In [1]:
```
## LINEAR REGRESSION DEMO (SimpliLearn)

## importing Numpy, Matplotlib and sklearn libraries
import matplotlib.pyplot as plt
import numpy as np
```

In [4]:
```
##importing datasets from scikit-learn
from sklearn import datasets, linear_model
```

In [5]:
```
##Load the dataset
house_price = [245, 312, 279, 308, 199, 219, 405, 324, 319, 255]
size = [1400, 1600, 1700, 1875, 1100, 1550, 2350, 2450, 1425, 1700]
```

In [7]:
```
# Reshape the input to your regression
size2 = np.array(size).reshape((-1,1))
print(size)
print(size2)
```

```
[1400, 1600, 1700, 1875, 1100, 1550, 2350, 2450, 1425, 1700]
[[1400]
 [1600]
 [1700]
 [1875]
 [1100]
 [1550]
 [2350]
 [2450]
 [1425]
 [1700]]
```

In [8]:
```
##By using fit module in linear regression, user can fit the data frequently a
nd quickly
regr = linear_model.LinearRegression()
regr.fit(size2, house_price)
print("Coefficients: \n", regr.coef_)
print("intercept: \n", regr.intercept_)
```

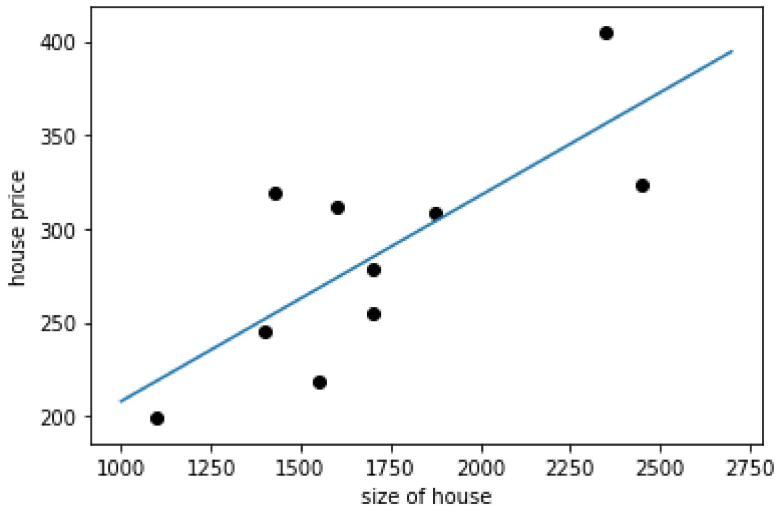
```
Coefficients:
 [0.10976774]
intercept:
 98.24832962138083
```

In [10]:
```
size_new = 1400
price = (size_new * regr.coef_) + regr.intercept_
print(price)
## OR
print(regr.predict([[size_new]])) ## using predict
```

```
[251.92316258]
[251.92316258]
```

In [12]:
```
## Formula obtained for the trained model
def graph(formula, x_range):
    x = np.array(x_range)
    y = eval(formula)
    plt.plot(x,y)
```

In [13]:
```
## Plotting the prediction line
graph('regr.coef_*x + regr.intercept_', range(1000,2700))
plt.scatter (size, house_price, color='black')
plt.ylabel('house price')
plt.xlabel('size of house')
plt.show()
```



In [ ]: