In [ ]:
```
#Importing the libraries
```

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [ ]:
```
#Importing the dataset
```

In [2]:
```
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

In [ ]:
```
#Splitting the dataset into the Training set and Test set
```

In [3]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)
```

In [ ]:
```
#Training the Simple Linear Regression model on the Training set
```

In [4]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

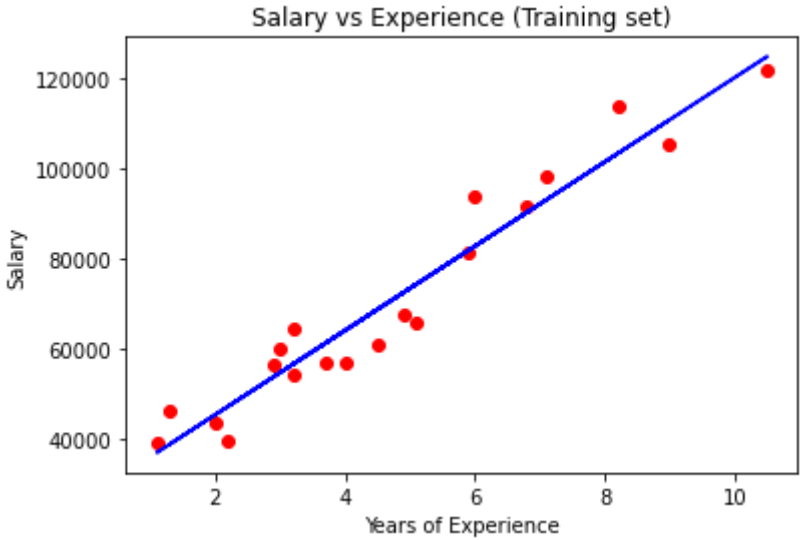
Out[4]:
```
LinearRegression()
```

In [ ]:
```
#Predicting the Test set results
```

In [5]:
```
y_pred = regressor.predict(X_test)
```

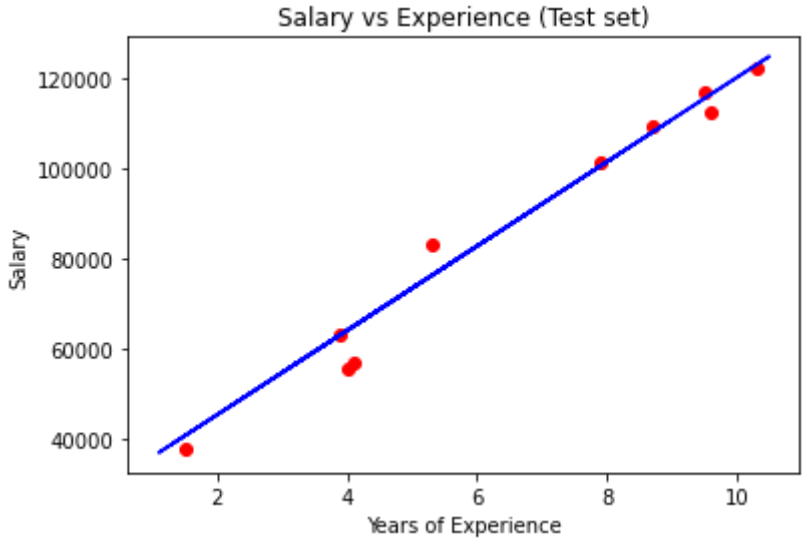
In [ ]:
```
#Visualising the Training set results
```

In [6]:
```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



In [ ]:
```
#Visualising the Test set results
```

In [7]:
```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



In [11]:
```
#Making a single prediction (for example the salary of an employee with 12 years of experience)
print(regressor.predict([[12]]))
```

```
[138967.5015615]
```

In [9]:
```
#Getting the final linear regression equation with the values of the coefficients
print(regressor.coef_)
print(regressor.intercept_)
```

```
[9345.94244312]
26816.192244031176
```

In [ ]:
```
Therefore, the equation of our simple linear regression model is:

Salary=9345.94×YearsExperience+26816.19

Important Note: To get these coefficients we called the "coef_" and "intercept_"
    attributes from our regressor object. Attributes in Python are different
    than methods and usually return a simple value or an array of values.
```