In [1]:
```
##importing libraries
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')  # to supress warnings
plt.style.use('ggplot')
plt.style.use('fivethirtyeight')# for better visualization
```

In [3]:
```
data=pd.read_csv('http://bit.ly/w-data')
data
```

Out[3]:

| | 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 |

In [5]:
```
data.shape
```

Out[5]:
```
(25, 2)
```

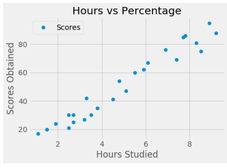
In [6]:
```
data.describe()
```

Out[6]:

| | 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 [29]:
```
data.plot(x='Hours',y='Scores',style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Scores Obtained')
plt.show()
```

Out[29]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e9776ce988>
Text(0.5, 1.0, 'Hours vs Percentage')
Text(0.5, 0, 'Hours Studied')
Text(0, 0.5, 'Scores Obtained')
```




In [11]:
```
X = data.iloc[:,:-1].values
y = data.iloc[:,1].values
```

In [12]:
```
from sklearn.model_selection import train_test_split
```

In [14]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,rando
m_state=0)
```

In [15]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
```

In [16]:
```
reg.fit(X_train,y_train)
```

Out[16]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize
=False)
```

In [ ]:
```

```

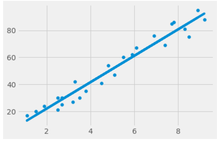
In [17]:
```
print("Training is done")
```

```
Training is done
```

In [18]:
```
l = reg.coef_*X+reg.intercept_

plt.scatter(X,y)
plt.plot(X,l)
plt.show()
```

Out[18]:
```
<matplotlib.collections.PathCollection at 0x1e97720450>
[<matplotlib.lines.Line2D at 0x1e97719edc8>]
```




In [19]:
```
print('Intercept is :')
print(reg.intercept_)
```

```
Intercept is :
2.370815382341881
```

In [21]:
```
print('Coefficient is : ')
print(reg.coef_)
```

```
Coefficient is :
[9.78856669]
```

In [22]:
```
#precitions
y_predict = reg.predict(X_test)
```

In [23]:
```
y_predict
```

Out[23]:
```
array([17.05366541, 33.69422878, 74.80620886, 26.8422321 , 60.12335883,
       39.56736879, 20.96909209, 78.72163554])
```

In [25]:
```
#comparing
df = pd.DataFrame({'Actual':y_test, 'Predicted': y_predict})
df
```

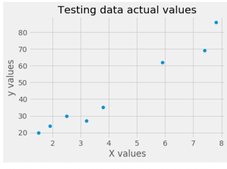
Out[25]:

| | 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 |

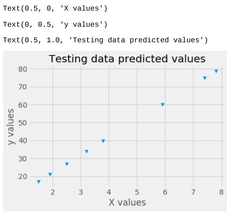
In [27]:
```
#visualize
plt.scatter(X_test,y_test)
plt.xlabel('X values')
plt.ylabel('y values')
plt.title('Testing data actual values')
plt.show()

plt.scatter(X_test,y_predict,marker='v')
plt.xlabel('X values')
plt.ylabel('y values')
plt.title('Testing data predicted values')
plt.show()
```

Out[27]:
```
<matplotlib.collections.PathCollection at 0x1e977258288>
Text(0.5, 0, 'X values')
Text(0, 0.5, 'y values')
Text(0.5, 1.0, 'Testing data actual values')
```




Out[27]:
```
<matplotlib.collections.PathCollection at 0x1e9774cf148>
Text(0.5, 0, 'X values')
Text(0, 0.5, 'y values')
Text(0.5, 1.0, 'Testing data predicted values')
```




In [30]:
```
print('Training score')
print(reg.score(X_train,y_train))
print('Test score')
print(reg.score(X_test,y_test))
```

```
Training score
0.9484097422695115
Test score
0.9568211104435257
```

In [33]:
```
print('No of Hours = ',9.25)
print("predicted score=",reg.predict([[9.25]]))
```

```
No of Hours =  9.25
predicted score= [92.91505723]
```

In [ ]:
```

```