Importing neccessary libraries

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

importing data set

In [8]:
```
d=pd.read_csv("student_scores.csv")
d
```

Out[8]:

	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

Analyzing dataset

In [7]:
```
#total no. of rows and columns present in the dataset
d.shape
```

Out[7]:
```
(25, 2)
```

In [10]:
```
#presence of null values
d.isnull().sum()
```

Out[10]:
```
Hours     0
Scores    0
dtype: int64
```

In [11]:
```
#datatypes present at each columns
d.dtypes
```

Out[11]:
```
Hours     float64
Scores      int64
dtype: object
```

In [12]:
```
#correlation between 2 columns
d.corr()
```

Out[12]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

In [13]:
```
#statistical information of given datas
d.describe()
```

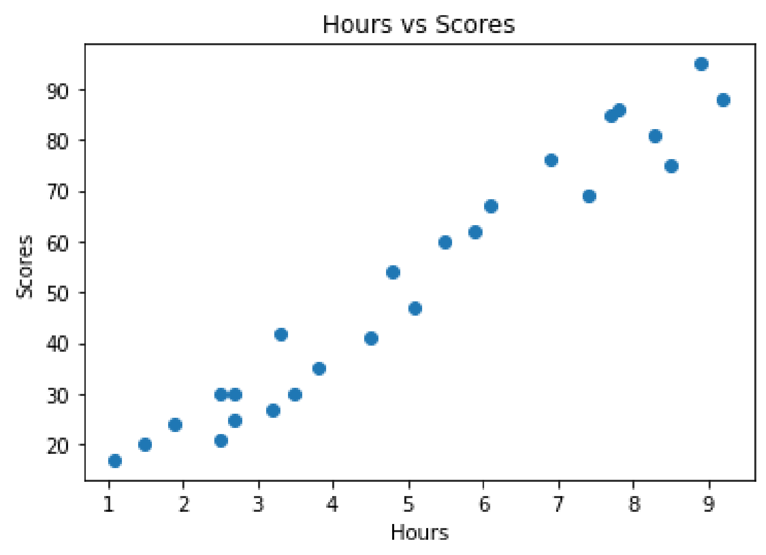
Out[13]:

	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

Data visualization

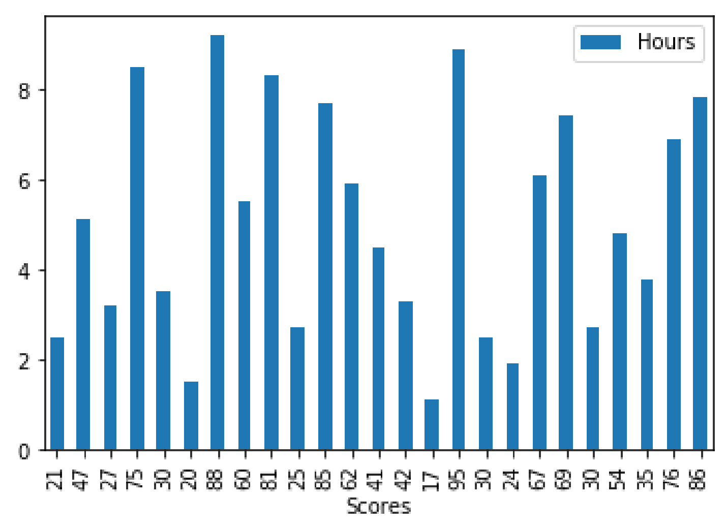
In [15]:
```
#Scatter plot
x=d["Hours"]
y=d["Scores"]
plt.scatter(x,y)
plt.title("Hours vs Scores")
plt.xlabel("Hours")
plt.ylabel("Scores")
```

Out[15]:
```
Text(0, 0.5, 'Scores')
```



In [17]:
```
#barplot
d.plot.bar(x="Scores",y="Hours")
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe0ffb2b510>
```



Model building on dataset

In [25]:
```
x=d[['Hours']]
y=d[['Scores']]
```

In [26]:
```
#splitting the dataset into training & testing data
import sklearn
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [27]:
```
#import linear regression model using training data
from sklearn.linear_model import LinearRegression
t=LinearRegression()
t.fit(x_train,y_train)
```

Out[27]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [36]:
```
#making predictions
y_pred=t.predict(x_test)
y_pred
```

Out[36]:
```
array([[33.5159921 ],
       [84.75597236],
       [26.48305364],
       [52.60539651],
       [34.5206976 ],
       [62.65245146],
       [79.73244488],
       [86.76538335]])
```

What will be predicted score if a student study for 9.25 hrs in a day?

In [37]:
```
#solutions
h=9.25
a=t.predict([[h]])
print("hours of studying:",format(h))
print("Predicted scores:",format(a[0]))
```

```
hours of studying: 9.25
Predicted scores: [94.30067456]
```

In [38]:
```
#model evaluation
from sklearn import metrics
print("MAE:",metrics.mean_absolute_error(y_test,y_pred))
print("MSE:",metrics.mean_squared_error(y_test,y_pred))
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
MAE: 6.156762092793679
MSE: 44.11281230398366
RMSE: 6.641747684456528
```