In [178]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [242]:
```
df=pd.read_csv('dhaka homeprices.csv')
```

In [243]:
```
df
```

Out[243]:

	area	price
0	2600	55000
1	3000	56500
2	3200	61000
3	3600	68000
4	4000	72000
5	5000	71000
6	2500	40000
7	2700	38000
8	1200	17000
9	5000	100000

In [244]:
```
df.head()
```

Out[244]:

	area	price
0	2600	55000
1	3000	56500
2	3200	61000
3	3600	68000
4	4000	72000

In [245]:
```
df.tail(3)
```

Out[245]:

	area	price
7	2700	38000
8	1200	17000
9	5000	100000

In [246]:
```
df.shape
```

Out[246]:
```
(10, 2)
```

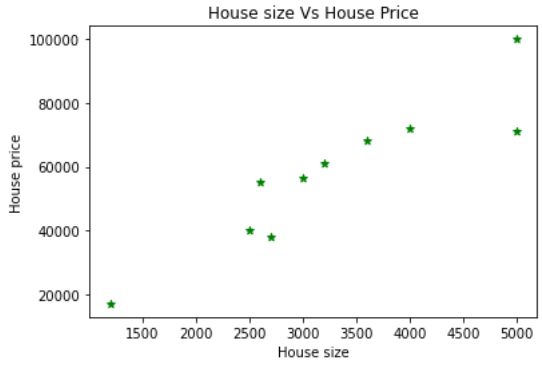
In [247]:
```
df.isnull().any()
```

Out[247]:
```
area     False
price    False
dtype: bool
```

In [248]:
```
df.isnull().sum()
```

Out[248]:
```
area     0
price    0
dtype: int64
```

In [249]:
```
plt.scatter(df['area'], df['price'],marker='*', color='green')
plt.xlabel("House size")
plt.ylabel("House price")
plt.title("House size Vs House Price")
plt.show()
```



In [250]:
```
x=df[["area"]]
y=df['price']
```

In [251]:
```
x
```

Out[251]:

	area
0	2600
1	3000
2	3200
3	3600
4	4000
5	5000
6	2500
7	2700
8	1200
9	5000

In [252]:
```
y
```

Out[252]:
```
0     55000
1     56500
2     61000
3     68000
4     72000
5     71000
6     40000
7     38000
8     17000
9    100000
Name: price, dtype: int64
```

In [253]:
```
from sklearn.model_selection import train_test_split
```

In [254]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.30, random_state=1)
```

In [255]:
```
xtrain
```

Out[255]:

	area
4	4000
0	2600
3	3600
1	3000
7	2700
8	1200
5	5000

In [256]:
```
ytrain
```

Out[256]:
```
4    72000
0    55000
3    68000
1    56500
7    38000
8    17000
5    71000
Name: price, dtype: int64
```

In [257]:
```
xtest
```

Out[257]:

	area
2	3200
9	5000
6	2500

In [258]:
```
ytest
```

Out[258]:
```
2     61000
9    100000
6     40000
Name: price, dtype: int64
```

In [259]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score, explained_variance_score
```

In [260]:
```
reg=LinearRegression()
```

In [261]:
```
reg.fit(xtest, ytest)
```

Out[261]:
```
LinearRegression()
```

In [262]:
```
reg.predict(xtest)
```

Out[262]:
```
array([ 58369.73947896, 100736.47294589,  41893.78757515])
```

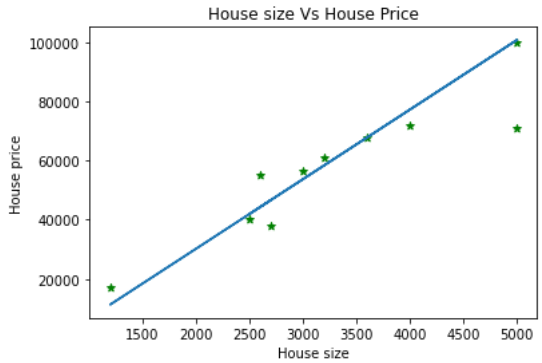
In [263]:
```
reg.predict([[2300]])
```

```
/Users/jonayetmiah/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning: X does not have valid
feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[263]:
```
array([37186.37274549])
```

In [264]:
```
plt.scatter(df['area'], df['price'],marker='*', color='Green')
plt.xlabel("House size")
plt.ylabel("House price")
plt.title("House size Vs House Price")
plt.plot(df.area, reg.predict(df[['area']]))
```

Out[264]:
```
[<matplotlib.lines.Line2D at 0x7f7c3a0e21f0>]
```



In [265]:
```
reg.predict([[2500]])
```

```
/Users/jonayetmiah/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning: X does not have valid
feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[265]:
```
array([41893.78757515])
```

In [266]:
```
reg.coef_
```

Out[266]:
```
array([23.53707415])
```

In [267]:
```
reg.intercept_
```

Out[267]:
```
-16948.897795591198
```

In [268]:
```
y=23.53707415*2500+16948.897795591198
```

In [269]:
```
y
```

Out[269]:
```
75791.58317059118
```

In [270]:
```
reg = LinearRegression().fit(xtrain, ytrain)
```

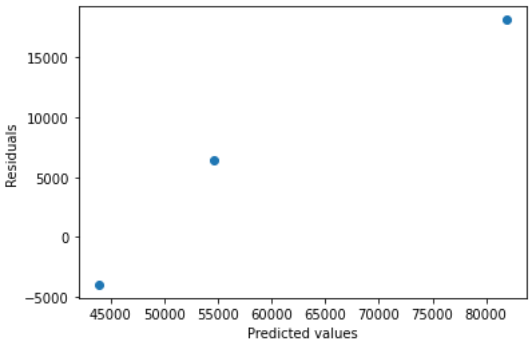
In [271]:
```
ypred = reg.predict(xtest)
```

In [272]:
```
mae = mean_absolute_error(ytest, ypred)
mse = mean_squared_error(ytest, ypred)
rmse = np.sqrt(mse)
r2 = r2_score(ytest, ypred)
evs = explained_variance_score(ytest, ypred)
```

In [273]:
```
print("Mean Absolute Error: ", mae)
print("Mean Squared Error: ", mse)
print("Root Mean Squared Error: ", rmse)
print("R-squared: ", r2)
print("Explained variance score: ", evs)
```

```
Mean Absolute Error:  9513.774558226316
Mean Squared Error:  128787169.7650102
Root Mean Squared Error:  11348.443495255646
R-squared:  0.7916065214158411
Explained variance score:  0.8678925740286283
```

In [274]:
```
residuals = ytest - ypred
plt.scatter(ypred, residuals)
plt.xlabel("Predicted values")
plt.ylabel("Residuals")
plt.show()
```



Author

Jonayet Miah

In [ ]:
```
```