

In []:

```
#import all the libraries required for the given task

import import_ipynb
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [17]:

```
# reading data from the link
df=pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv')
print("Data imported successfully")
df.shape
df.head(20)
```

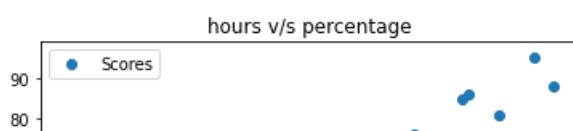
Data imported successfully

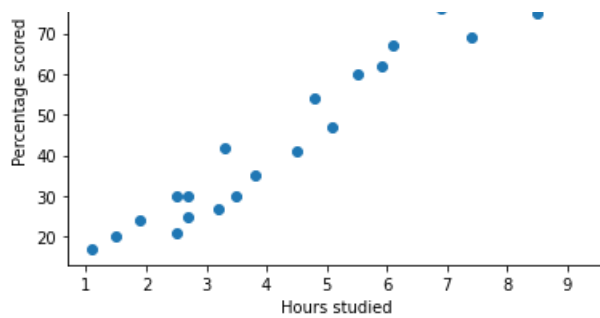
Out[17]:

	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

In [14]:

```
# plotting of graph hours V/S percentage
df.plot(x='Hours',y='Scores',style='o')
plt.title('hours v/s percentage')
plt.xlabel('Hours studied')
plt.ylabel('Percentage scored')
plt.show()
```





In [4]:

```
x = df.iloc[:, :-1].values
y = df.iloc[:, 1].values
```

In [5]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.5 , random_state=0)
```

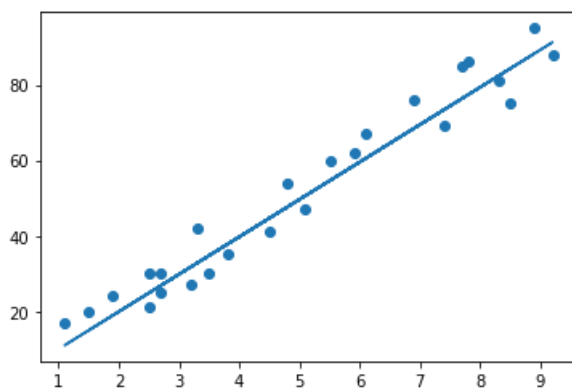
In [6]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
print("split is completed")
```

split is completed

In [7]:

```
#plotting the regression line
line = regressor.coef_*x+regressor.intercept_
#plotting for the test data
plt.scatter(x,y)
plt.plot(x,line);
plt.show()
```



In [8]:

```
print(x_test)# testing data in hours
y_pred = regressor.predict(x_test)# predicting the scores
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]
 [3.8]
 [1.9]
 [7.8]
 [6.9]]
```

```
[1.1]
[5.1]
[7.7]
[3.3]]
```

In [9]:

```
#comparing Actual vs predicted
ds = pd.DataFrame({'actual':y_test,'predicted':y_pred})
ds
```

Out[9]:

	actual	predicted
0	20	14.973540
1	27	31.811747
2	69	73.412023
3	30	24.878368
4	62	58.554782
5	35	37.754644
6	24	18.935471
7	86	77.373954
8	76	68.459609
9	17	11.011609
10	47	50.630920
11	85	76.383471
12	42	32.802230

In [11]:

```
#testing the given data
hours = [9.25]
own_pred = regressor.predict([hours])
print("Number of Hours = {}".format(hours))
print("predicted score = {}".format(own_pred[0]))
```

```
Number of Hours = [9.25]
predicted score = 91.73595402298852
```

In [12]:

```
#finding the mean error
from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(y_test,y_pred))
```

```
Mean Absolute Error: 5.710484526967277
```