matplotlib library

What is matplotlib?

matplotlib is a comprehensive library for creating static, animated and interactive visualization

what is data visualization?

Data Visualization is the presentation of data in a pictorial or graphical format

In [1]:
```
import matplotlib.pyplot as plt
```

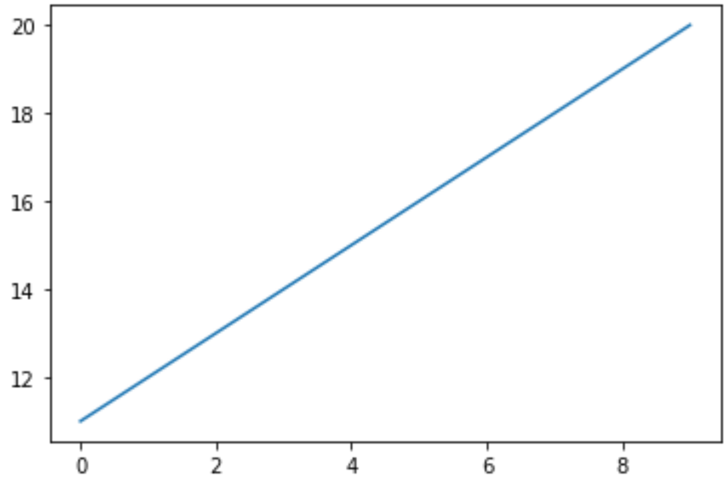
In [2]:
```
from matplotlib import pyplot as pl
```

In [3]:
```
import numpy as np
```

In [4]:
```
x=np.arange(0,10)
y=np.arange(11,21)
```

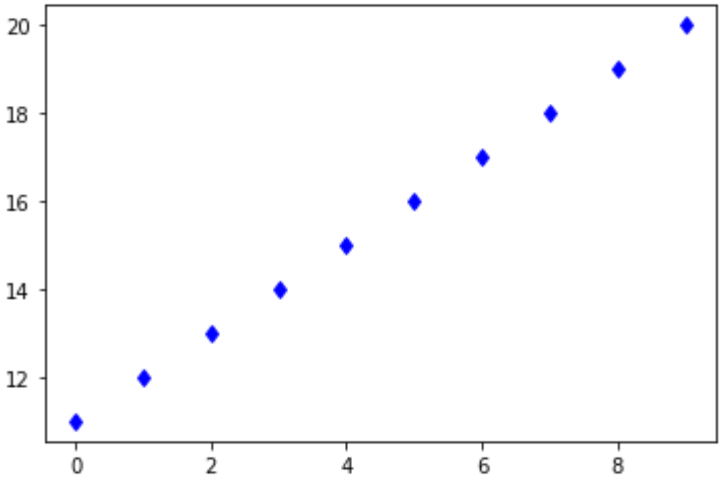
In [5]:
```
plt.plot(x,y)
```

Out[5]:
```
[<matplotlib.lines.Line2D at 0x282d496d610>]
```

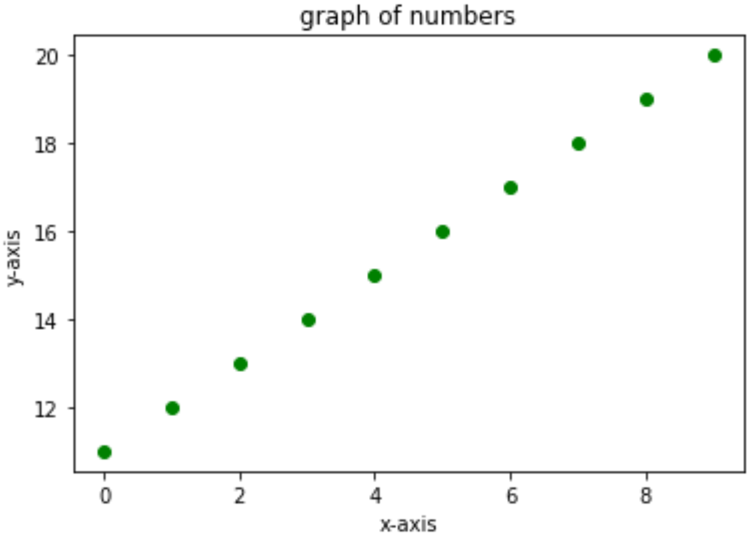


In [7]:
```
plt.plot(x,y,'bd')
```

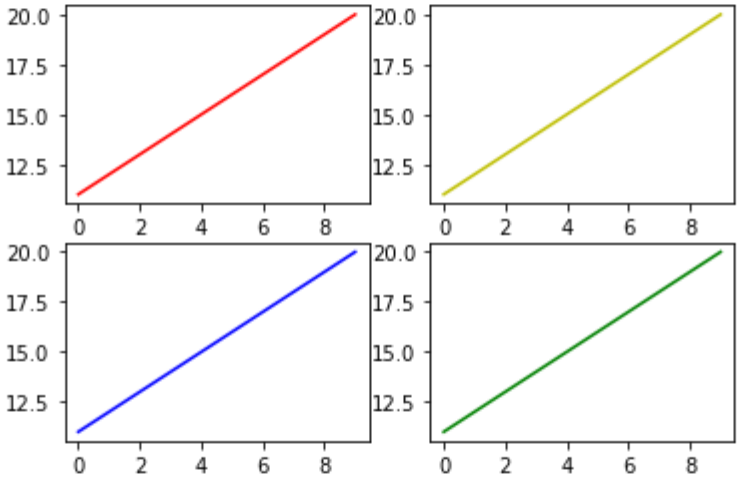
Out[7]:
```
[<matplotlib.lines.Line2D at 0x282d4ac4070>]
```



In [12]:
```
## scatter plot
plt.scatter(x,y,c='g')
plt.xlabel("x-axis")#labeling x axis
plt.ylabel("y-axis")# labeling y axis
plt.title("graph of numbers")#give title of the graph
plt.savefig("test.png")
```



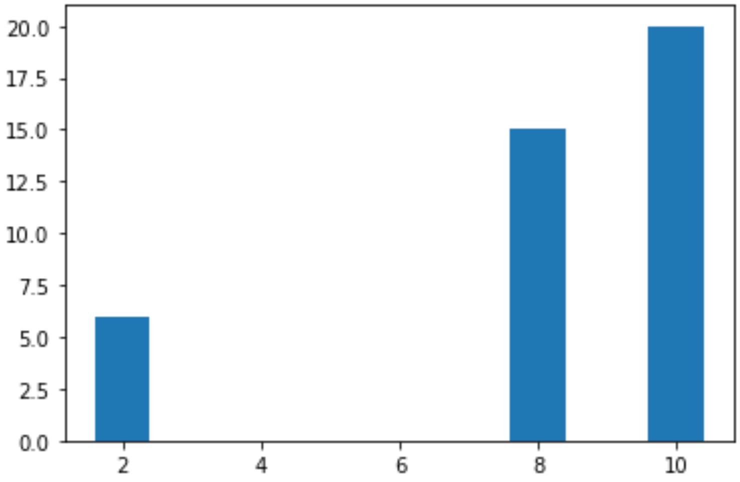
In [17]:
```
plt.subplot(2,2,1)
plt.plot(x,y,'r')
plt.subplot(2,2,2)
plt.plot(x,y,'y')
plt.subplot(2,2,3)
plt.plot(x,y,'b')
plt.subplot(2,2,4)
plt.plot(x,y,'g')
plt.savefig("subplot.png")
```



Bar graph

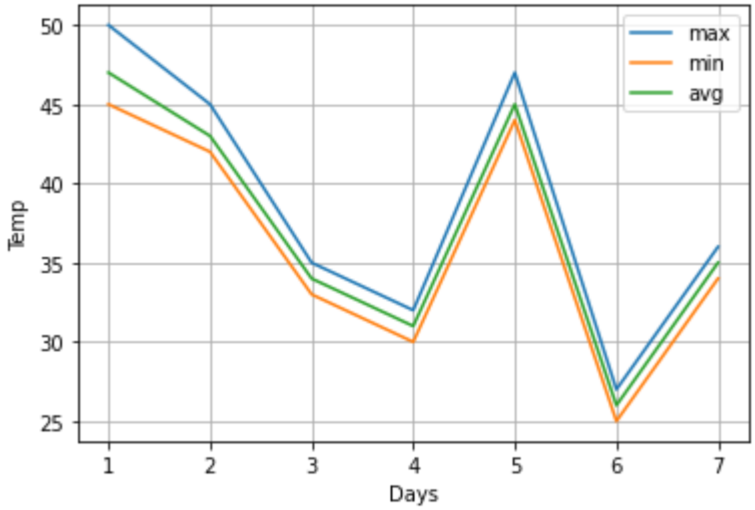
In [19]:
```
x=[2,8,10]
y=[6,15,20]
plt.bar(x,y,width=0.8)
```

Out[19]:
```
<BarContainer object of 3 artists>
```



In [20]:
```
#analyse weekly temp data
day=[1,2,3,4,5,6,7]
max_t=[50,45,35,32,47,27,36]
min_t=[45,42,33,30,44,25,34]
avg_t=[47,43,34,31,45,26,35]
```

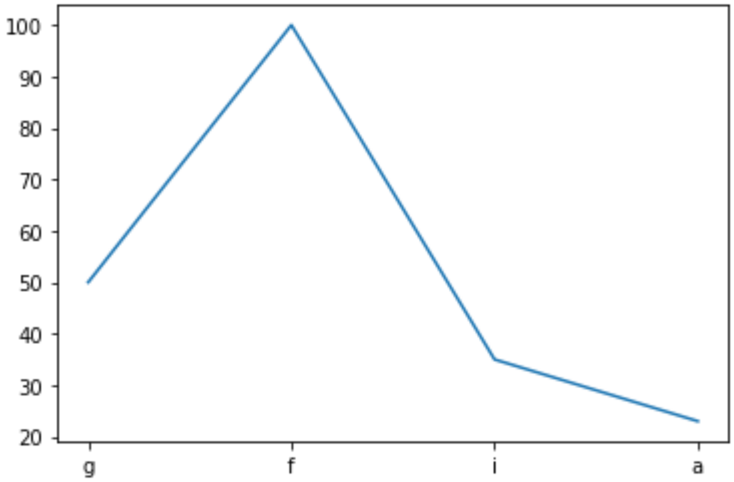
In [21]:
```
plt.plot(day,max_t,label="max")
plt.plot(day,min_t,label="min")
plt.plot(day,avg_t,label="avg")
plt.xlabel("Days")
plt.ylabel("Temp")
plt.legend()
plt.grid()
```



In [22]:
```
company=["g","f","i","a"]
revenue=[50,100,35,23]
```

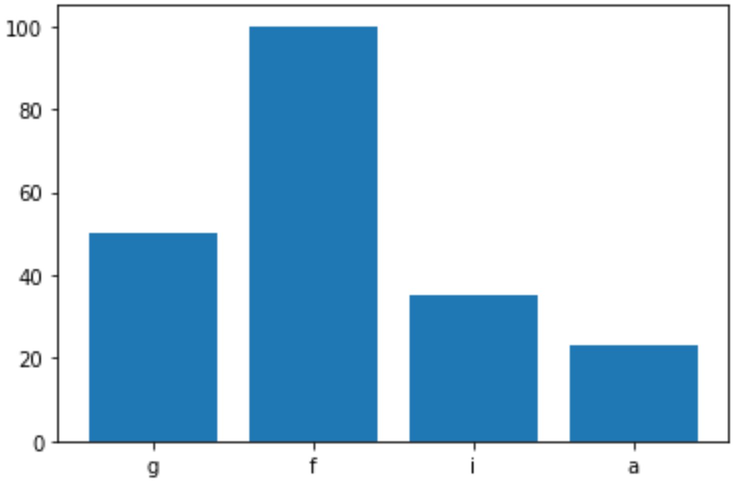
In [23]:
```
plt.plot(company,revenue)
```

Out[23]:
```
[<matplotlib.lines.Line2D at 0x282d626fb80>]
```



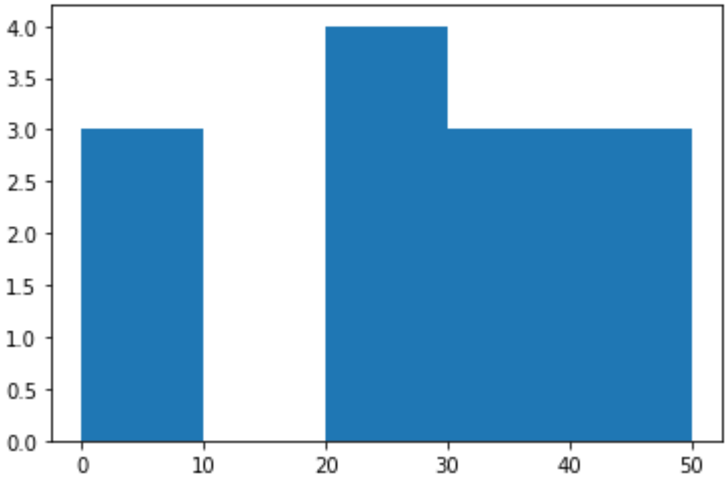
```
In [25]: plt.bar(company.revenue)
```

```
Out[25]: <BarContainer object of 4 artists>
```



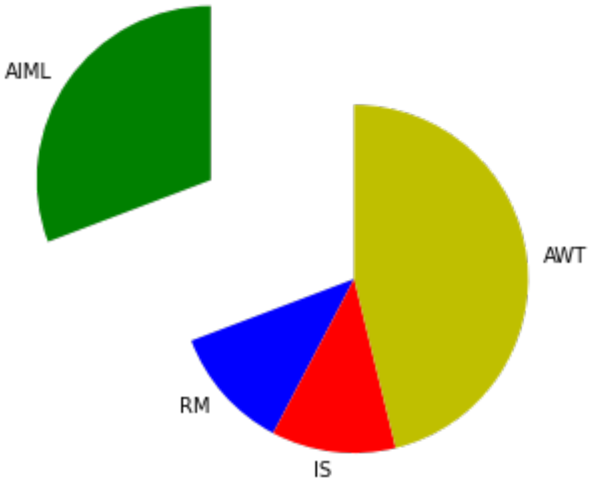
```
In [28]: # histogram
         a=np.array([22,25,23,22,5,6,7,31,32,35,40,45,50])
         bins=[0,10,20,30,40,50]
         plt.hist(a,bins)
```

```
Out[28]: (array([3., 0., 4., 3., 3.]),
          array([ 0, 10, 20, 30, 40, 50]),
          <BarContainer object of 5 artists>)
```



In [30]:
```
# pie chart
slices=[8,3,3,12]
subjects=['AIML','RM','IS','AWT']
cols=['g','b','r','y']
explodes=(1.0,0,0,0)
plt.pie(slices,labels=subjects,colors=cols,startangle=90,explode=explodes)
```

Out[30]:
```
([<matplotlib.patches.Wedge at 0x282d62b4880>,
  <matplotlib.patches.Wedge at 0x282d62b4ca0>,
  <matplotlib.patches.Wedge at 0x282d6d1f190>,
  <matplotlib.patches.Wedge at 0x282d6d1f670>],
 [Text(-1.7282661613346488, 1.1929359059000604, 'AIML'),
  Text(-0.8233617691405228, -0.7294349848463457, 'RM'),
  Text(-0.13259026373746033, -1.0919797717732829, 'IS'),
  Text(1.0919797500487745, 0.13259044265487666, 'AWT')])
```



scipy library

scipy is a python library used to solve scientific and mathematical problems.

In [31]:
```
import numpy as np
from scipy.integrate import quad
```

In [32]:
```
def f(x):
    return 2*x+1
```

In [35]:
```
I,err = quad(f,0,1)
```

In [36]:
```
print(I)
```

```
2.0
```

In [37]:
```
print(err)
```

```
2.220446049250313e-14
```

In [38]:
```
def f(x):
    return np.cos(x)
```

In [39]:
```
xlo=0
xhi=np.pi/2
```

In [40]:
```
I,err=quad(f,xlo,xhi)
```

In [41]:
```
print(I)
```

```
0.9999999999999999
```

In [42]:
```
from scipy import special
```

In [43]:
```
help(special)
```

```
Help on package scipy.special in scipy:

NAME
    scipy.special

DESCRIPTION
    ========================================
    Special functions (:mod:`scipy.special`)
    ========================================

    .. currentmodule:: scipy.special

    Nearly all of the functions below are universal functions and follow
    broadcasting and automatic array-looping rules.

    .. seealso::

       `scipy.special.cython_special` -- Typed Cython versions of special fun
ctions
```

In [44]:
```
a=special.exp10(2)
a
```

Out[44]:
```
100.0
```

In [45]:
```
b=special.exp2(3)
b
```

Out[45]:
```
8.0
```

In [46]:
```
c=special.sindg(90)
c
```

Out[46]:
```
1.0
```

In [47]:
```
d=special.tandg(45)
d
```

Out[47]:
```
1.0
```

```
In [48]: from scipy import linalg
```

```
In [49]: a=np.array([[1,2],[3,4]])
         a
```

```
Out[49]: array([[1, 2],
                [3, 4]])
```

```
In [50]: b=linalg.inv(a)
         b
```

```
Out[50]: array([[-2. ,  1. ],
                [ 1.5, -0.5]])
```

```
In [ ]:
```