```python
In [ ]:  #import libraries to be used
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         %matplotlib inline
```

```python
In [2]:  #importing the dataset and exploring it
         url="http://bit.ly/w-data"
         df=pd.read_csv(url)
         df.head(10)
```

Out[2]:

	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

```python
In [3]:  #fitting the dataset
         X=df.iloc[:,:-1].values
         y=df.iloc[:,1].values
```

```python
In [4]:  #splitting the dataset into training and testing sets
         from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

```python
In [5]:  #training the simple linear regression model on the training set
         from sklearn.linear_model import LinearRegression
         regressor=LinearRegression()
         regressor.fit(X_train,y_train)
```

Out[5]: LinearRegression()

```python
In [6]:  #predicting the test set result
         y_pred=regressor.predict(X_test)
         y_pred
```

Out[6]: array([16.88414476, 33.73226078, 75.357018 , 26.79480124, 60.49103328])

```python
In [7]:  #checking the R-squared score of the model
         print('R-squared score (training): {:.3f}'
              .format(regressor.score(X_train, y_train)))
         print('R-squared score (test): {:.3f}'
              .format(regressor.score(X_test, y_test)))
```

```
R-squared score (training): 0.952
R-squared score (test): 0.945
```

```python
In [8]:  #visualising the training set
         plt.scatter(X_train,y_train,color='red')
         plt.plot(X_train,regressor.predict(X_train),color='blue')
         plt.title('Hours vs Scores(train)')
         plt.xlabel('Hours')
         plt.ylabel('Scores')
         plt.show()
```
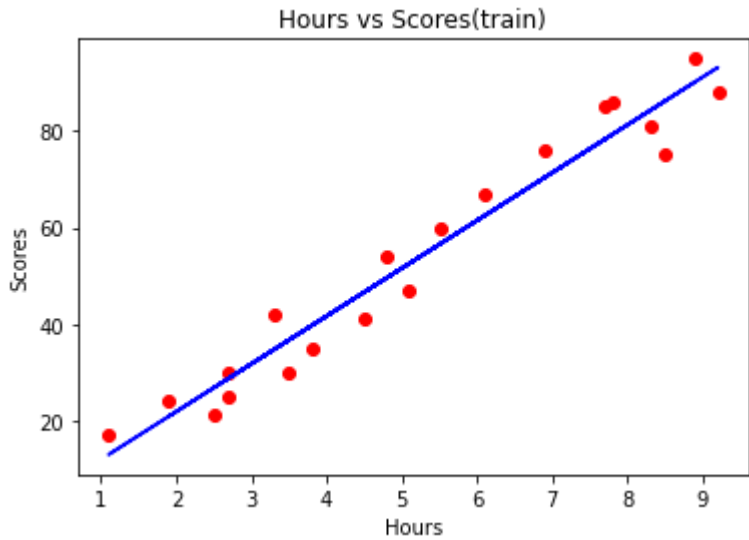


```python
In [9]:  #visualizing the test set
         plt.scatter(X_test,y_test,color='red')
         plt.plot(X_train,regressor.predict(X_train),color='blue')
         plt.title('Hours vs Scores(test)')
         plt.xlabel('Hours')
         plt.ylabel('Scores')
         plt.show()
```
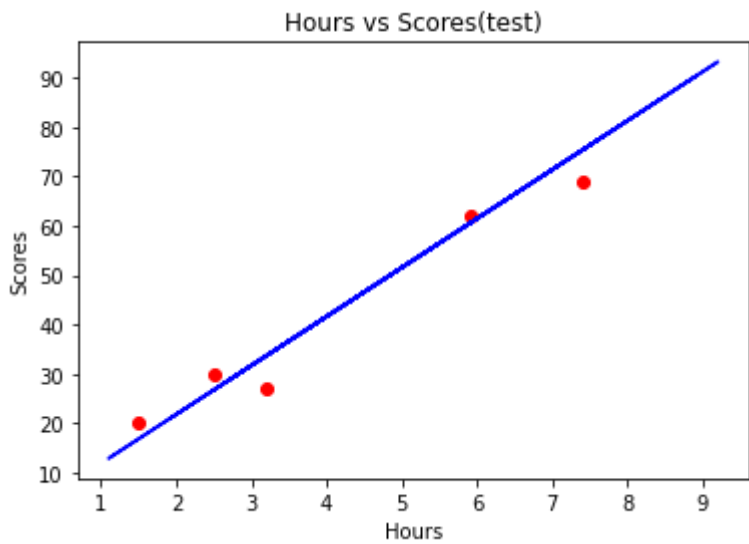


```python
In [10]: #predicting the score when a student puts in 92.5 hours of studying time
         answer=regressor.predict([[9.25]])
         print("when a student puts in 9.25 hours,he is expected to get a score of {} .".format(answer))
```

```
when a student puts in 9.25 hours,he is expected to get a score of [93.69173249] .
```

```python
In [ ]:
```