In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("/Users/dhanashreesunilyadav/Desktop/Book1.csv")
from sklearn.model_selection import train_test_split
print(df)
print(df.head())
print(df.tail())
```

```
    Hours  Scores
0     2.5      21
1     5.1      47
2     3.2      27
3     8.5      75
4     3.5      30
5     1.5      20
6     9.2      88
7     5.5      60
8     8.3      81
9     2.7      25
10    7.7      85
11    5.9      62
12    4.5      41
13    3.3      42
14    1.1      17
15    8.9      95
16    2.5      30
17    1.9      24
18    6.1      67
19    7.4      69
20    2.7      30
21    4.8      54
22    3.8      35
23    6.9      76
24    7.8      86
   Hours  Scores
0    2.5      21
1    5.1      47
2    3.2      27
3    8.5      75
4    3.5      30
    Hours  Scores
20    2.7      30
21    4.8      54
22    3.8      35
23    6.9      76
24    7.8      86
```

In [2]:

```
print(df.info())
print(df.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
None
(25, 2)
```

In [3]:

```
print(df.groupby("Hours").size())
```

```
Hours
1.1    1
1.5    1
1.9    1
2.5    2
2.7    2
3.2    1
3.3    1
3.5    1
3.8    1
4.5    1
4.8    1
5.1    1
5.5    1
5.9    1
6.1    1
6.9    1
7.4    1
7.7    1
7.8    1
8.3    1
8.5    1
8.9    1
9.2    1
dtype: int64
```

In [4]:

```
print(df.groupby("Scores").size())
```

```
Scores
17    1
20    1
21    1
24    1
25    1
27    1
30    3
35    1
41    1
42    1
47    1
54    1
60    1
62    1
67    1
69    1
75    1
76    1
81    1
85    1
86    1
88    1
95    1
dtype: int64
```

In [5]:

```
print((df.isnull()).sum())
```

```
Hours     0
Scores    0
dtype: int64
```

In [6]:

```
print(df.describe())
```

```
           Hours     Scores
count  25.000000  25.000000
mean    5.012000  51.480000
std     2.525094  25.286887
min     1.100000  17.000000
25%     2.700000  30.000000
50%     4.800000  47.000000
75%     7.400000  75.000000
max     9.200000  95.000000
```

In [7]:

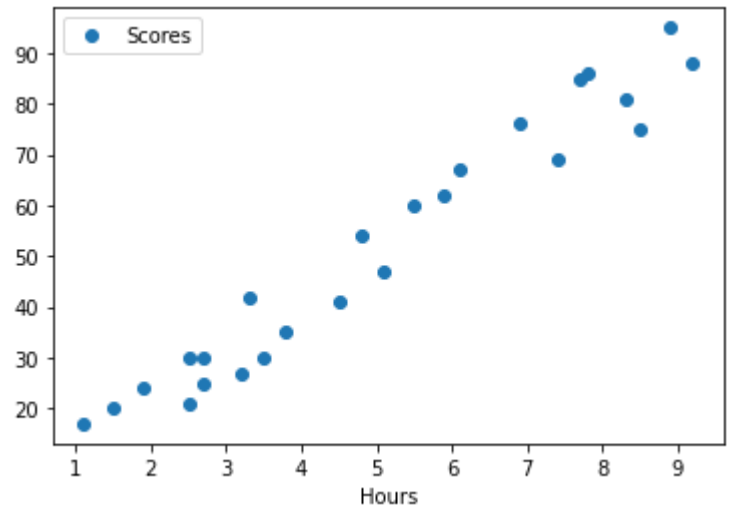
```
print(df.corr())
```

```
           Hours    Scores
Hours   1.000000  0.976191
Scores  0.976191  1.000000
```

In [8]:

```
print(df.plot(x="Hours",y="Scores",style="o"))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



In [9]:

```
x=df.iloc[:,:-1].values
print(x)
```

```
[[2.5]
 [5.1]
 [3.2]
 [8.5]
 [3.5]
 [1.5]
 [9.2]
 [5.5]
 [8.3]
 [2.7]
 [7.7]
 [5.9]
 [4.5]
 [3.3]
 [1.1]
 [8.9]
 [2.5]
 [1.9]
 [6.1]
 [7.4]
 [2.7]
 [4.8]
 [3.8]
 [6.9]
 [7.8]]
```

In [10]:

```
y=df.iloc[:,1].values
print(y)
```

```
[21 47 27 75 30 20 88 60 81 25 85 62 41 42 17 95 30 24 67 69 30 54 35
76
 86]
```

In [11]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

In [12]:

```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)
```

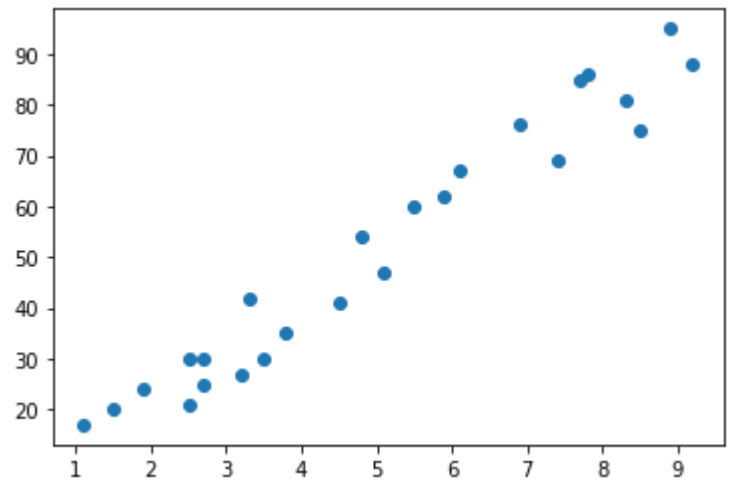
Out[12]:

```
LinearRegression()
```

In [13]:

```
print(plt.scatter(x,y))
```

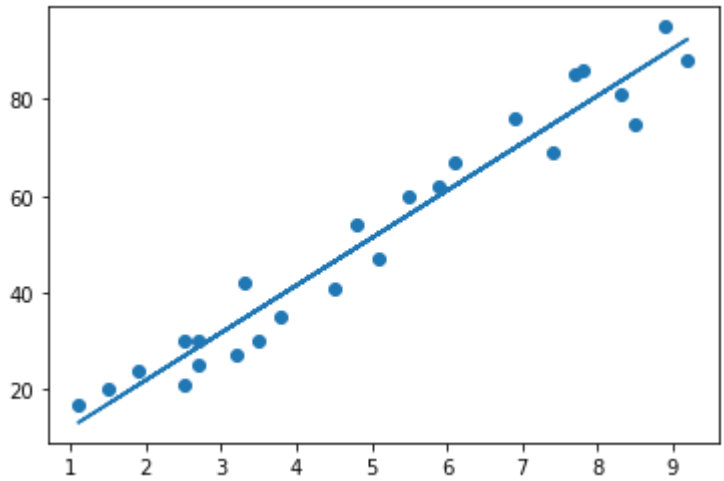
```
<matplotlib.collections.PathCollection object at 0x7f879d94ac70>
```



In [14]:

```
line=regressor.coef_*x+regressor.intercept_

plt.scatter(x,y)
plt.plot(x,line);
```



In [15]:

```
print(x_test)
y_pred=regressor.predict(x_test)
#print(y_pred)
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]
 [3.8]
 [1.9]
 [7.8]]
```

In [16]:

```
dataframe=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
print(dataframe)
```

```
   Actual  Predicted
0      20  17.053665
1      27  33.694229
2      69  74.806209
3      30  26.842232
4      62  60.123359
5      35  39.567369
6      24  20.969092
7      86  78.721636
```

In [17]:

```
study_hour=[[9.25]]
score_prediction=regressor.predict(study_hour)

print('Number of Hours of studying:',study_hour)
print('Predicted Score:',score_prediction)
```

```
Number of Hours of studying: [[9.25]]
Predicted Score: [92.91505723]
```

In [18]:

```
from sklearn.metrics import r2_score
print("R2 Score:", r2_score(y_test,y_pred))
```

```
R2 Score: 0.9568211104435257
```

In [19]:

```
from sklearn.metrics import mean_squared_error
print('Mean Square Error:', mean_squared_error(y_test,y_pred))
```

```
Mean Square Error: 22.965097212700428
```

In [20]:

```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test,y_pred))
```

```
Mean Absolute Error: 4.419727808027651
```

In [21]:

```
#If a student studies 9.25 hrs per day then he would score 92.91
```