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

In [2]:
```
data=pd.read_excel('Salary (1).xlsx')
print(data)
```

```
    Year of experience  Salary
0                    1      20
1                    2      30
2                    3      40
3                    4      50
4                    5      60
5                    6      70
6                    7      80
7                    8      90
8                    9     100
9                   10     110
10                  11     120
```

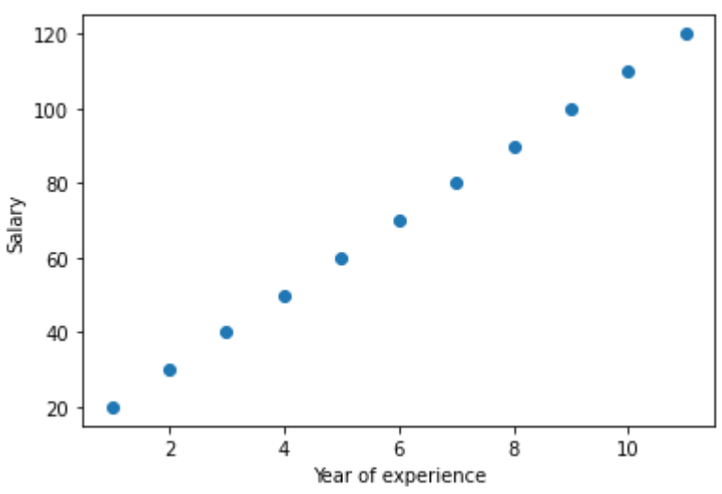
In [3]:
```
data.head()
```

Out[3]:

	Year of experience	Salary
0	1	20
1	2	30
2	3	40
3	4	50
4	5	60

In [4]:
```
independent=data[["Year of experience"]]
dependent=data[["Salary"]]
```

In [6]:
```
plt.scatter(independent,dependent)
plt.xlabel('Year of experience')
plt.ylabel('Salary')
plt.show()
```



In [8]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(independent,dependent,test_size = 1/3, random_state = 0)
```

In [9]:
```
import numpy as np
from sklearn import linear_model
regressor=linear_model.RidgeCV(alphas=np.logspace(-6, 6, 13))
regressor.fit(X_train, Y_train)
```

Out[9]:
```
RidgeCV(alphas=array([1.e-06, 1.e-05, 1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01,
       1.e+02, 1.e+03, 1.e+04, 1.e+05, 1.e+06]))
```

In [10]:
```
weight=regressor.coef_
print("weight of the model={}".format(weight))
bias=regressor.intercept_
print("Bias of the model={}".format(bias))
```

```
weight of the model=[[9.99999966]]
Bias of the model=[10.00000177]
```

In [11]:
```
Y_pred=regressor.predict(X_test)
```

In [12]:
```
from sklearn.metrics import r2_score
r_score=r2_score(Y_test,Y_pred)
print("r_score=",r_score)
```

```
r_score= 0.9999999999999984
```

In [13]:
```
import pickle
filename="finalModel.sav"
pickle.dump(regressor,open(filename, 'wb'))
```

In [14]:
```
load_model=pickle.load(open(filename, 'rb'))
```

In [15]:
```
result=load_model.predict([[30]])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RidgeCV was fitted with feature names
  warnings.warn(
```

In [16]:
```
result
```

Out[16]:
```
array([[309.99999171]])
```