In [1]:
```
import numpy as np
```

In [2]:
```
import pandas as pd
```

In [3]:
```
import matplotlib.pyplot as plt
```

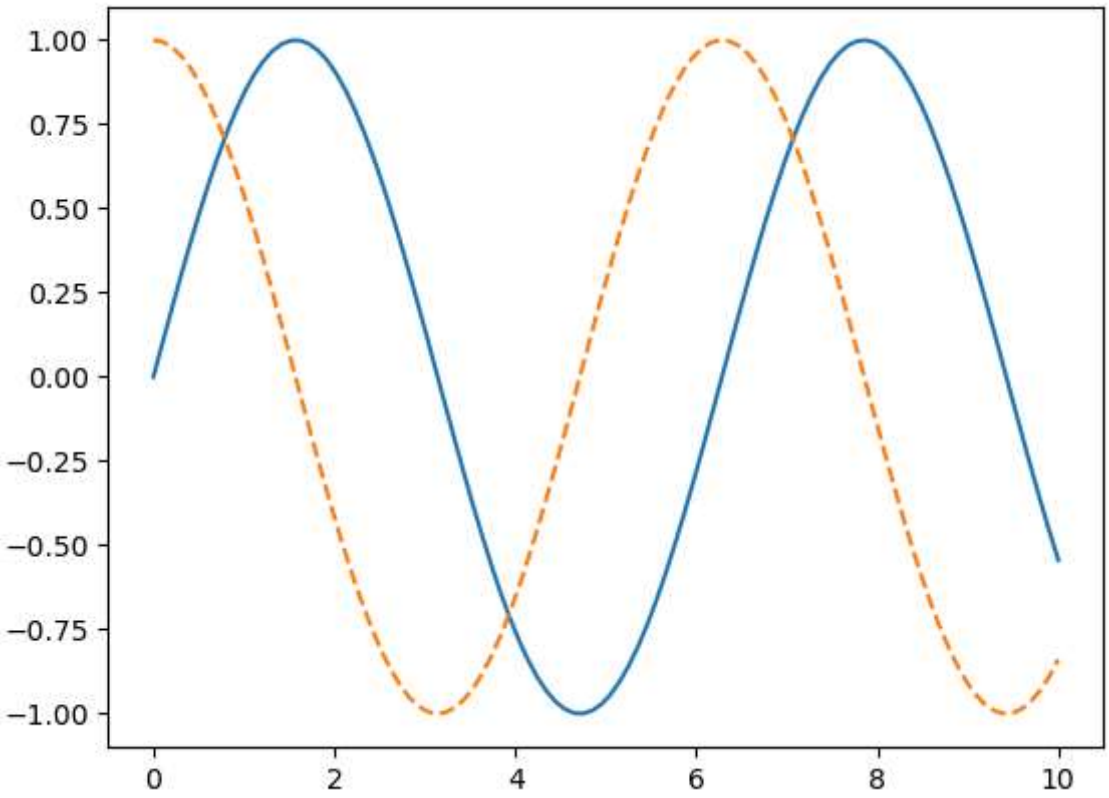
In [4]:
```
%matplotlib inline
```

In [5]:
```
x1=np.linspace(0,10,100)
```

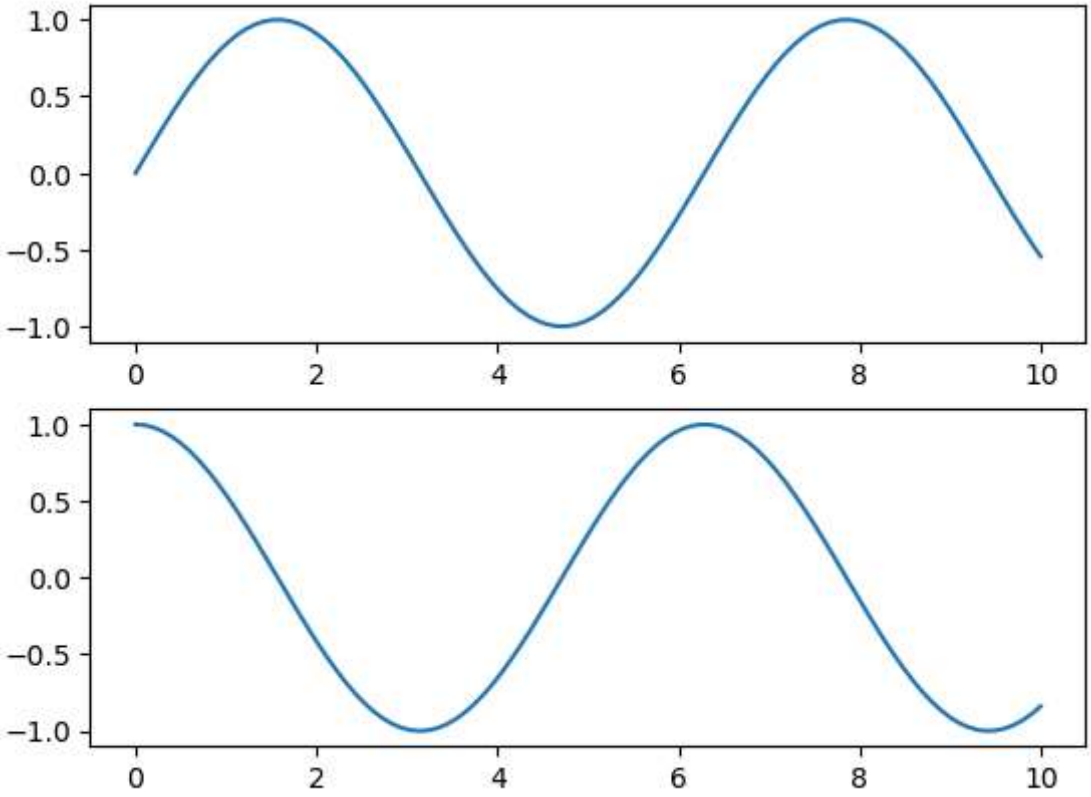
In [6]:
```
fig=plt.figure()
```

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

In [7]:
```
plt.plot(x1,np.sin(x1),'-')
plt.plot(x1,np.cos(x1),'--');
```



In [8]:
```
plt.figure()
plt.subplot(2,1,1)
plt.plot(x1,np.sin(x1))
plt.subplot(2,1,2)
plt.plot(x1,np.cos(x1));
```

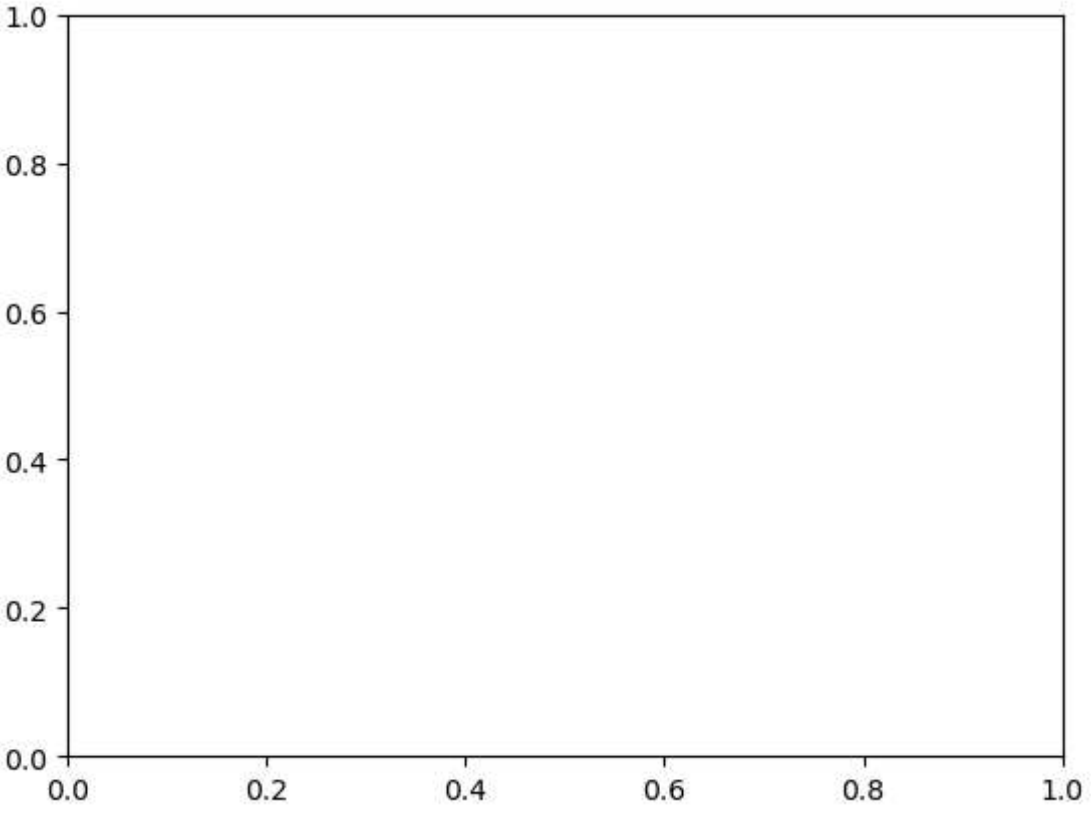


In [9]:
```
print(plt.gcf())
```

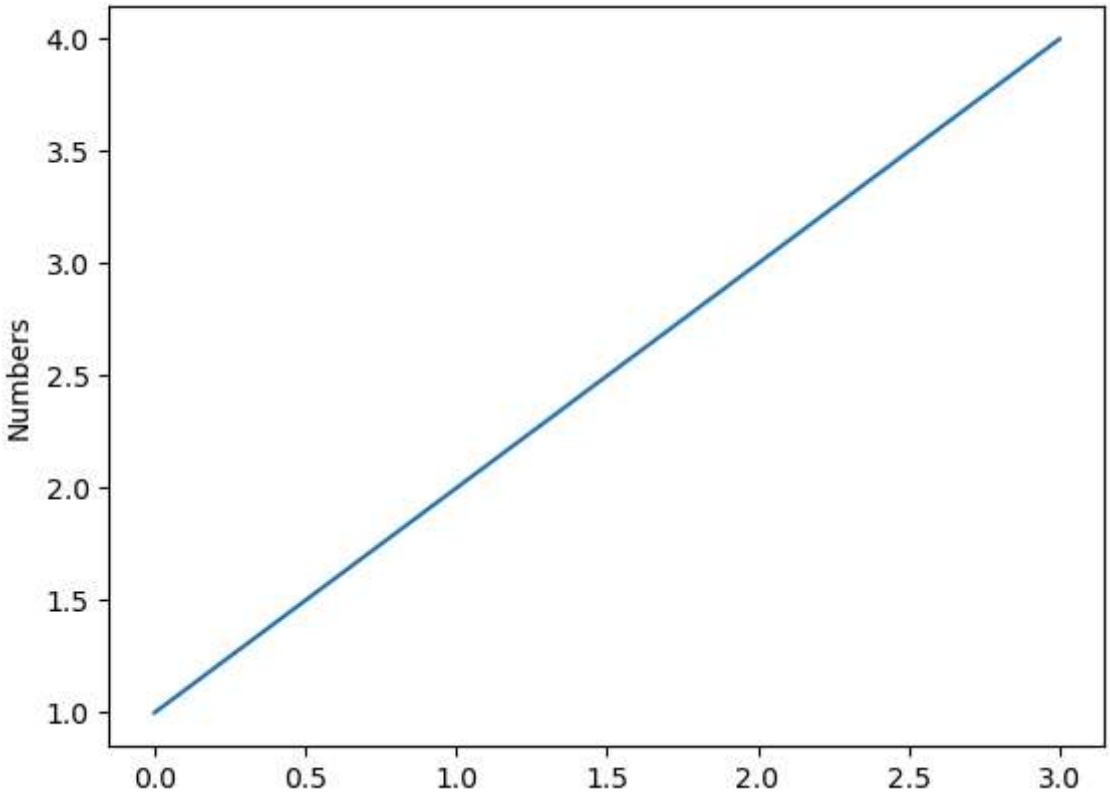
```
Figure(640x480)
<Figure size 640x480 with 0 Axes>
```

In [10]:
```
print(plt.gca())
```

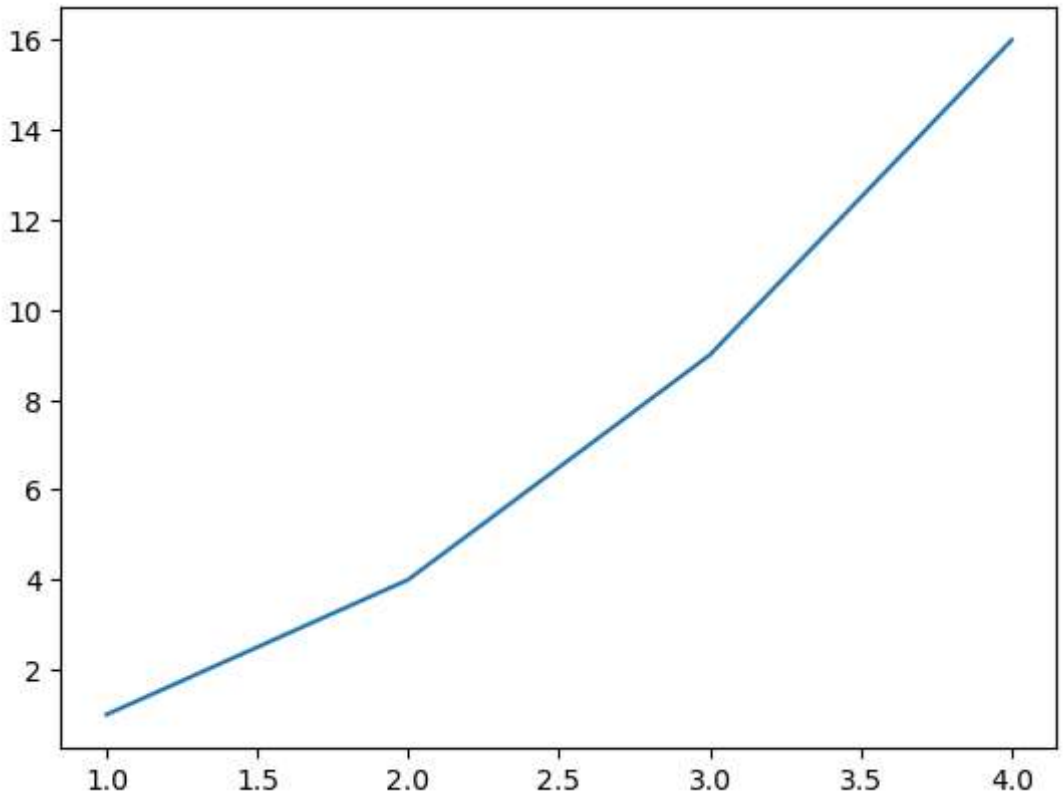
```
Axes(0.125,0.11;0.775x0.77)
```



In [11]:
```
plt.plot([1,2,3,4])
plt.ylabel('Numbers')
plt.show()
```

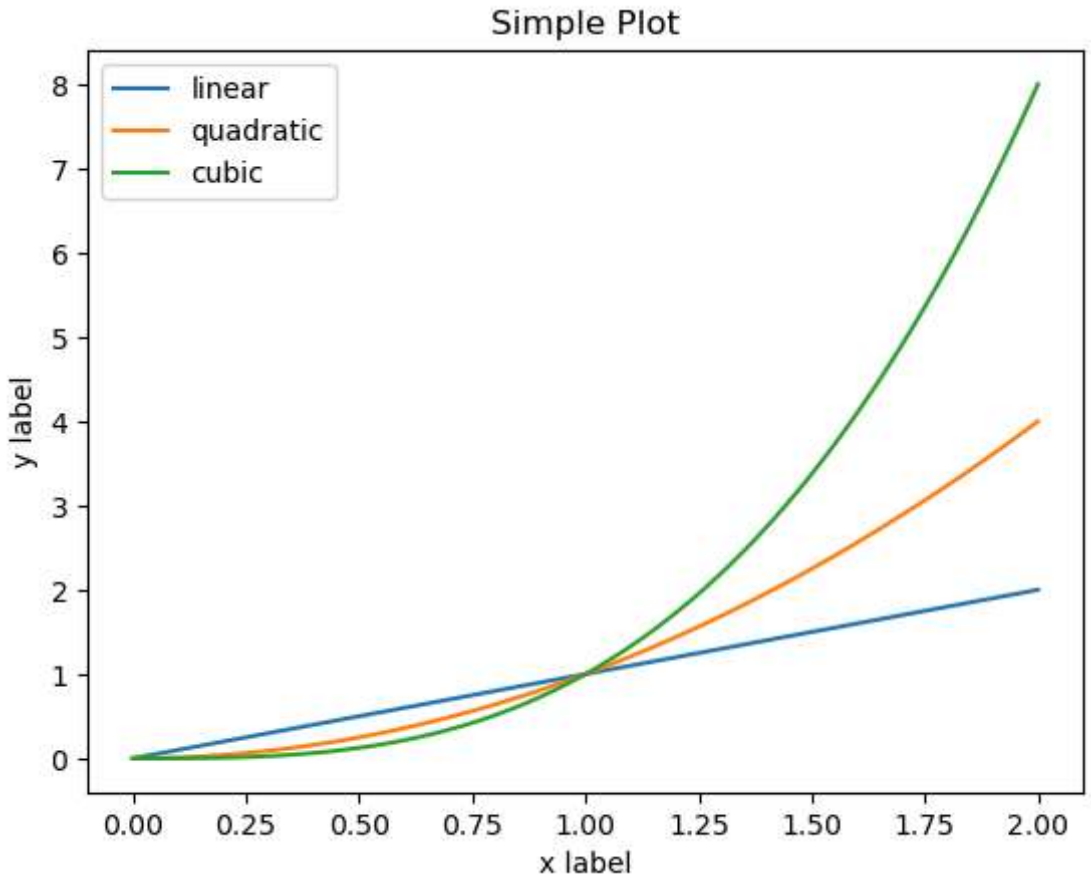


In [12]:
```
plt.plot([1,2,3,4],[1,4,9,16])
plt.show()
```

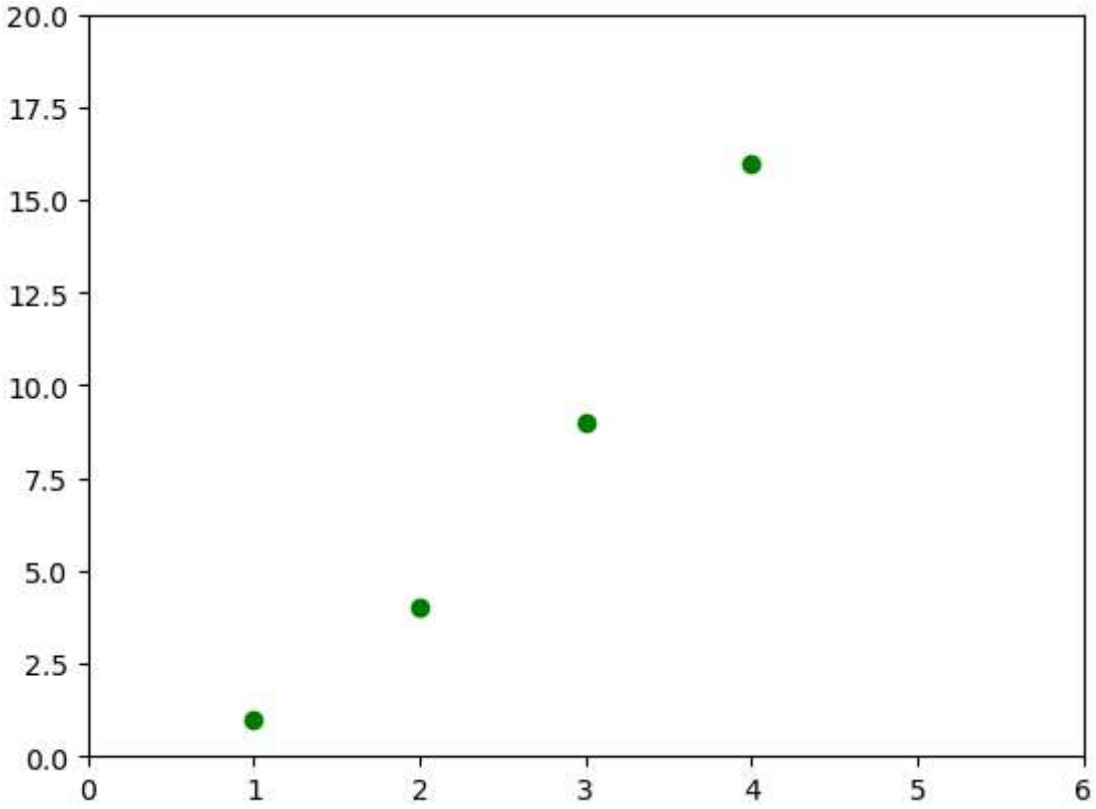


In [13]:
```
x=np.linspace(0,2,100)
```

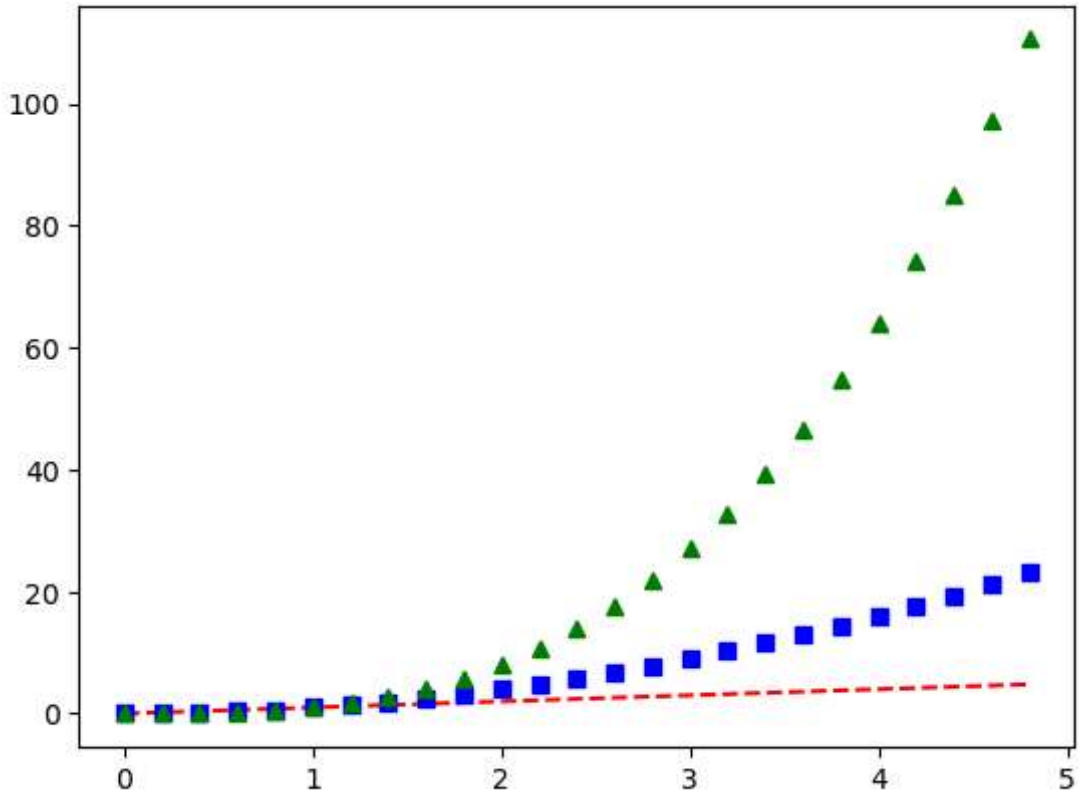
```
plt.plot(x,x,label='linear')
plt.plot(x,x**2,label='quadratic')
plt.plot(x,x**3,label='cubic')

plt.xlabel('x label')
plt.ylabel('y label')

plt.title("Simple Plot")


plt.legend()

plt.show()
```



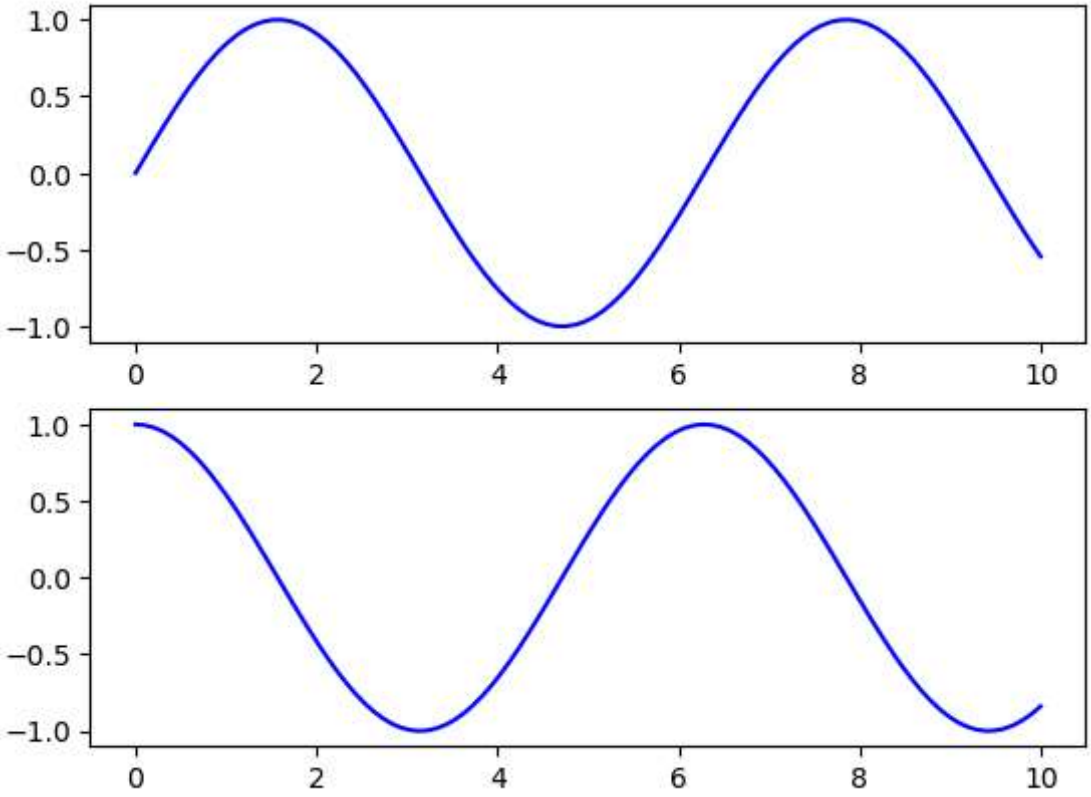
In [14]:
```
plt.plot([1,2,3,4],[1,4,9,16],'go')
plt.axis([0,6,0,20])
plt.show()
```



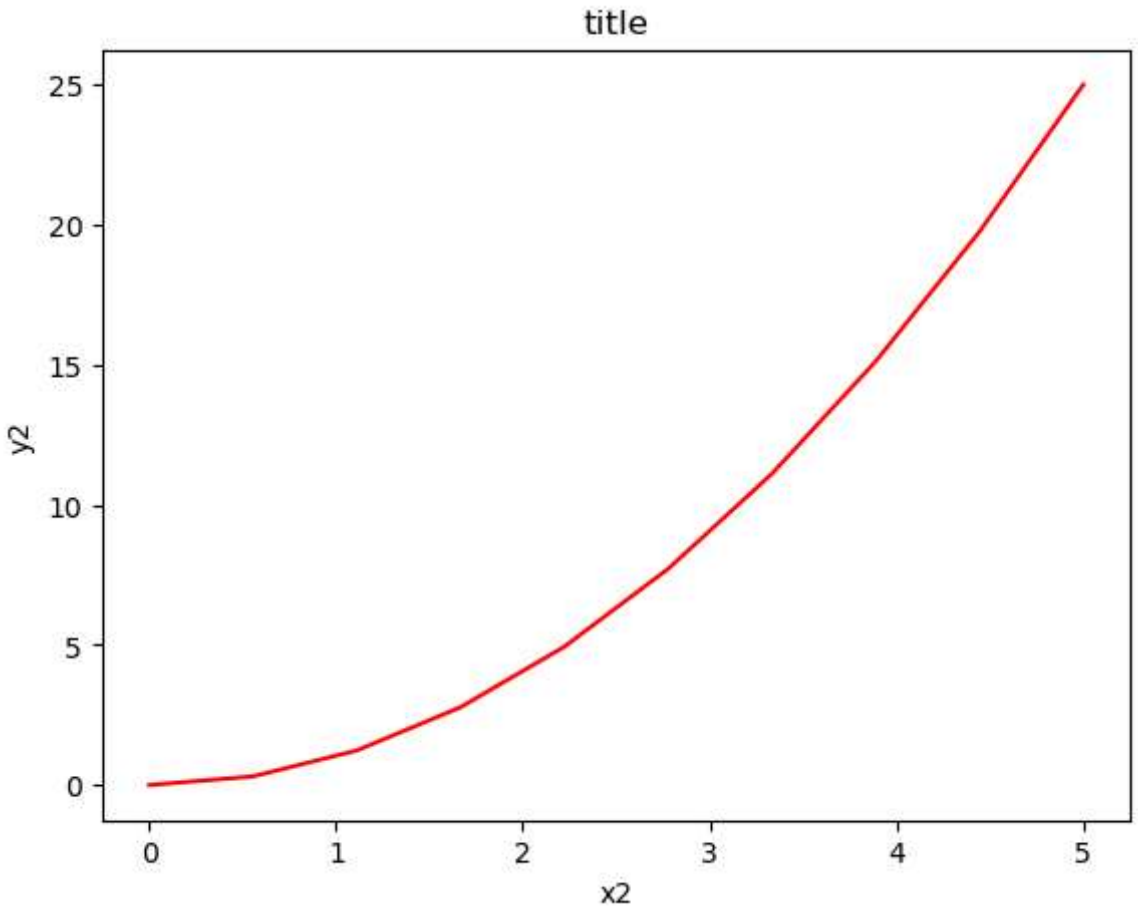
In [15]:
```
t=np.arange(0.,5.,0.2)

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



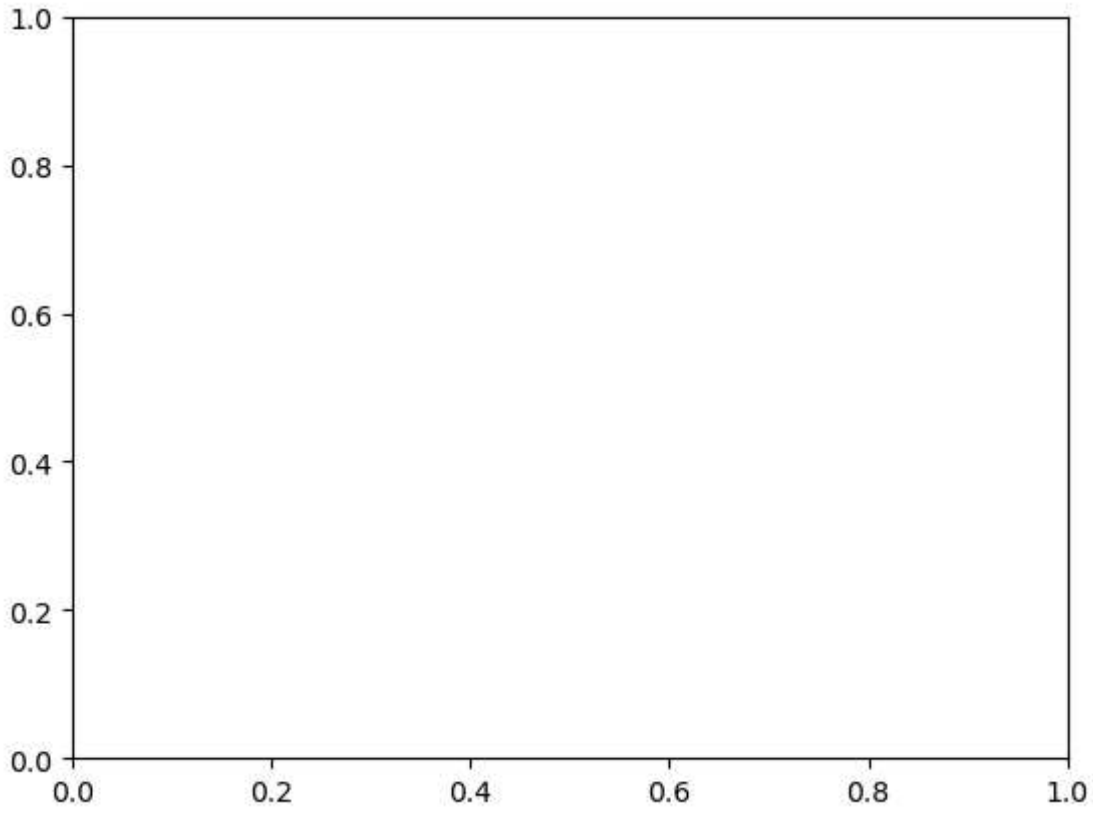
In [16]:
```
fig,ax=plt.subplots(2)

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



In [17]:
```
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,'r')

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

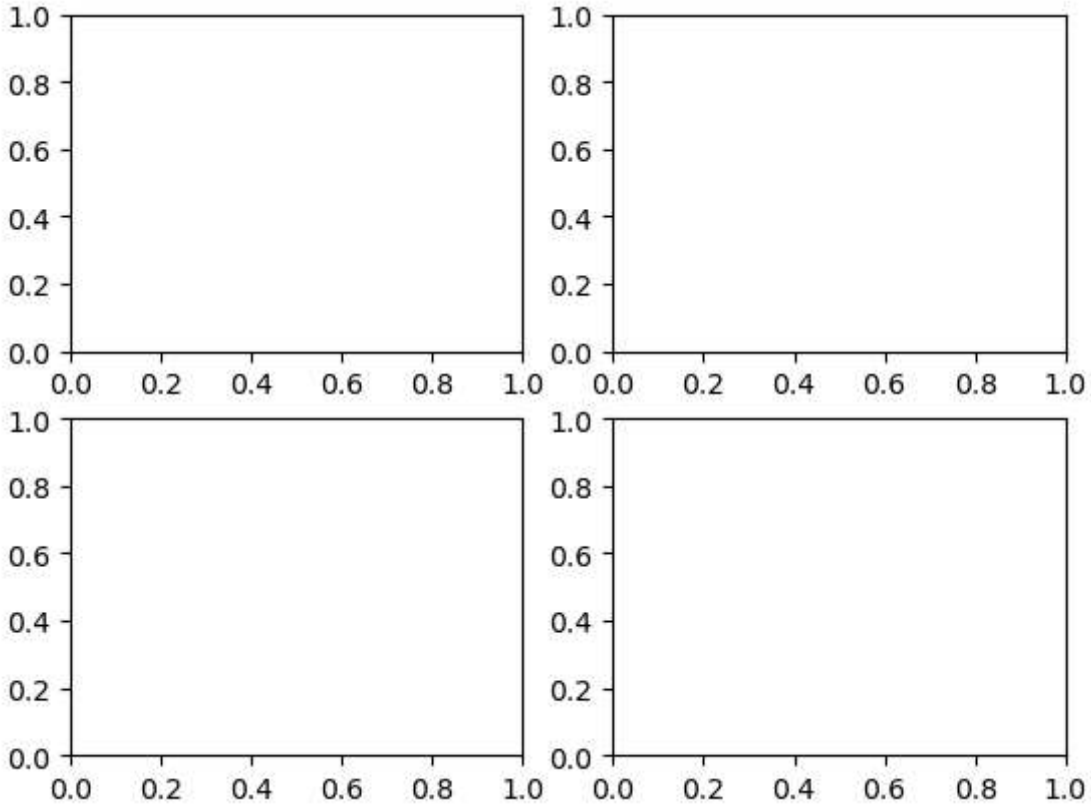


In [18]:
```
fig=plt.figure()
ax=plt.axes()
```

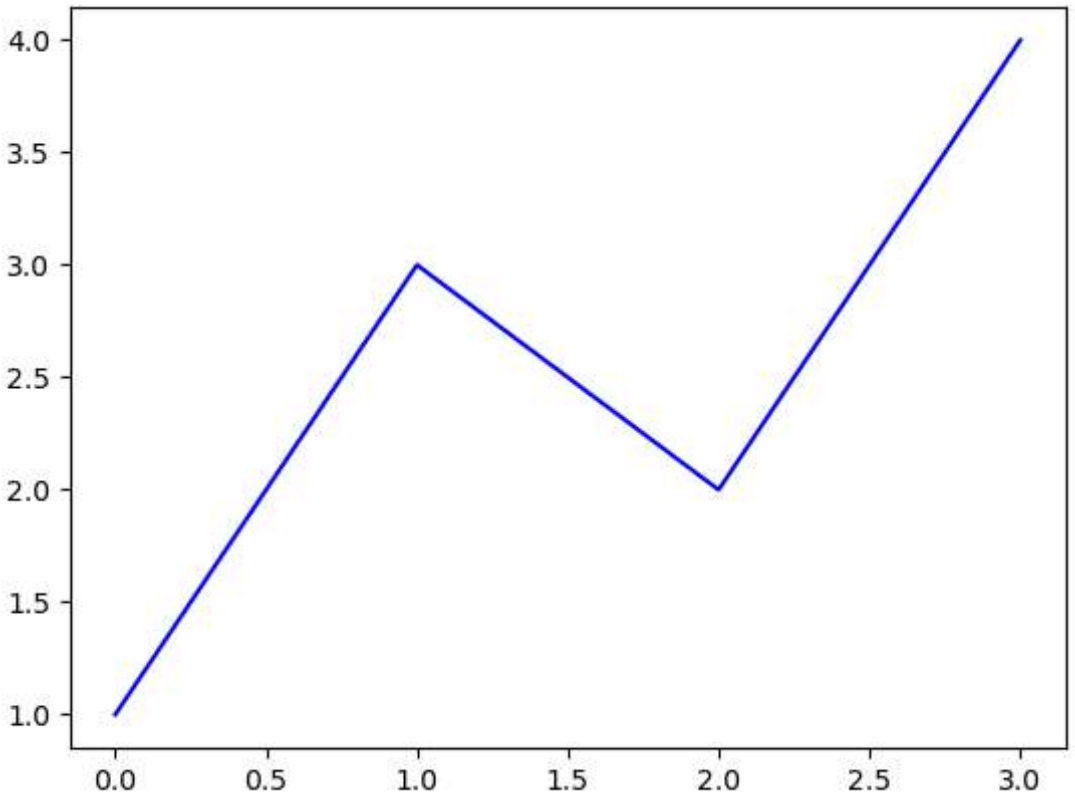


In [19]:
```
fig=plt.figure()

ax1=fig.add_subplot(2,2,1)
```

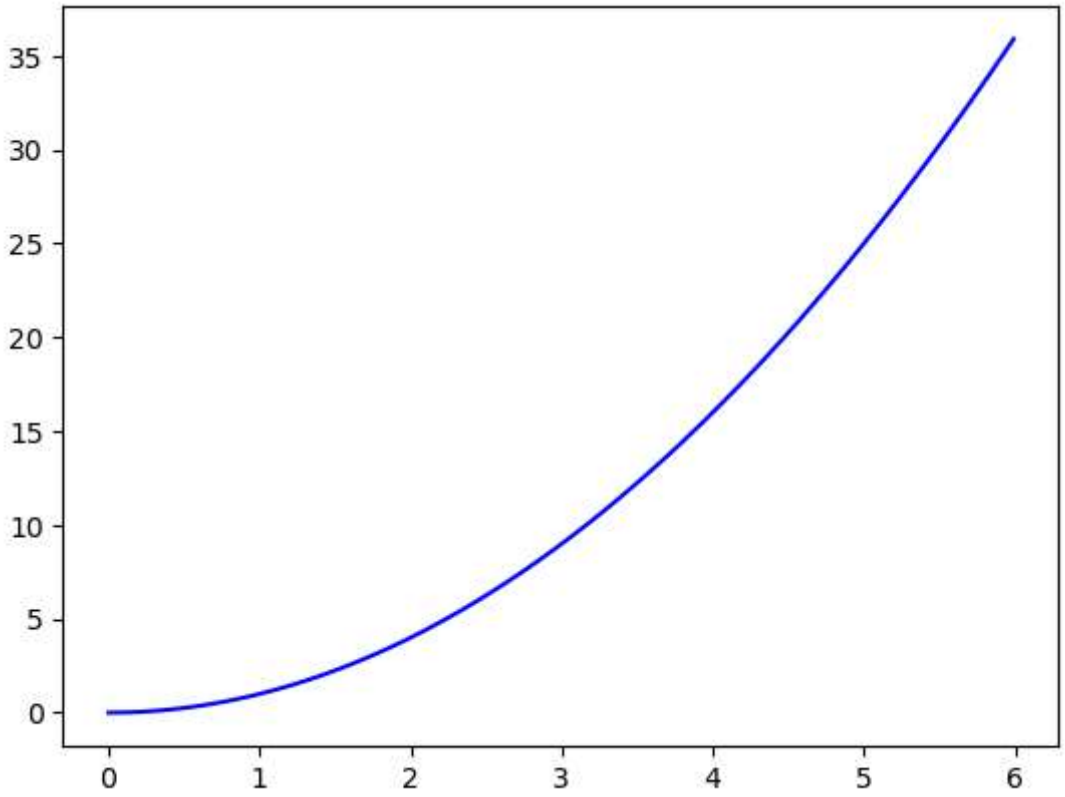
```
ax2=fig.add_subplot(2,2,2)
ax3=fig.add_subplot(2,2,3)
ax4=fig.add_subplot(2,2,4)
```



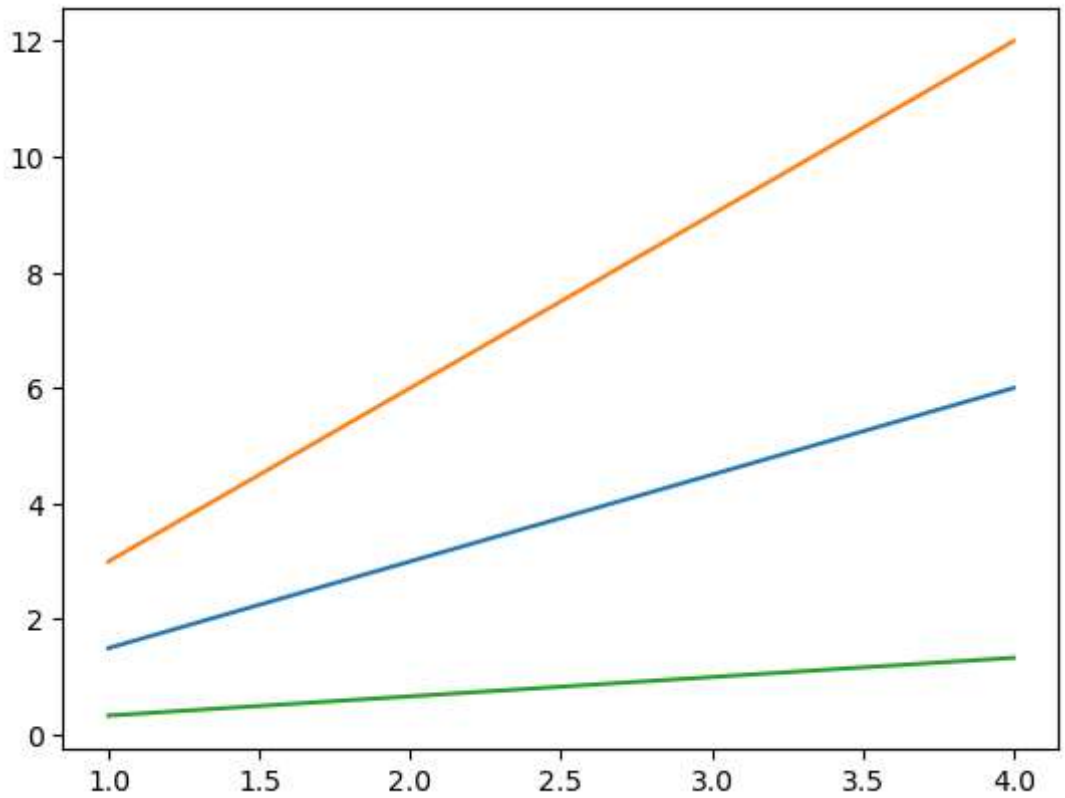
In [20]:
```
plt.plot([1,3,2,4],'b-')

plt.show()
```



In [21]:
```
x3=np.arange(0.0,6.0,0.01)
plt.plot(x3,[xi**2 for xi in x3],'b-')
plt.show()
```



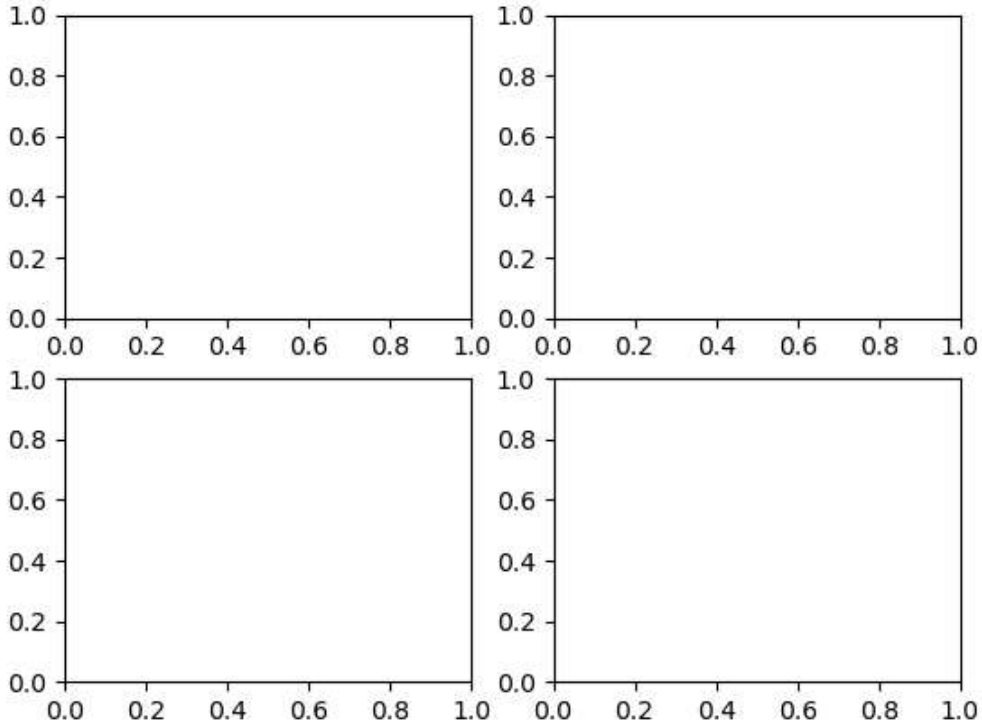
In [22]:
```
x4=range(1,5)
plt.plot(x4,[xi*1.5 for xi in x4])
plt.plot(x4,[xi*3 for xi in x4])
plt.plot(x4,[xi/3.0 for xi in x4])
plt.show()
```



In [23]:
```
fig.savefig('plot1.png')
```

In [24]:
```
from IPython.display import Image
Image('plot1.png')
```

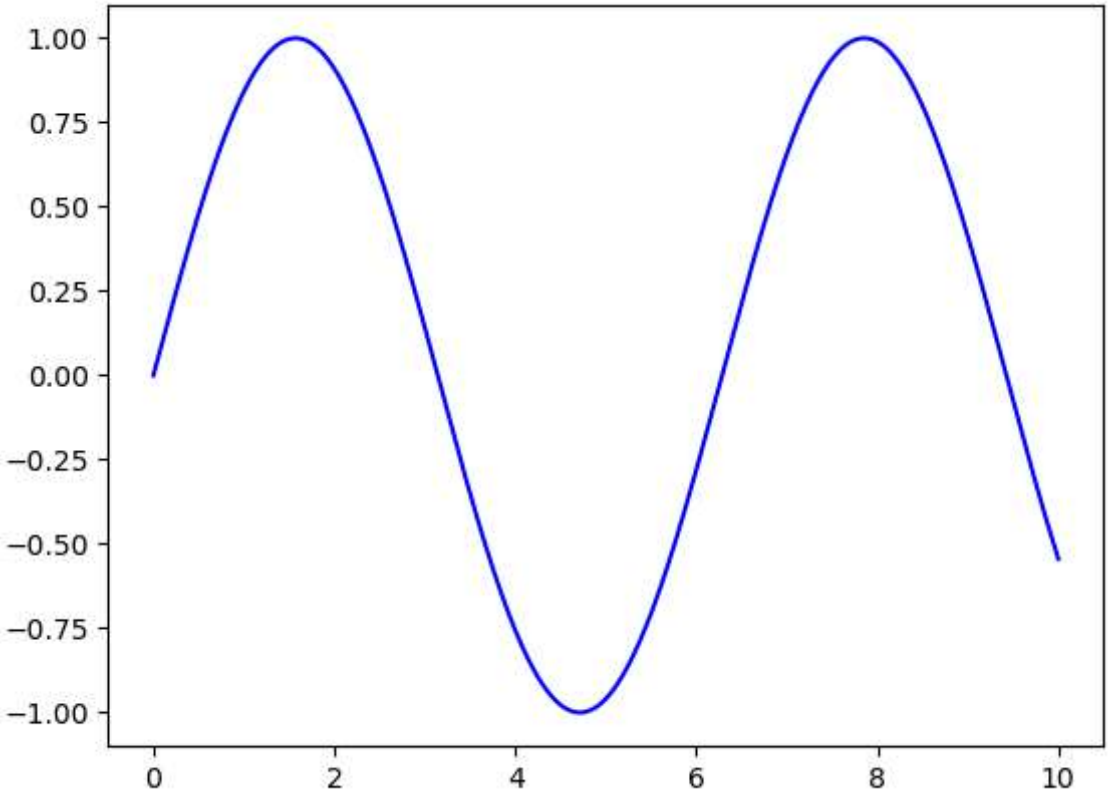
Out[24]:



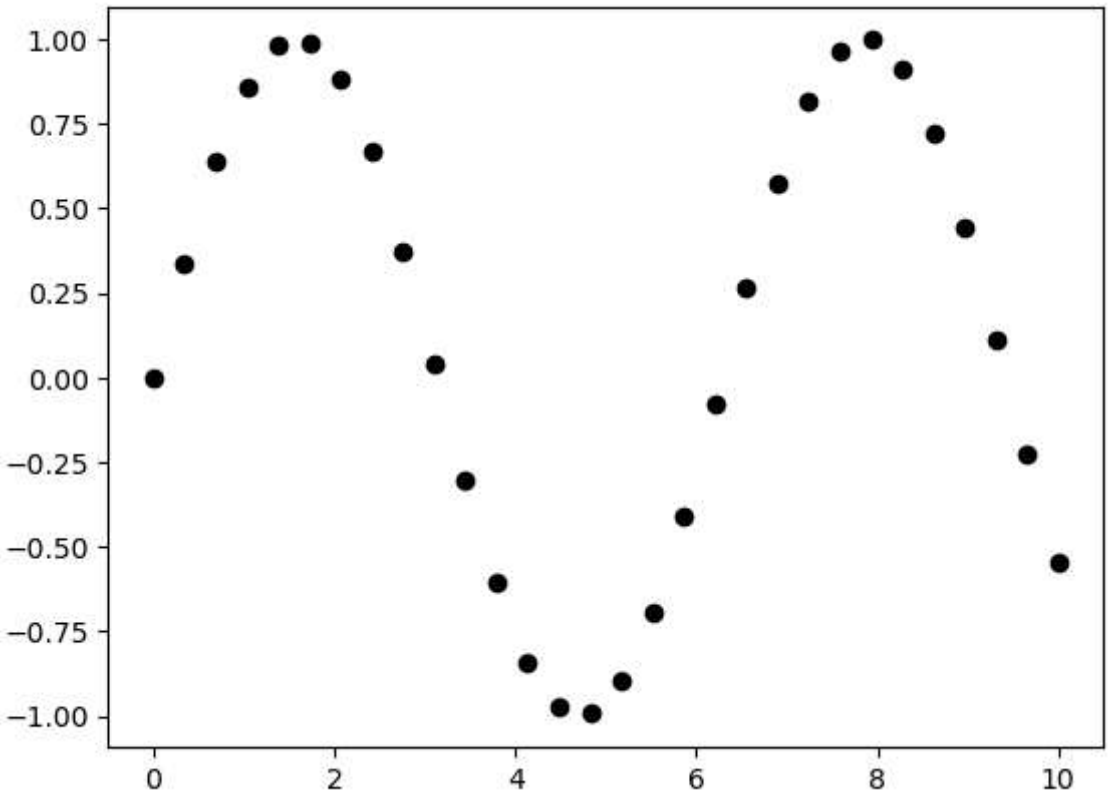
In [25]:
```
fig.canvas.get_supported_filetypes()
```

Out[25]:
```
{'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'}
```

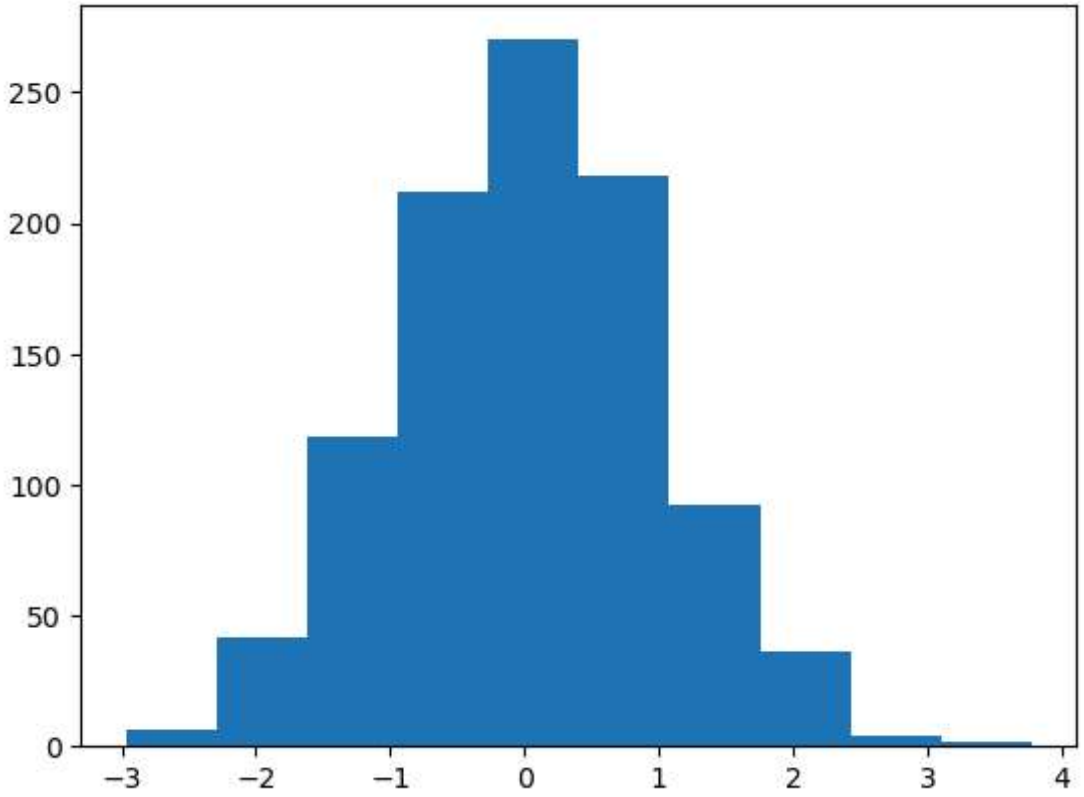
In [26]:
```
fig=plt.figure()
ax=plt.axes()
x5=np.linspace(0,10,1000)

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



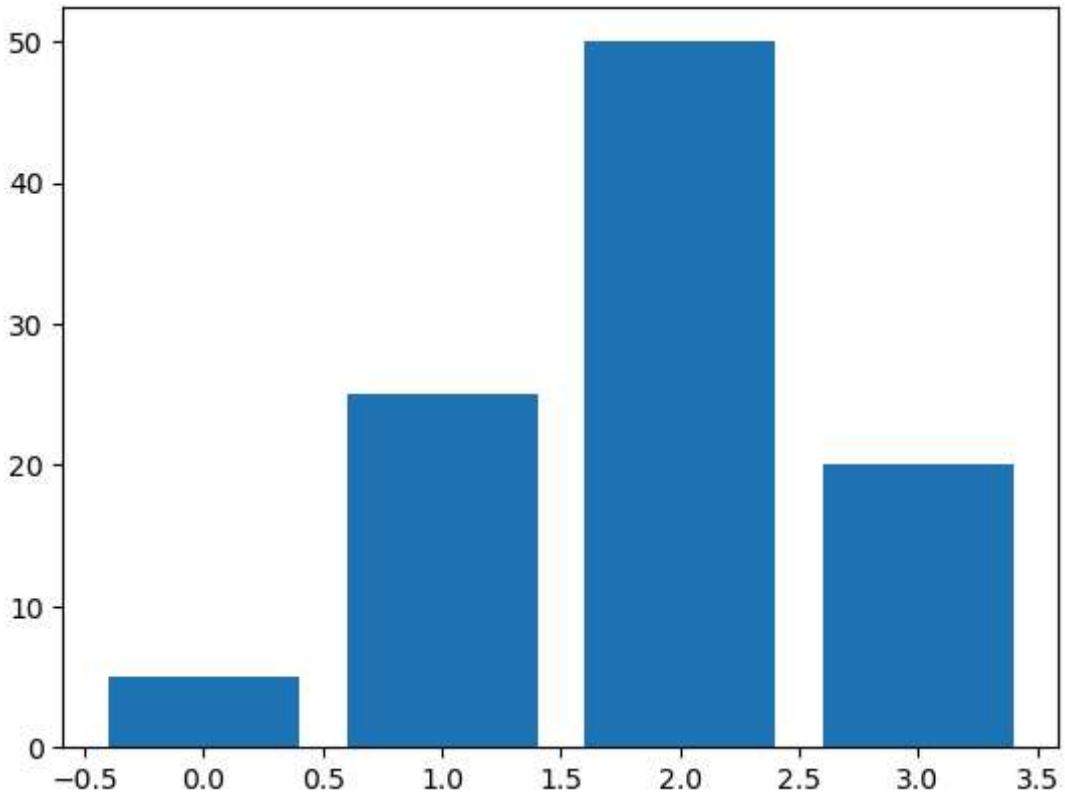
In [27]:
```
x7=np.linspace(0,10,30)
y7=np.sin(x7)
plt.plot(x7,y7,'o',color='black');
```



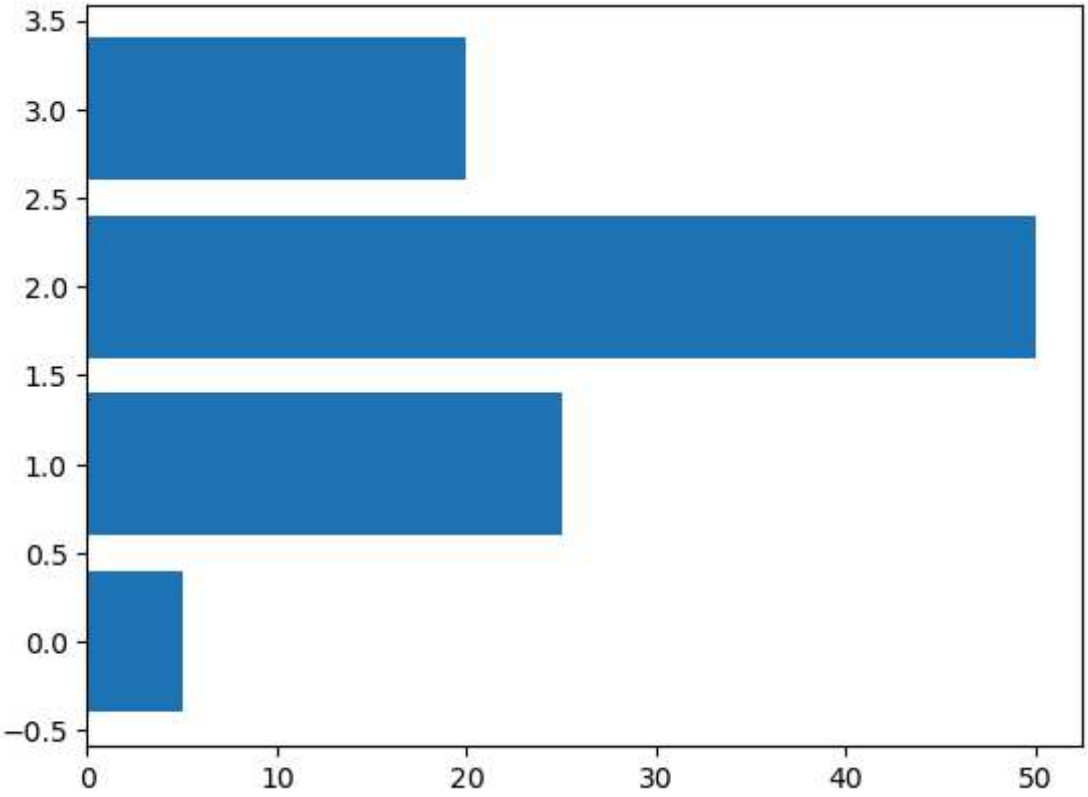
In [28]:
```
data1=np.random.randn(1000)
plt.hist(data1);
```



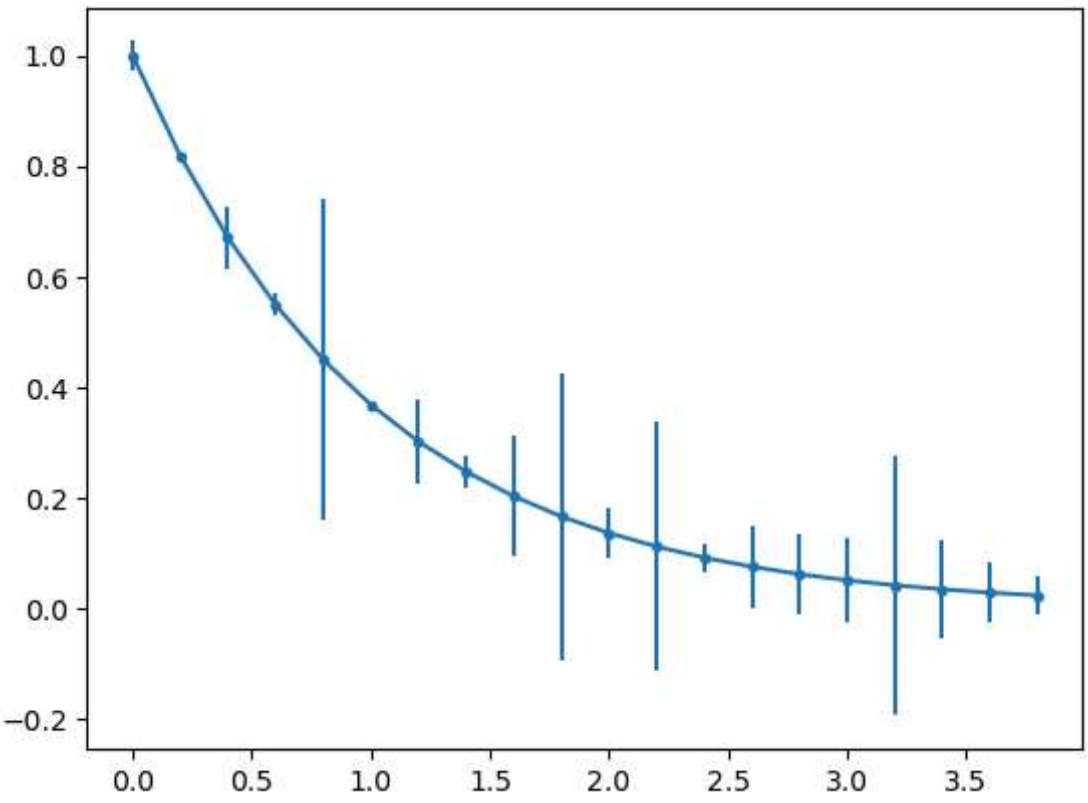
In [29]:
```
data2=[5.,25.,50.,20.]
plt.bar(range(len(data2)),data2)
plt.show()
```



In [30]:
```
data2=[5.,25.,50.,20.]
plt.barh(range(len(data2)),data2)
plt.show()
```

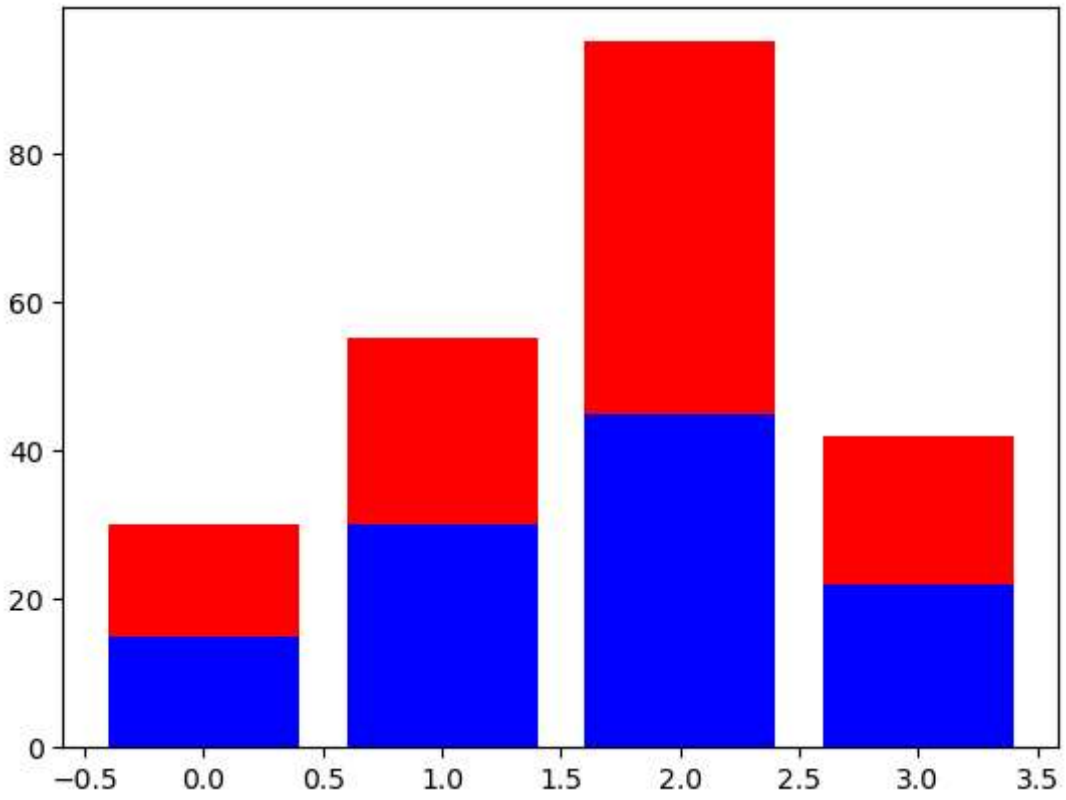


In [31]:
```
x9=np.arange(0,4,0.2)
y9=np.exp(-x9)
e1=0.1*np.abs(np.random.randn(len(y9)))
plt.errorbar(x9,y9,yerr=e1,fmt='.-')
plt.show();
```

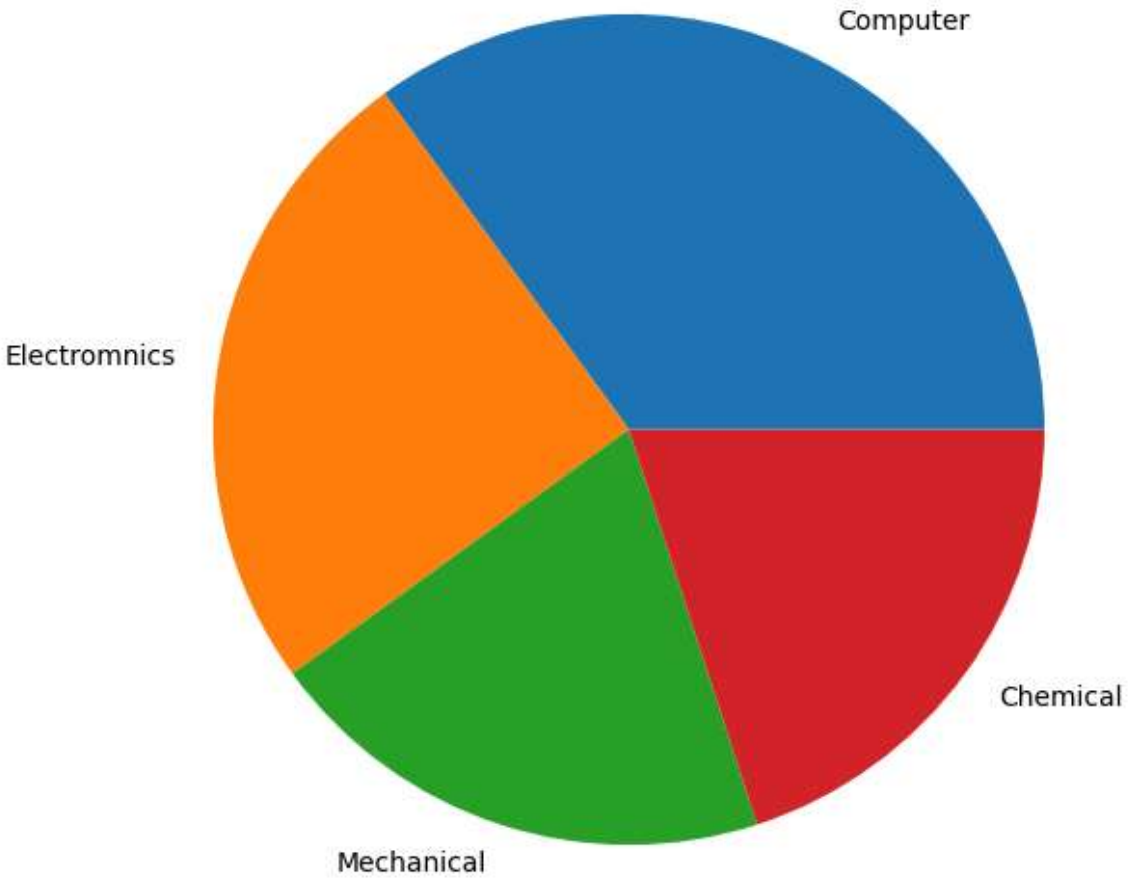


In [32]:
```
A=[15.,30.,45.,22.]
B=[15.,25.,50.,20.]
z2=range(4)
```

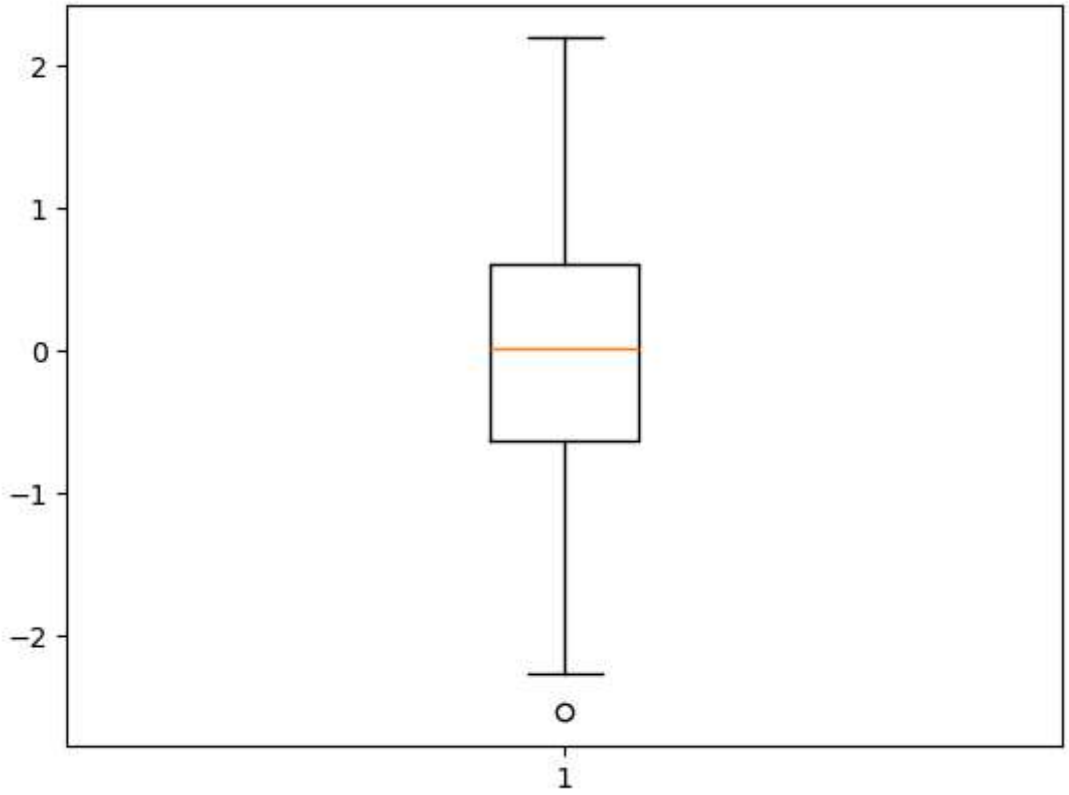
```
plt.bar(z2,A,color='b')
plt.bar(z2,B,color='r',bottom=A)
plt.show()
```



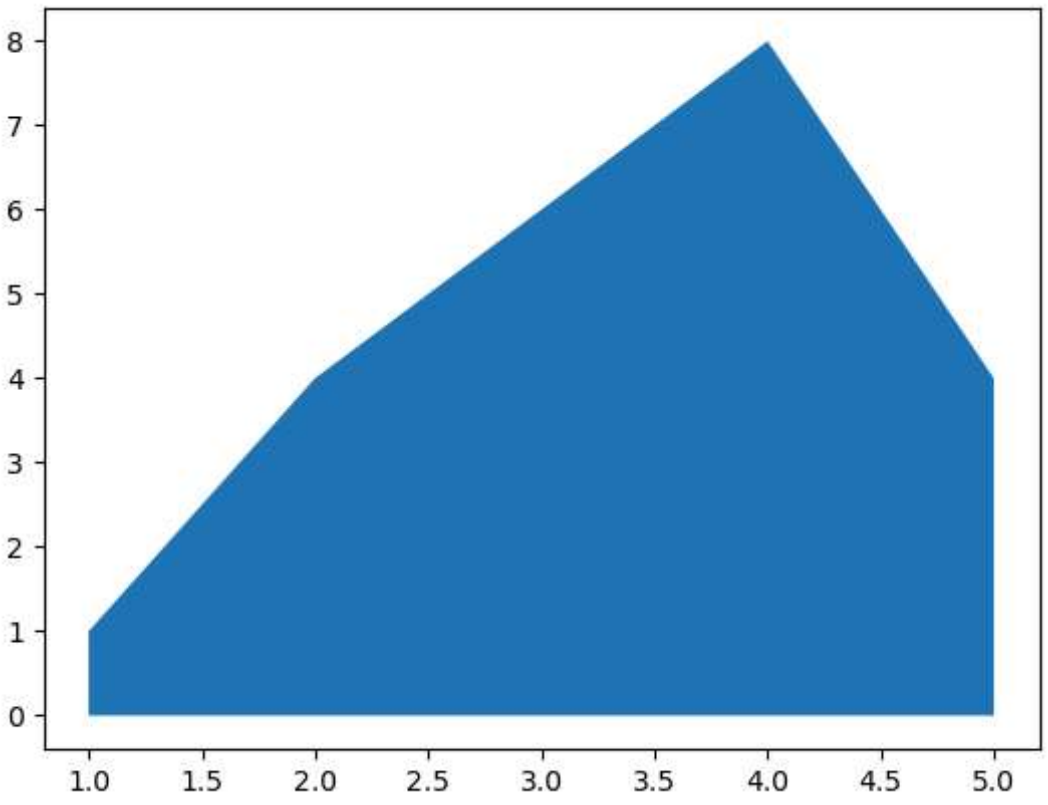
In [33]:
```
plt.figure(figsize=(7,7))
x10=[35,25,20,20]
labels=['Computer','Electromnics','Mechanical','Chemical']
plt.pie(x10,labels=labels);
plt.show()
```



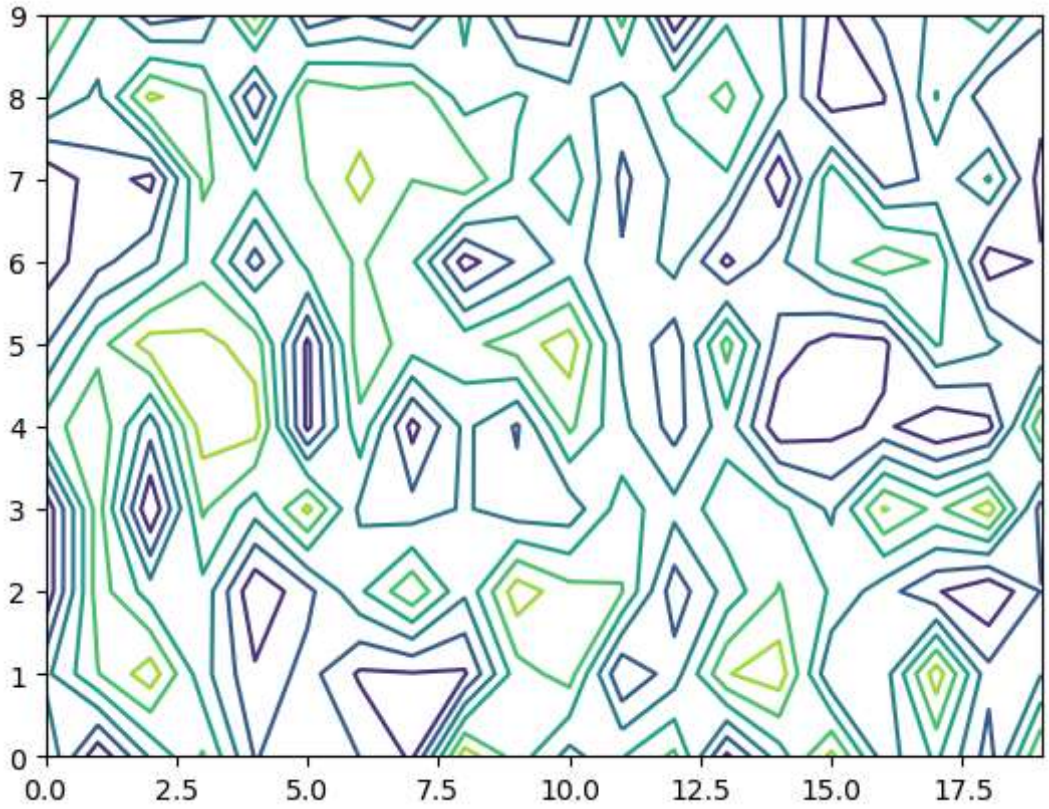
In [34]:
```
data3=np.random.randn(100)
plt.boxplot(data3)
plt.show();
```



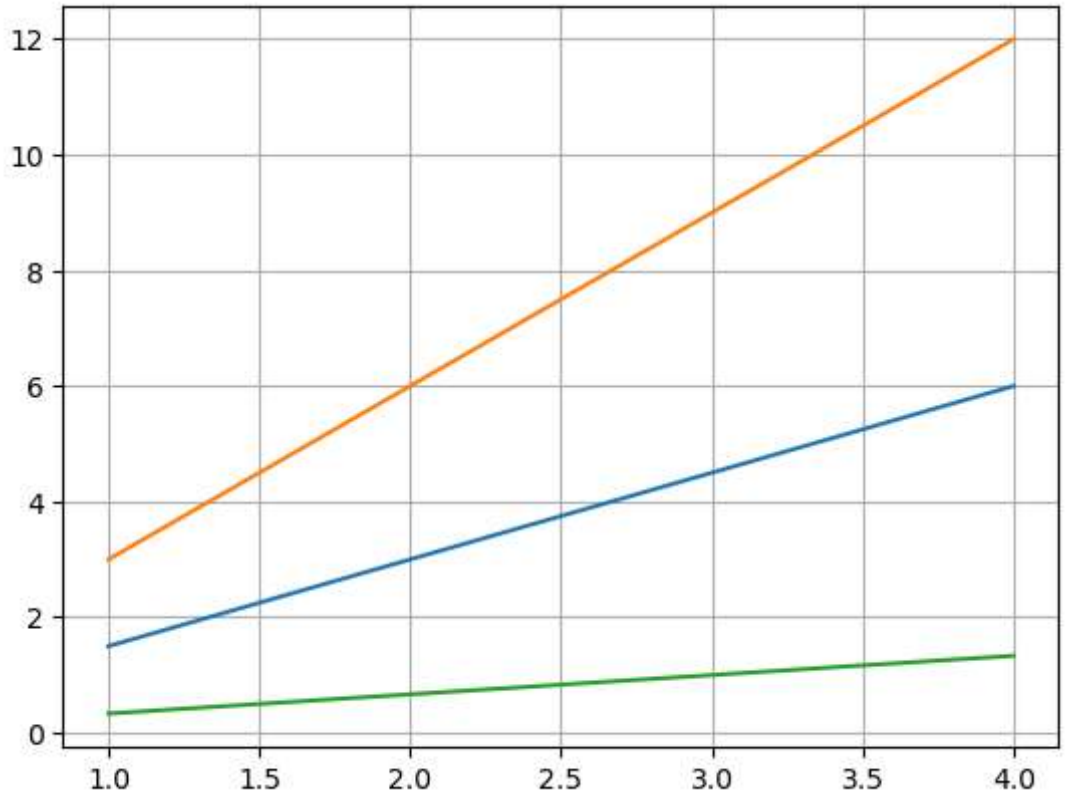
In [35]:
```
x12=range(1,6)
y12=[1,4,6,8,4]

plt.fill_between(x12,y12)
plt.show()
```



In [36]:
```
matrix1=np.random.rand(10,20)
cp=plt.contour(matrix1)
plt.show()
```

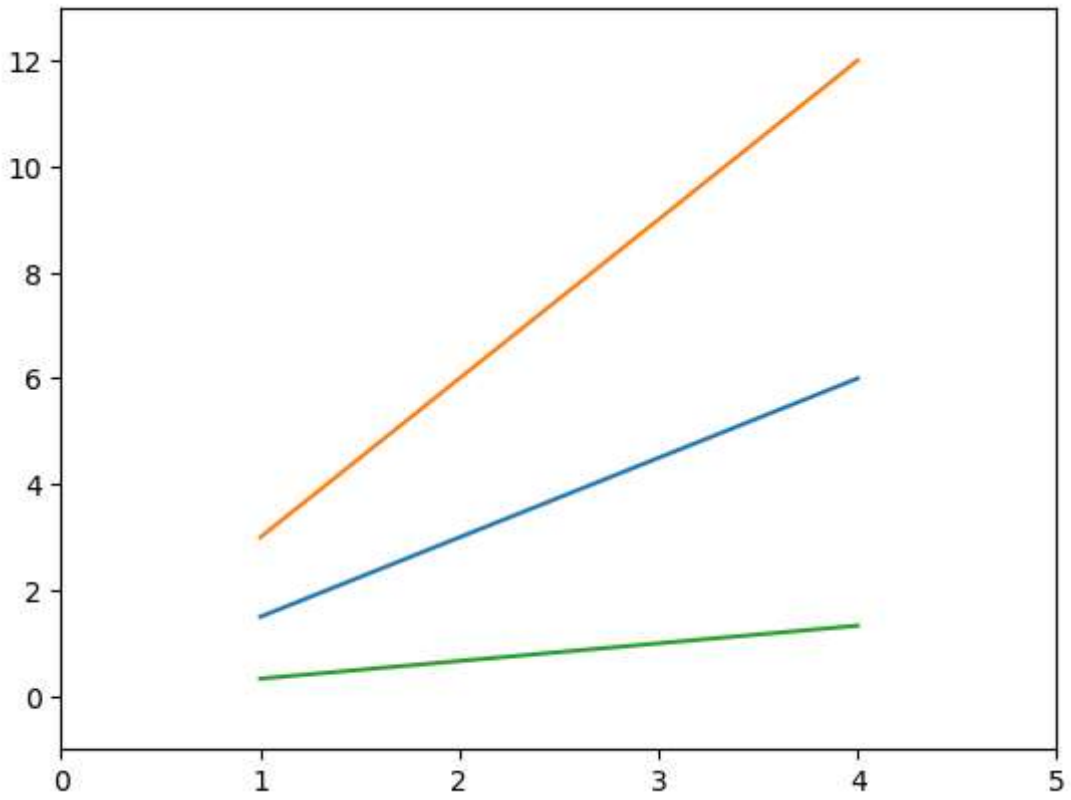


In [37]:
```
x15=np.arange(1,5)
plt.plot(x15,x15*1.5,x15,x15*3.0,x15,x15/3.0)
plt.grid(True)
plt.show()
```



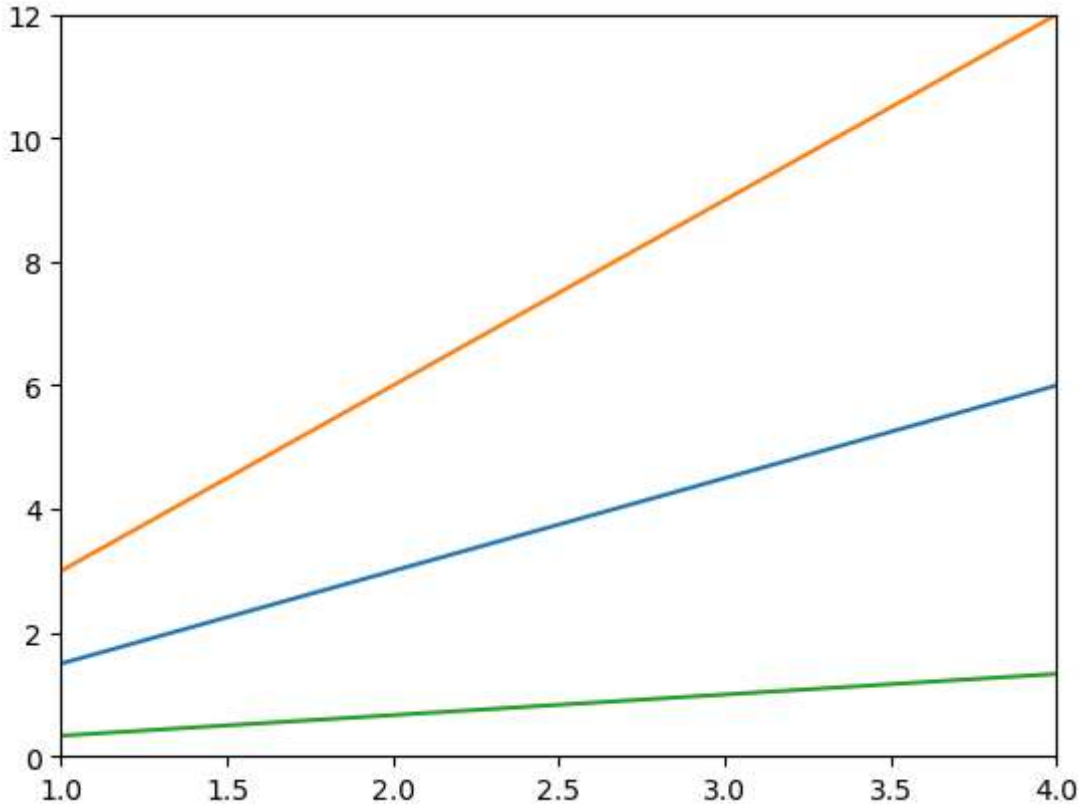
In [38]:
```
x15=np.arange(1,5)
plt.plot(x15,x15*1.5,x15,x15*3.0,x15,x15/3.0)
plt.axis()
```

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

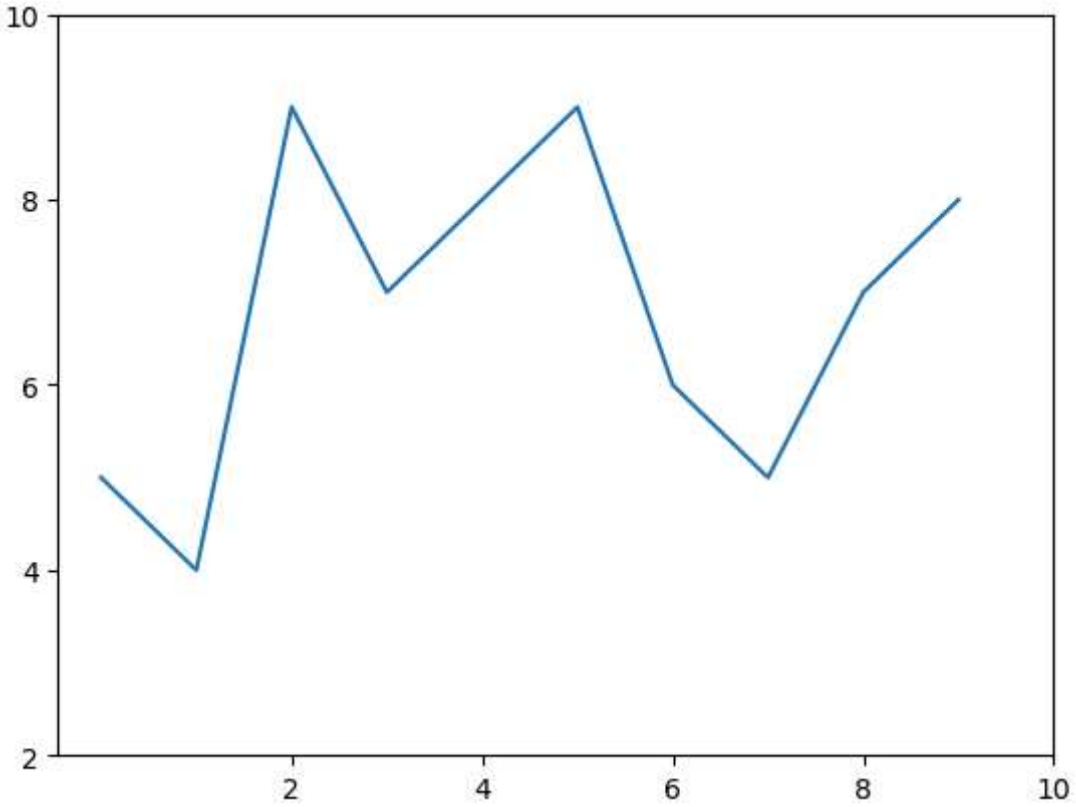


In [39]:
```
x15=np.arange(1,5)
plt.plot(x15,x15*1.5,x15,x15*3.0,x15,x15/3,0)
plt.xlim([1.0,4.0])
plt.ylim([0.0,12.0])
```

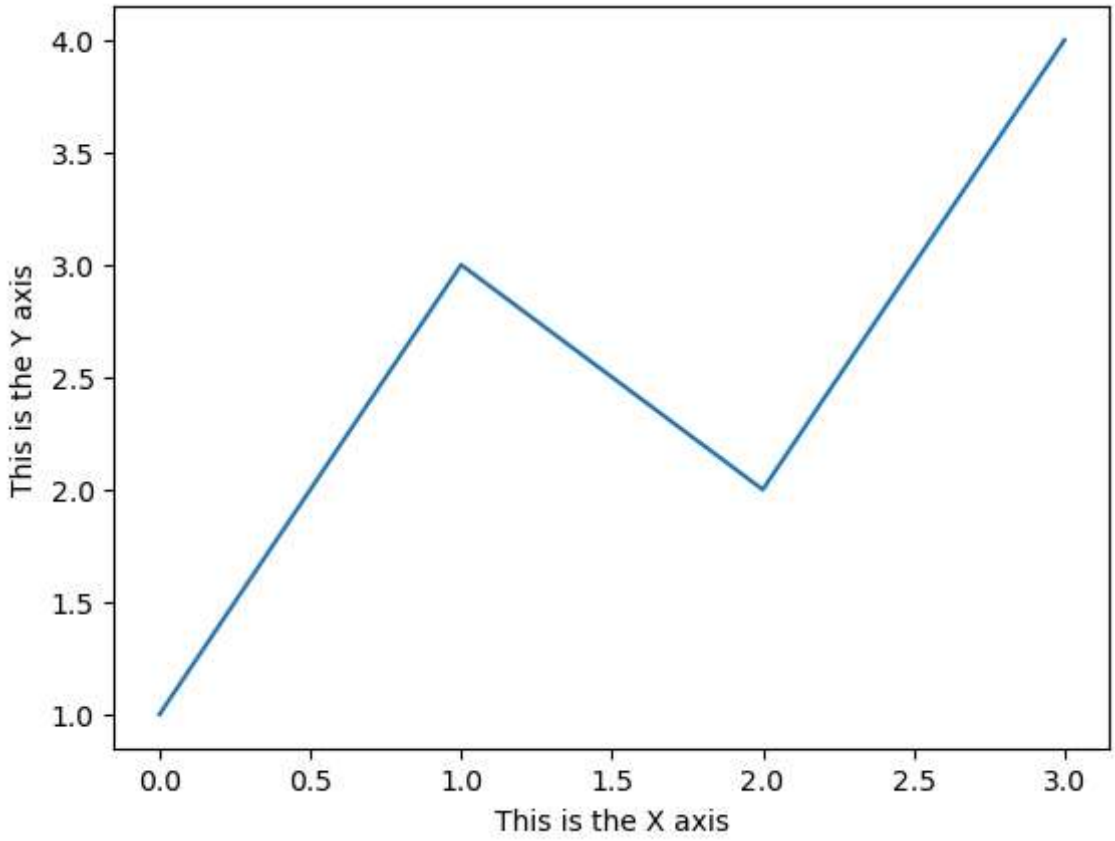
Out[39]: (0.0, 12.0)



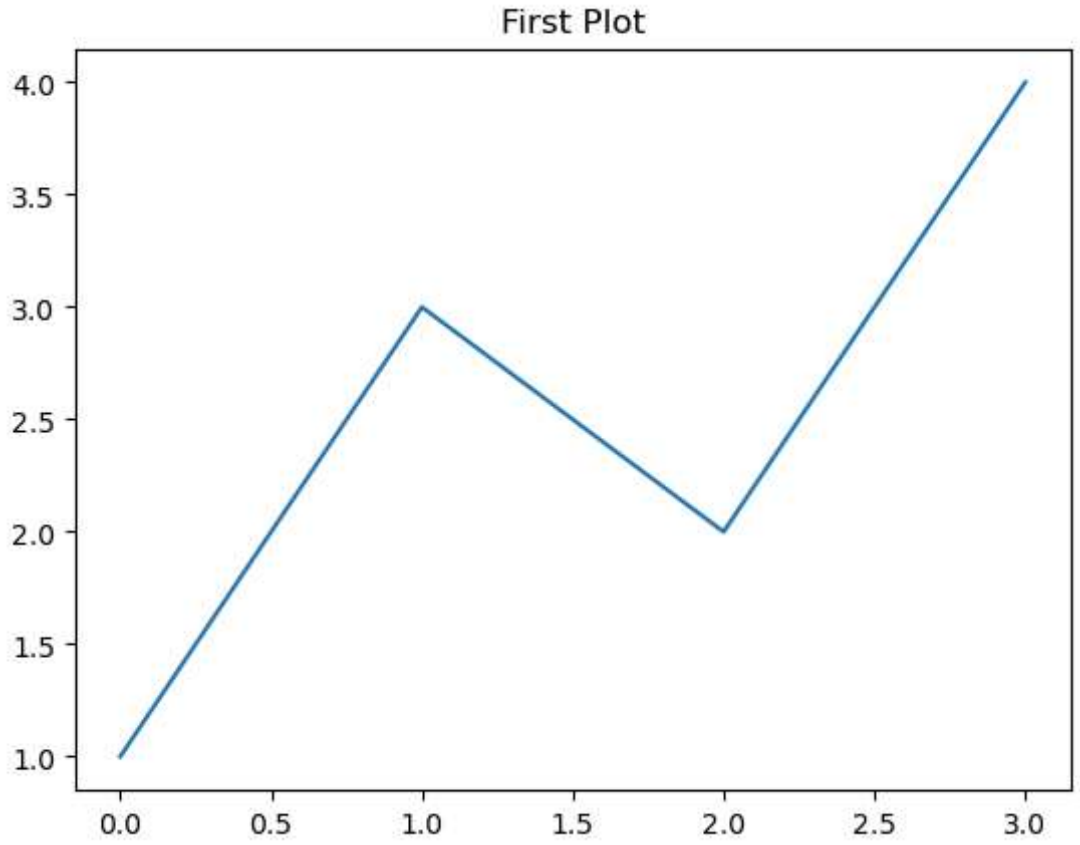
In [40]:
```
u=[5,4,9,7,8,9,6,5,7,8]
plt.plot(u)
plt.xticks([2,4,6,8,10])
plt.yticks([2,4,6,8,10])
plt.show()
```



In [41]:
```
plt.plot([1,3,2,4])
plt.xlabel('This is the X axis')
plt.ylabel('This is the Y axis')
plt.show()
```

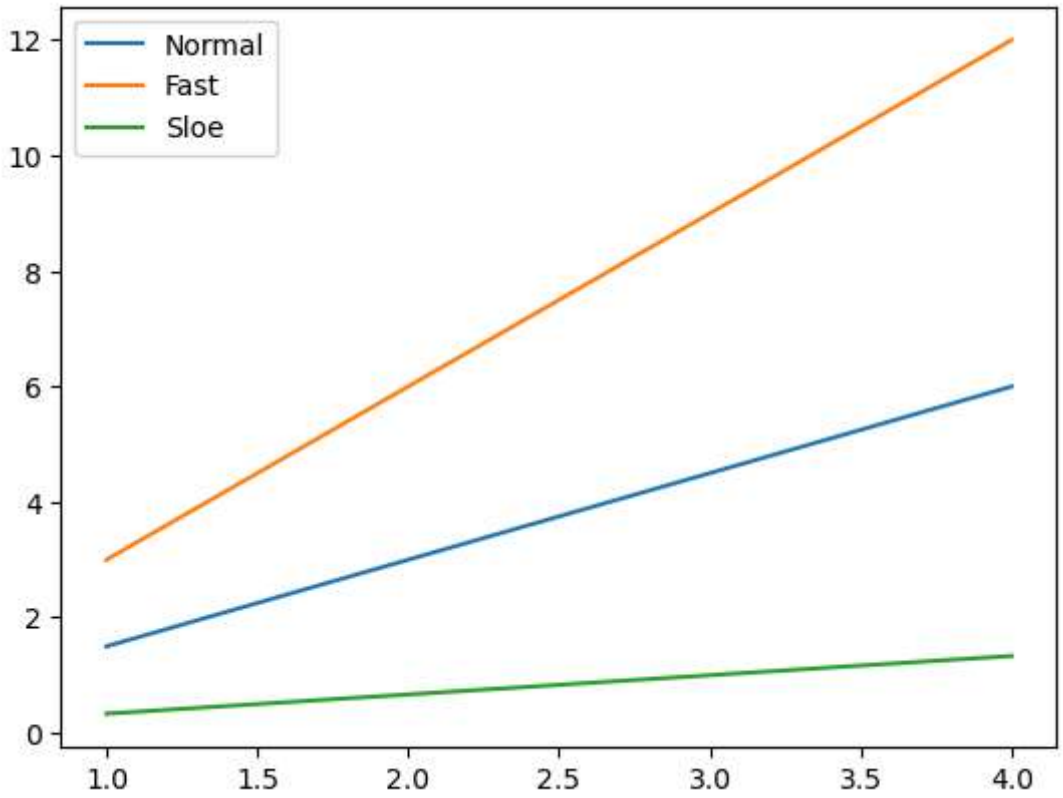


In [42]:
```
plt.plot([1,3,2,4])
plt.title('First Plot')
plt.show()
```

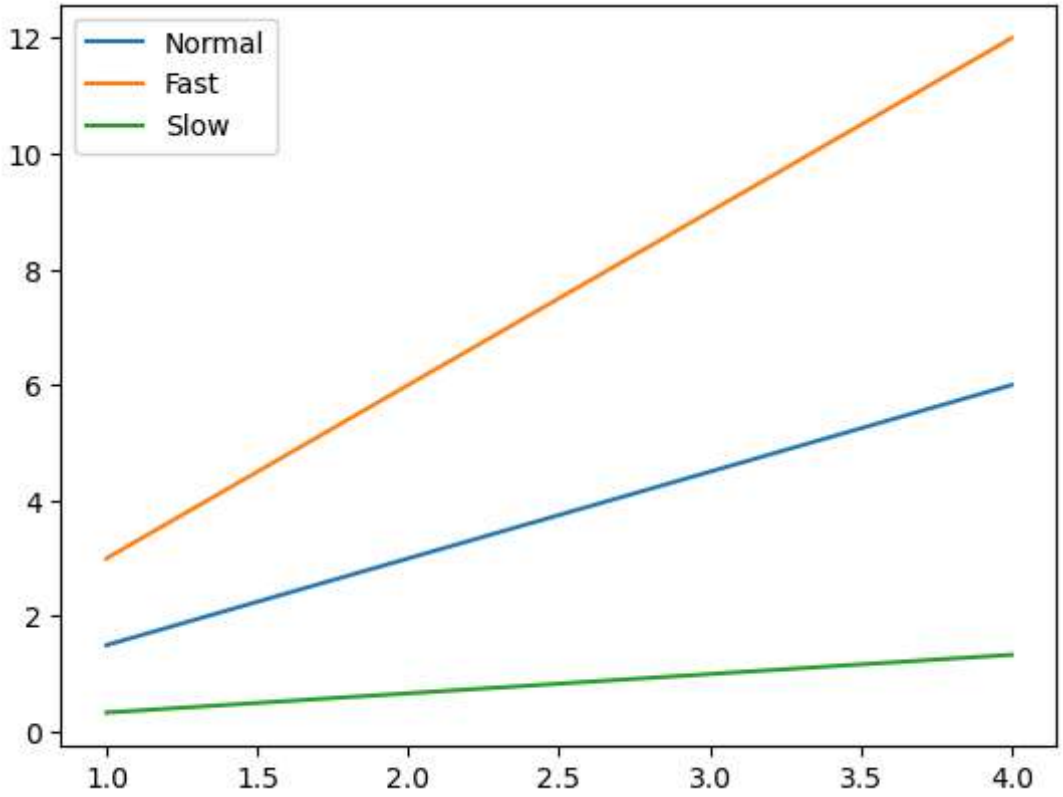


In [43]:
```
x15=np.arange(1,5)
fig,ax=plt.subplots()
ax.plot(x15,x15*1.5)
```

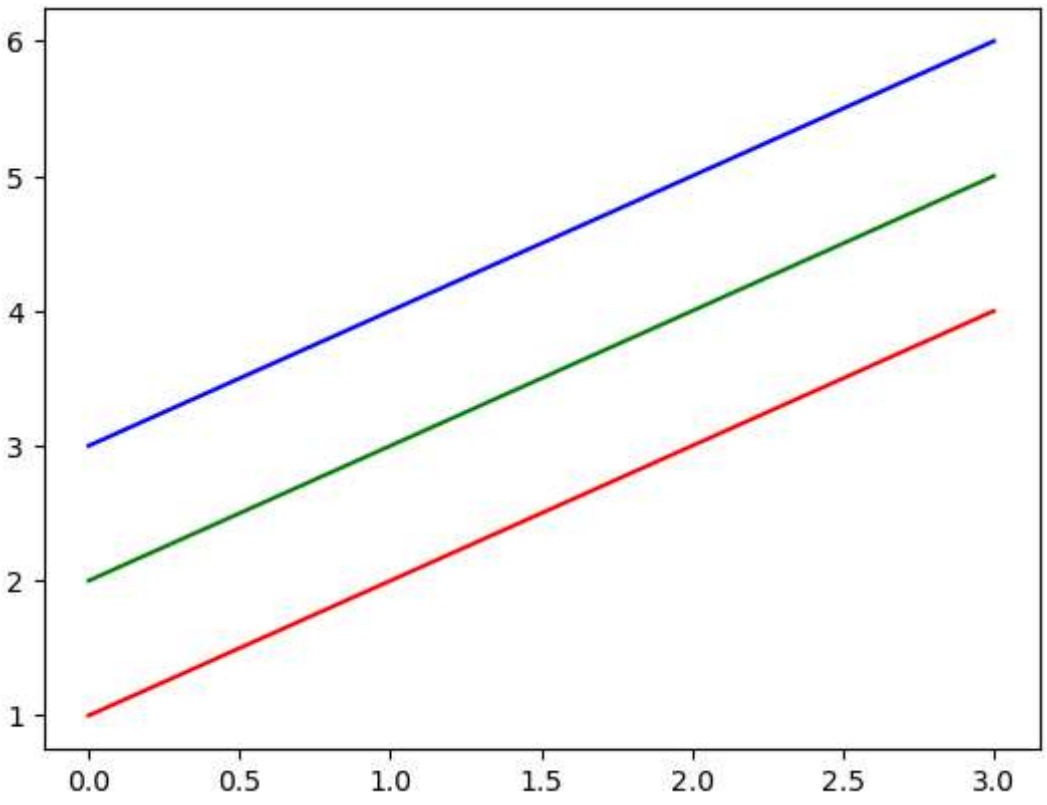
```
ax.plot(x15,x15*3.0)
ax.plot(x15,x15/3.0)
ax.legend(['Normal','Fast','Sloe']);
```



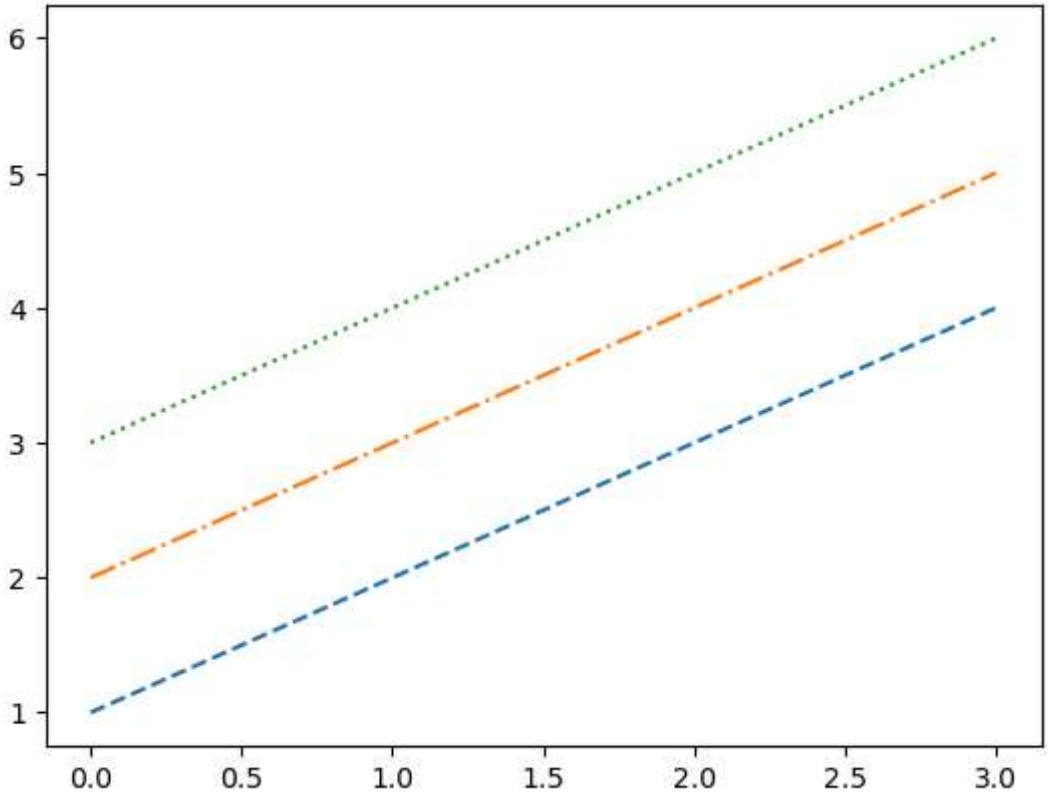
In [44]:
```
x15=np.arange(1,5)
fig,ax=plt.subplots()
ax.plot(x15,x15*1.5,label='Normal')
ax.plot(x15,x15*3.0,label='Fast')
ax.plot(x15,x15/3.0,label='Slow')
ax.legend();
```



In [45]:
```
x16=np.arange (1,5)
plt.plot(x16,'r')
plt.plot(x16+1,'g')
plt.plot(x16+2,'b')
plt.show()
```



In [46]:
```
x16=np.arange(1,5)
plt.plot(x16,'--',x16+1,'-.',x16+2,':')
plt.show()
```



In [ ]: