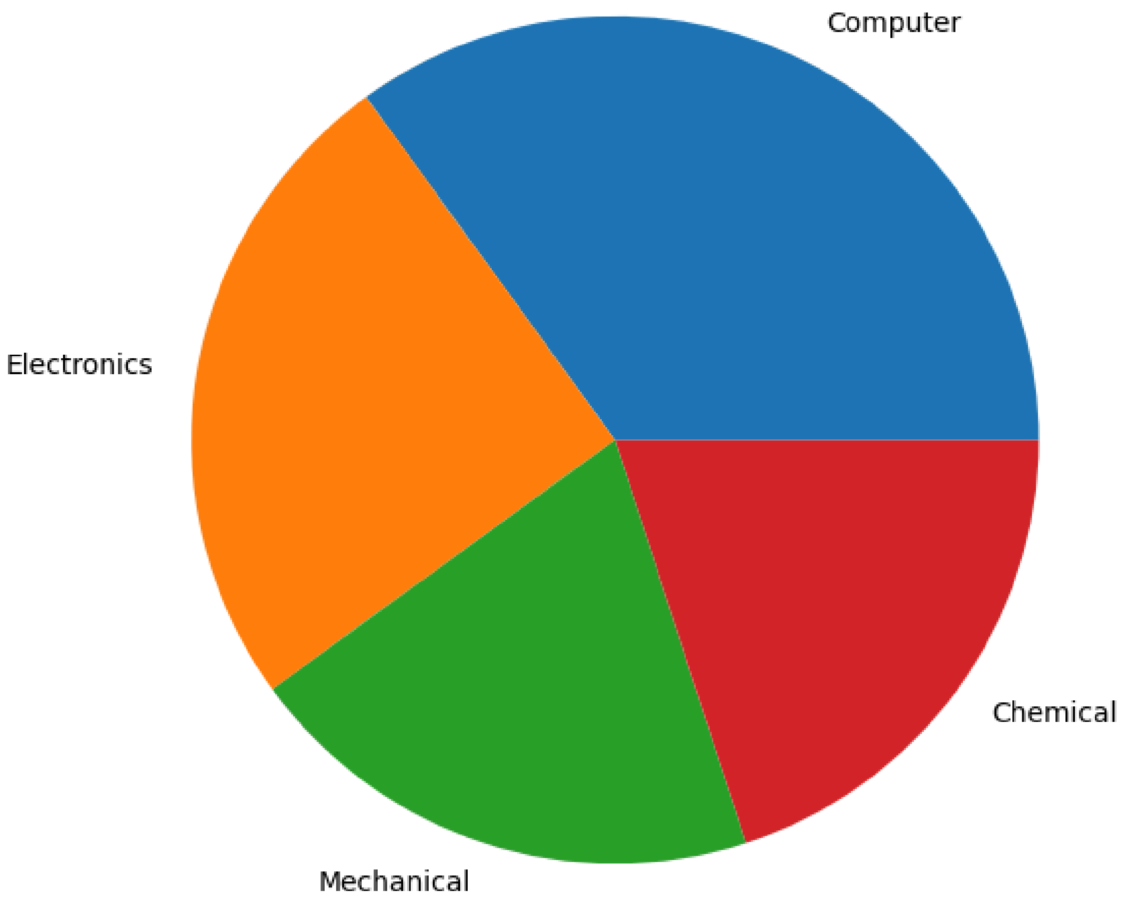
```
A = [15., 30., 45., 22.]

B = [15., 25., 50., 20.]

z2 = range(4)

plt.bar(z2, A, color = 'b')
plt.bar(z2, B, color = 'r', bottom = A)

plt.show()
```



Pie Chart

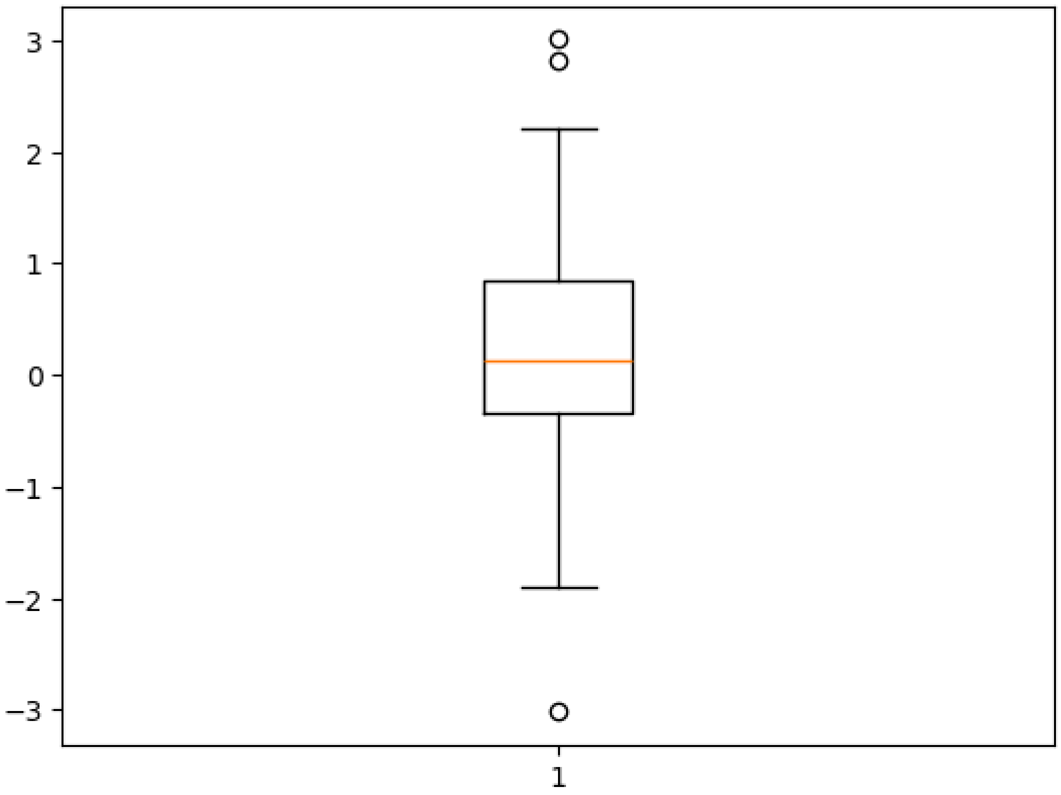
In [29]:

```
plt.figure(figsize=(7,7))

x10 = [35, 25, 20, 20]

labels = ['Computer', 'Electronics', 'Mechanical', 'Chemical']

plt.pie(x10, labels=labels);

plt.show()
```



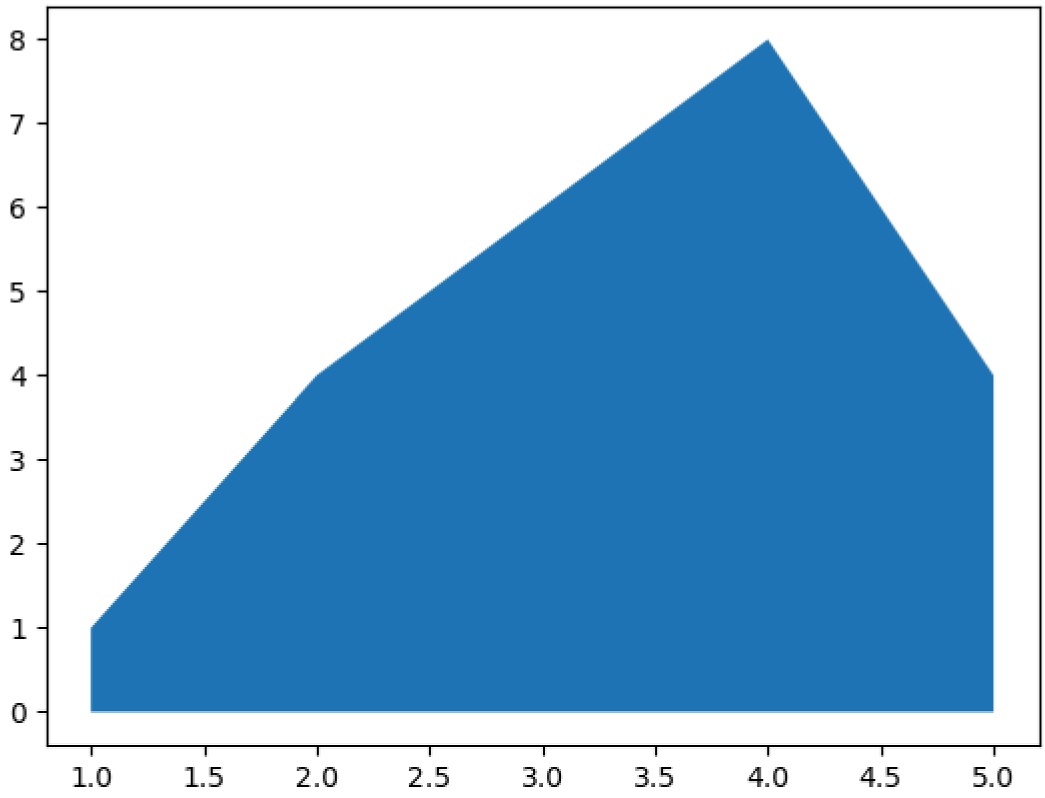
Box Plot

In [30]:
```
data3 = np.random.randn(100)

plt.boxplot(data3)

plt.show();
```



Area Chart

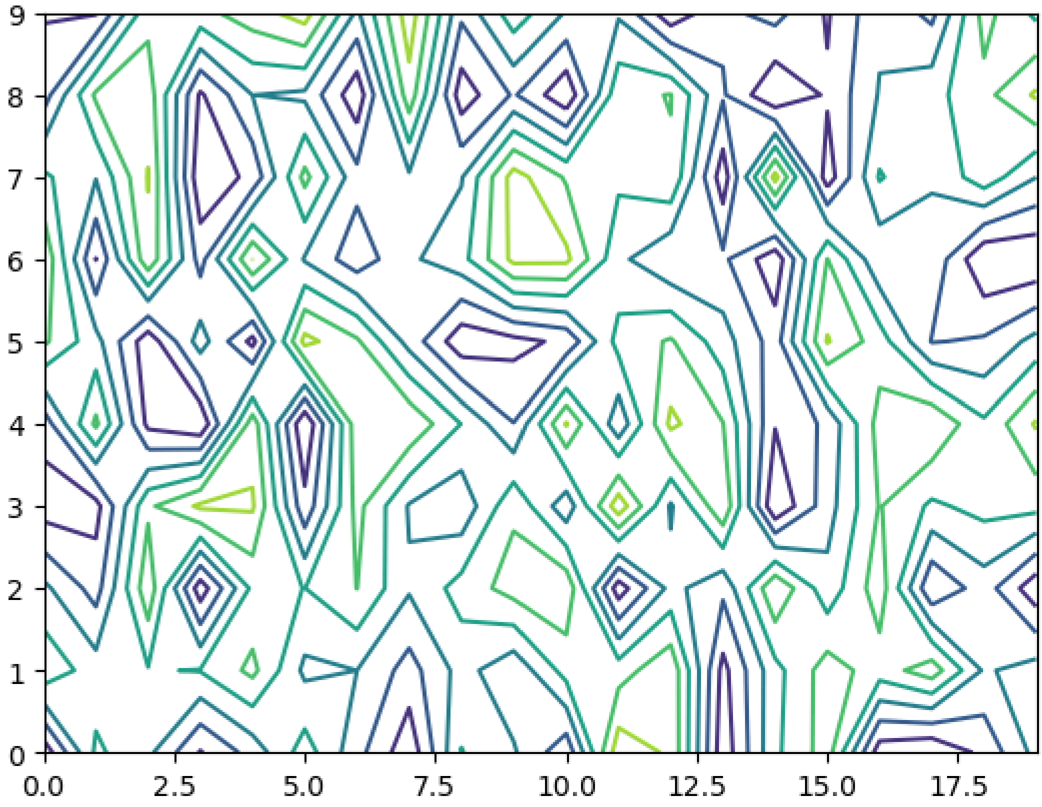
```
In [31]: # Create some data
x12 = range(1, 6)
y12 = [1, 4, 6, 8, 4]

# Area plot
plt.fill_between(x12, y12)
plt.show()
```



Contour Plot

In [32]:

```
# Create a matrix
matrix1 = np.random.rand(10, 20)

cp = plt.contour(matrix1)

plt.show()
```



Styles with Matplotlib plots

In [33]:
```
# View list of all available styles

print(plt.style.available)
```

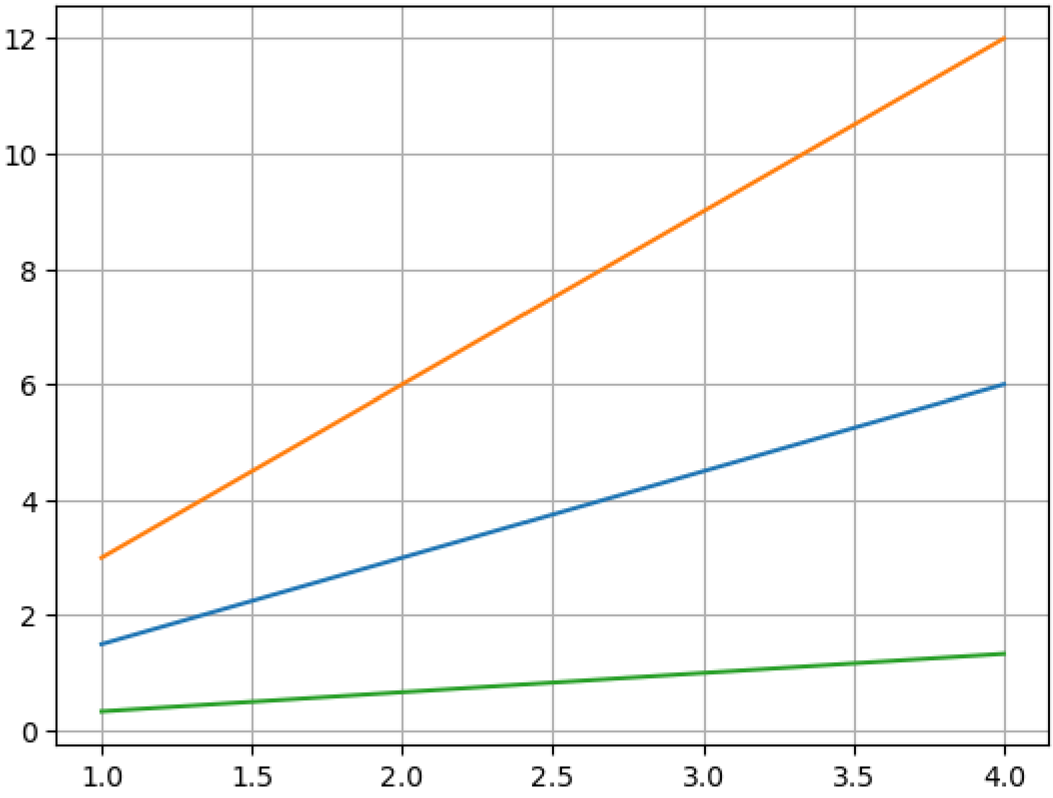
```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid',
'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'graysc
ale', 'seaborn-v0_8', 'seaborn-v0_8-bright', 'seaborn-v0_8-colorblind', 'seaborn-
v0_8-dark', 'seaborn-v0_8-dark-palette', 'seaborn-v0_8-darkgrid', 'seaborn-v0_8-d
eep', 'seaborn-v0_8-muted', 'seaborn-v0_8-notebook', 'seaborn-v0_8-paper', 'seabo
rn-v0_8-pastel', 'seaborn-v0_8-poster', 'seaborn-v0_8-talk', 'seaborn-v0_8-tick
s', 'seaborn-v0_8-white', 'seaborn-v0_8-whitegrid', 'tableau-colorblind10']
```

In [34]:
```
# View list of all available styles

print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid',
'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'graysc
ale', 'seaborn-v0_8', 'seaborn-v0_8-bright', 'seaborn-v0_8-colorblind', 'seaborn-
v0_8-dark', 'seaborn-v0_8-dark-palette', 'seaborn-v0_8-darkgrid', 'seaborn-v0_8-d
eep', 'seaborn-v0_8-muted', 'seaborn-v0_8-notebook', 'seaborn-v0_8-paper', 'seabo
rn-v0_8-pastel', 'seaborn-v0_8-poster', 'seaborn-v0_8-talk', 'seaborn-v0_8-tick
s', 'seaborn-v0_8-white', 'seaborn-v0_8-whitegrid', 'tableau-colorblind10']
```

adding a grid

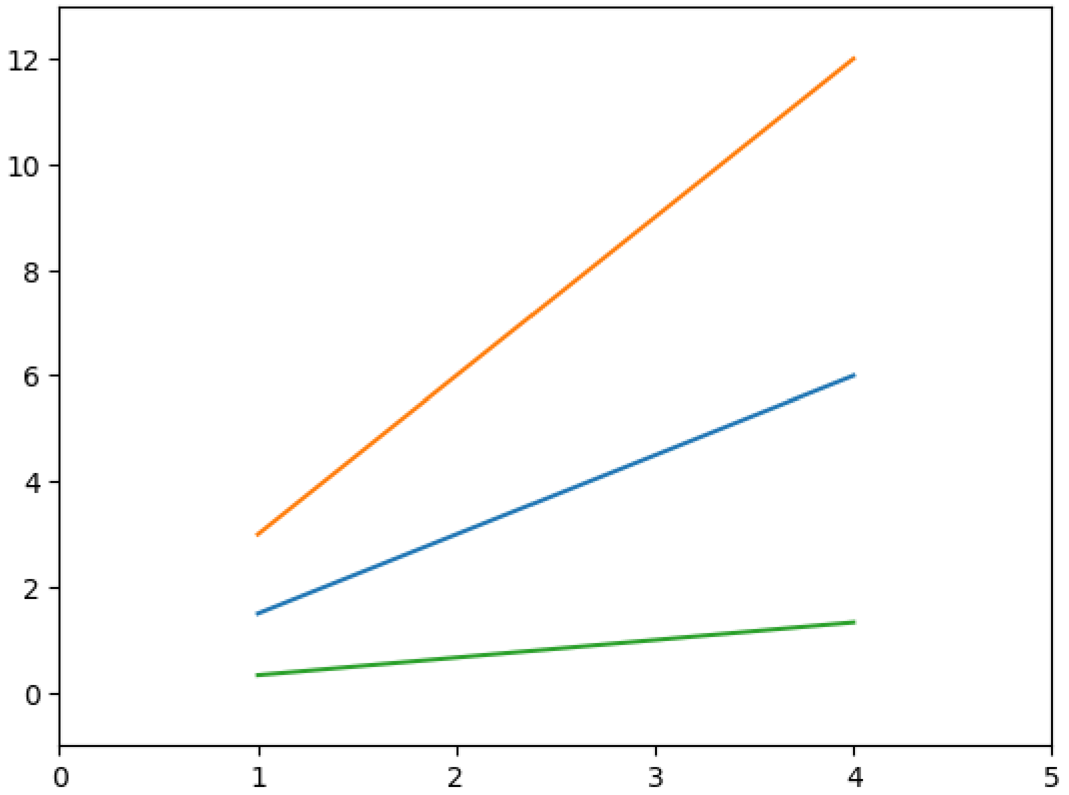
In [35]:
```
x15 = np.arange(1, 5)

plt.plot(x15, x15*1.5, x15, x15*3.0, x15, x15/3.0)
plt.grid(True)

plt.show()
```



Handling axes

In [36]:

```
x15 = np.arange(1, 5)

plt.plot(x15, x15*1.5, x15, x15*3.0, x15, x15/3.0)

plt.axis()   # shows the current axis limits values

plt.axis([0, 5, -1, 13])

plt.show()
```

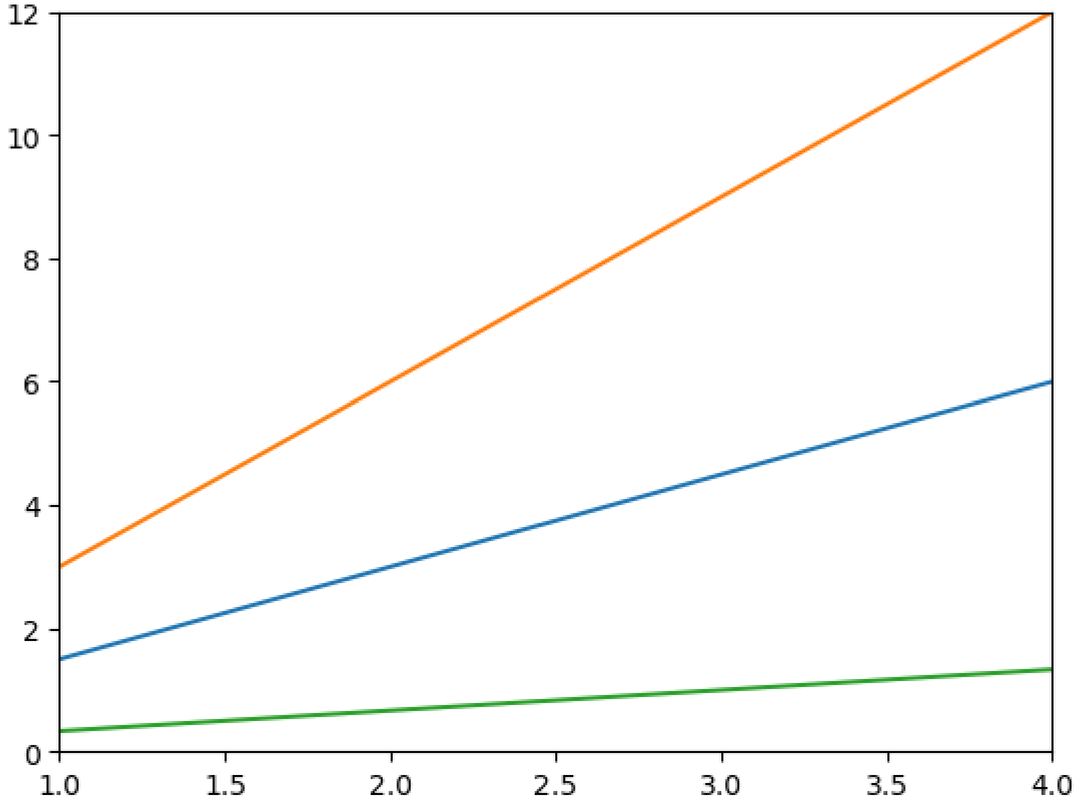


In [37]:

```
x15 = np.arange(1, 5)

plt.plot(x15, x15*1.5, x15, x15*3.0, x15, x15/3.0)

plt.xlim([1.0, 4.0])

plt.ylim([0.0, 12.0])
```

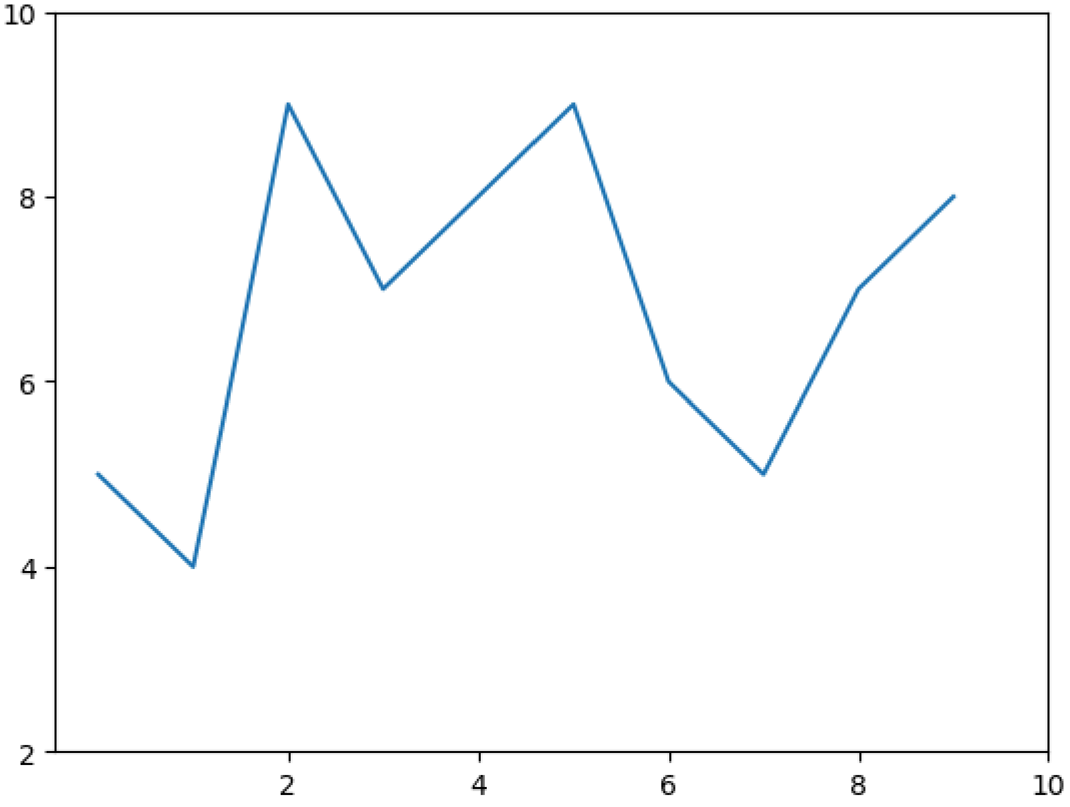
Out[37]:

```
(0.0, 12.0)
```

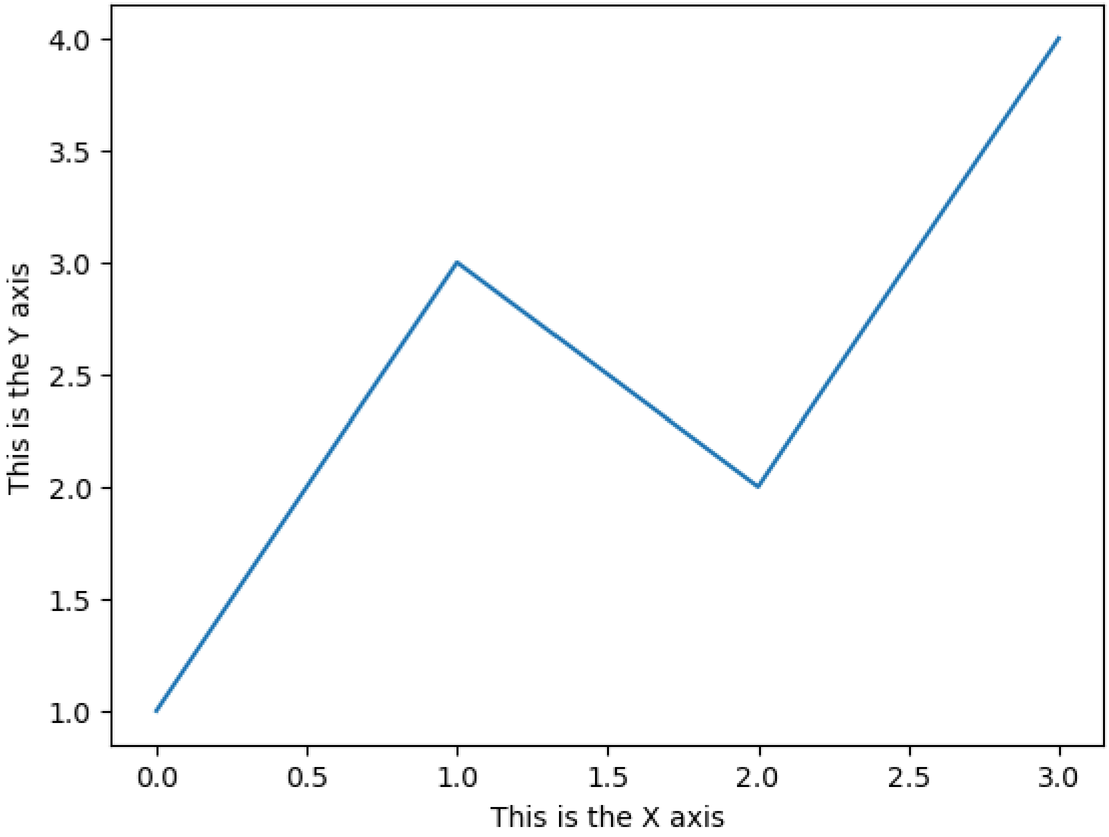


Handling x and y ticks

In [38]:
```
u = [5, 4, 9, 7, 8, 9, 6, 5, 7, 8]

plt.plot(u)

plt.xticks([2, 4, 6, 8, 10])
plt.yticks([2, 4, 6, 8, 10])

plt.show()
```

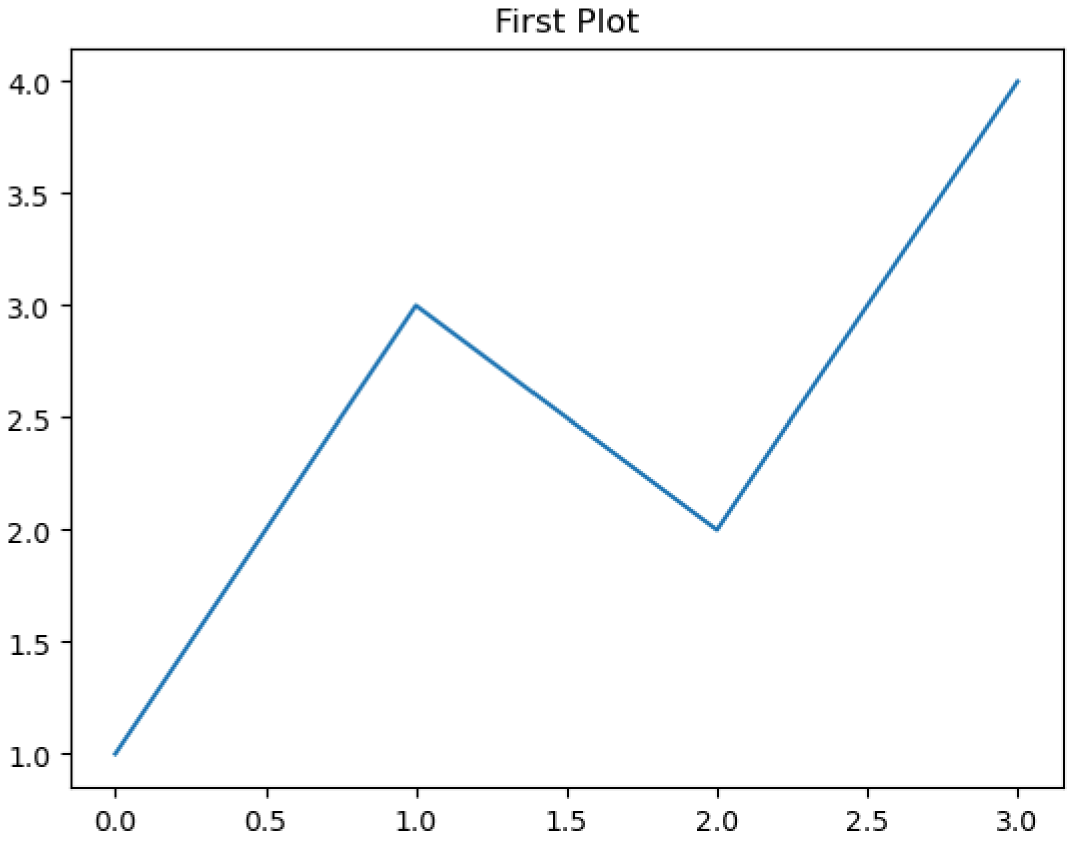


In [39]:
```
plt.plot([1, 3, 2, 4])

plt.xlabel('This is the X axis')

plt.ylabel('This is the Y axis')

plt.show()
```



Adding a title

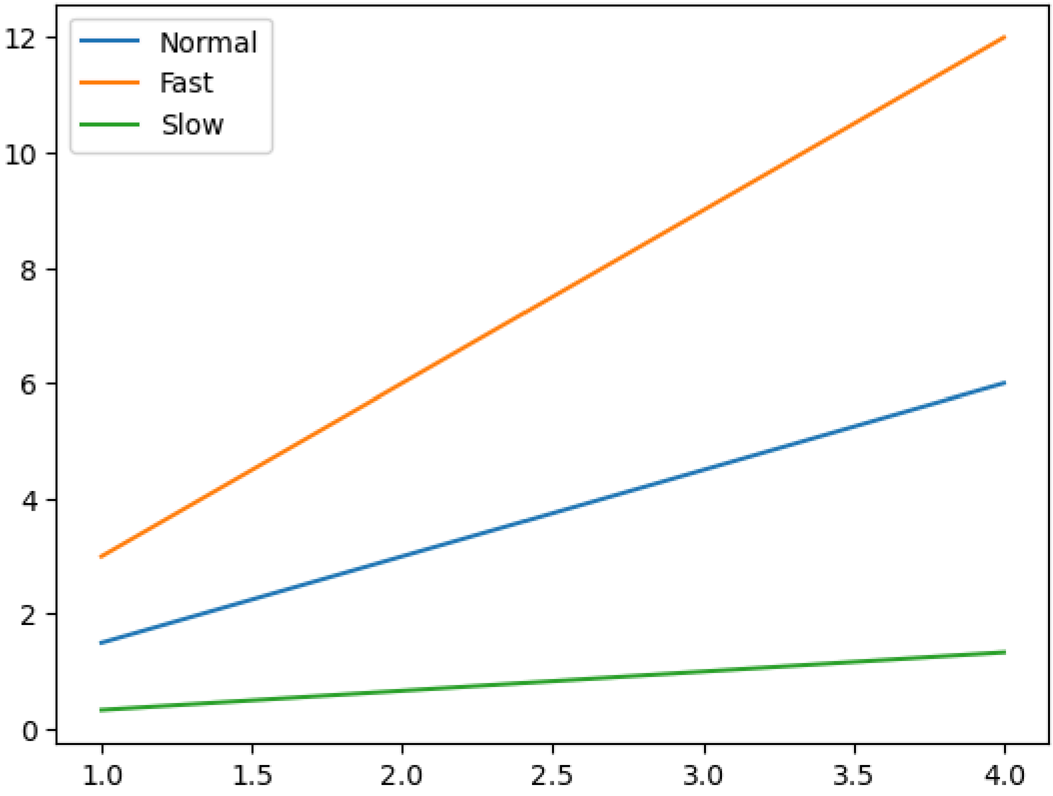
In [40]:
```
plt.plot([1, 3, 2, 4])

plt.title('First Plot')

plt.show()
```



Adding a legend

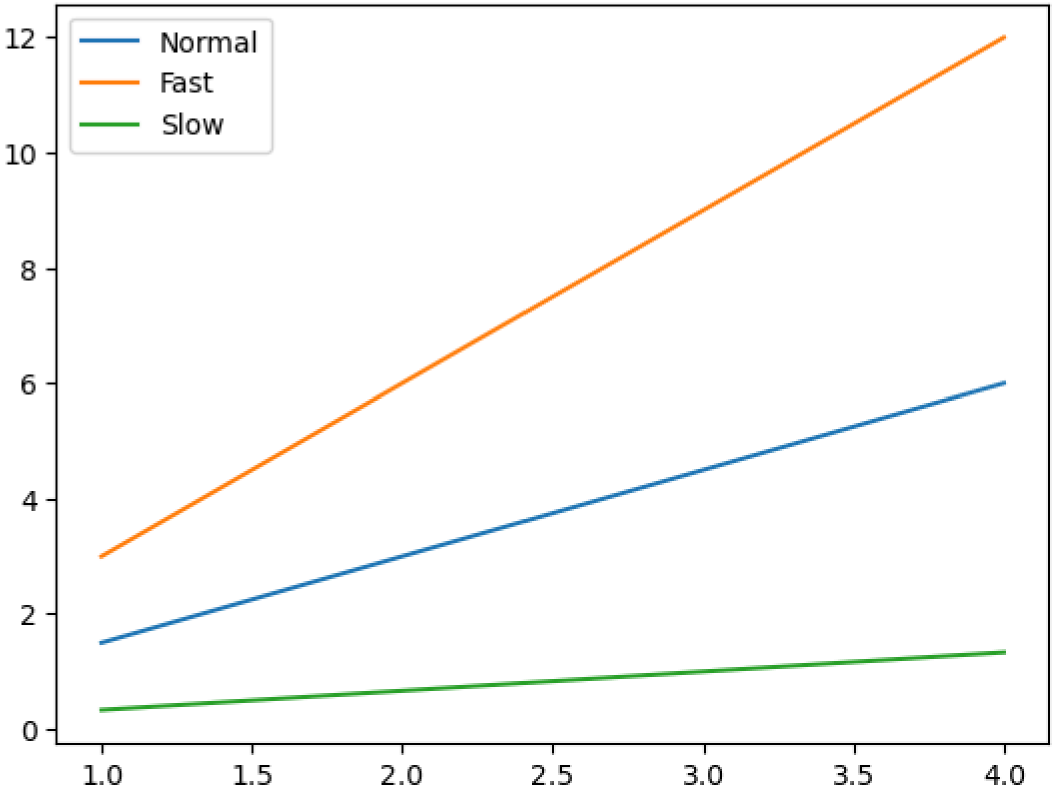
In [41]:

```
x15 = np.arange(1, 5)

fig, ax = plt.subplots()

ax.plot(x15, x15*1.5)
ax.plot(x15, x15*3.0)
ax.plot(x15, x15/3.0)

ax.legend(['Normal','Fast','Slow']);
```



In [42]:

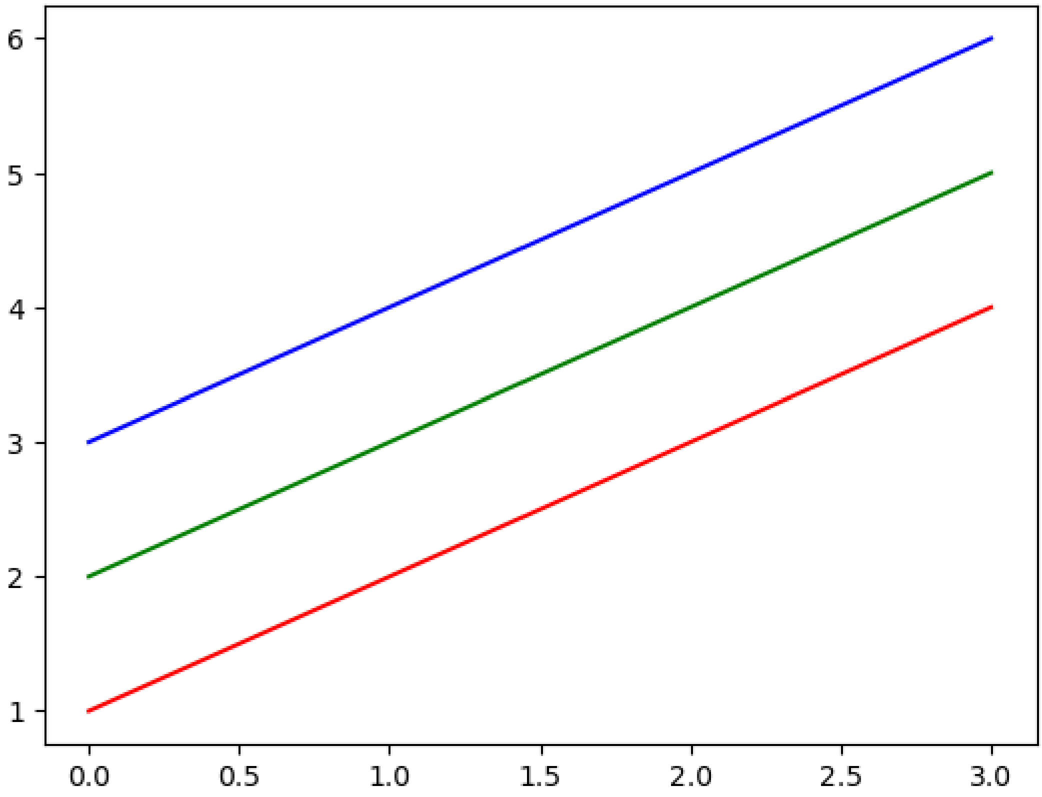
```
x15 = np.arange(1, 5)

fig, ax = plt.subplots()

ax.plot(x15, x15*1.5, label='Normal')
ax.plot(x15, x15*3.0, label='Fast')
ax.plot(x15, x15/3.0, label='Slow')

ax.legend();
```



Controls colours

In [43]:

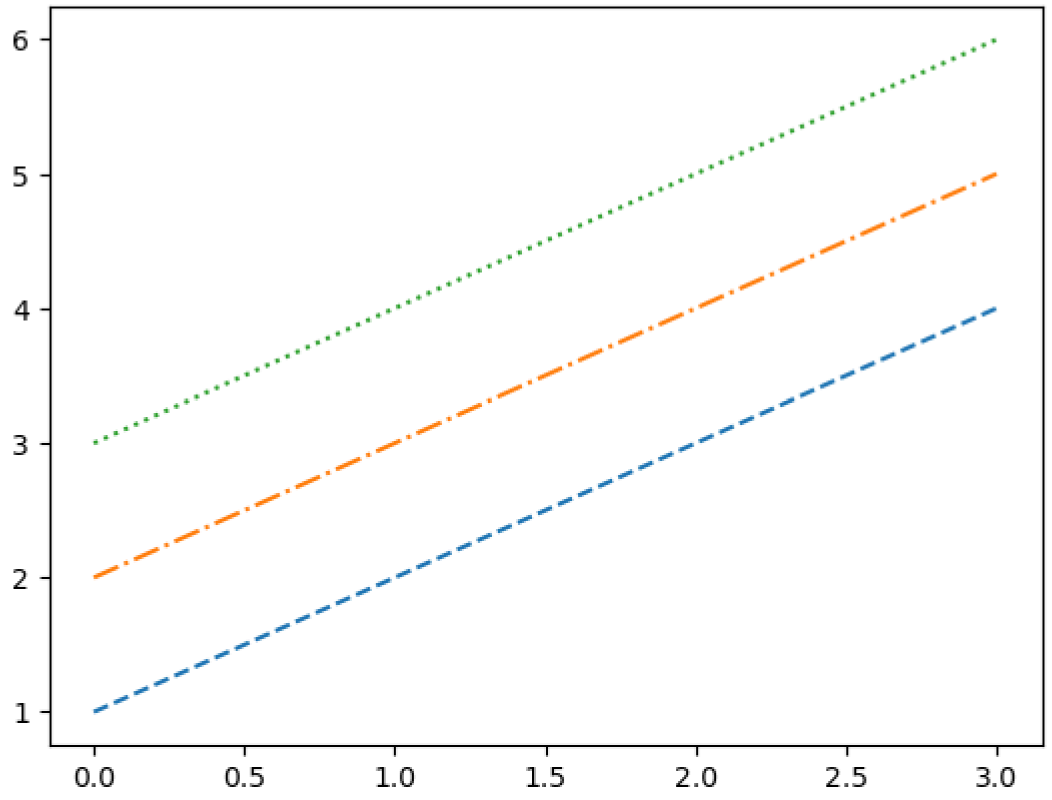
```
x16 = np.arange(1, 5)

plt.plot(x16, 'r')
plt.plot(x16+1, 'g')
plt.plot(x16+2, 'b')

plt.show()
```



Control line styles

In [44]:

```
x16 = np.arange(1, 5)

plt.plot(x16, '--', x16+1, '-.', x16+2, ':')

plt.show()
```


6
5
4
3
2
1
0.0
0.5
1.0
1.5
2.0
2.5
3.0