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

Gathering the Data :

In [4]:
```
df = pd.read_csv('Salary_Data.csv')
```

In [5]:
```
df.head()
```

Out[5]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [6]:
```
df.tail()
```

Out[6]:

	YearsExperience	Salary
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [7]:
```
df.describe()
```

Out[7]:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
YearsExperience    30 non-null float64
Salary             30 non-null float64
dtypes: float64(2)
memory usage: 560.0 bytes
```

There are 2 colums and 30 rows non-null float values. There are no empty fields. Hence we can proceed to the next step.

Cleaning the data

In [9]:
```
df
```

Out[9]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

the dataset is clean.

Visualization of Data

In [44]:
```
X = df['YearsExperience']
y = df['Salary']
```

In [45]:
```
plt.scatter(X,y);
plt.xlabel('Years Experience')
plt.ylabel('Salary')
plt.title('Scatter plot between Salary and Years Experience');
```

In [46]:
```
plt.plot(X,y);
plt.xlabel('Years Experience')
plt.ylabel('Salary')
plt.title('Graph between Salary and Years Experience');
```

This graph is not a linear regression graph. We cannot predict Salary based on this graph.

Splitting the dataset into test and train dataset

In [24]:
```
from sklearn.model_selection import train_test_split
```

In [93]:
```
X = df.iloc[:, :-1].values
y = df.iloc[:,1].values
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.2, random_state = 0)
```

Importing Linear Regression Model

In [35]:
```
from sklearn.linear_model import LinearRegression
```

In [36]:
```
reg = LinearRegression()
```

In [102]:
```
reg.fit(X_train,y_train)
```

Out[102]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

Training Set Visualization

In [103]:
```
plt.scatter(X_train,y_train,color = 'Red',);
plt.plot(X_train,reg.predict(X_train),color='Blue');
plt.xlabel('Years Experience')
plt.ylabel('Salary')
plt.title('Salary Vs Experience (Train Set)');
```

Predicting Salary for X_Test Dataset

In [104]:
```
y_pred = reg.predict(X_test)
y_pred
```

Out[104]:
```
array([ 40748.96184072, 122699.62295594,  64961.65717022,  63099.14214487,
       115249.56285456, 107799.50275317])
```

Testing Set Visualization

In [105]:
```
plt.scatter(X_test, y_test, color = 'red');
plt.plot(X_train,reg.predict(X_train),color='Blue');
plt.title('Salary vs Experience (Test set)');
plt.xlabel('Years of Experience')
plt.ylabel('Salary');
```

Conclusion

In [106]:
```
x2 = 12
y2 = reg.predict(x2)
y2
```

Out[106]:
```
array([138531.00067138])
```

For 12 years of Experience, the Salary should be 138531.0

According to the dataset, 10.5 Years of Experience got a Salary of 121872.0.

Linear Regression model is completed and we can predict Salary based on the Years of Experience.