```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('/content/melon.csv')
df
```

	Weight	Price
0	1	50
1	2	100
2	3	140
3	4	180
4	5	220
5	6	250
6	7	300
7	8	340
8	9	380
9	10	390

Next steps: Generate code with `df` View recommended plots

```
df.head()
```

	Weight	Price
0	1	50
1	2	100
2	3	140
3	4	180
4	5	220

Next steps: Generate code with `df` View recommended plots

```
df.tail()
```

	Weight	Price
5	6	250
6	7	300
7	8	340
8	9	380
9	10	390

```
df.isnull().sum()
```

```
Weight    0
Price     0
dtype: int64
```

```
df.dtypes
```

```
Weight    int64
Price     int64
dtype: object
```

```
x=df.iloc[:,:-1]
x
```

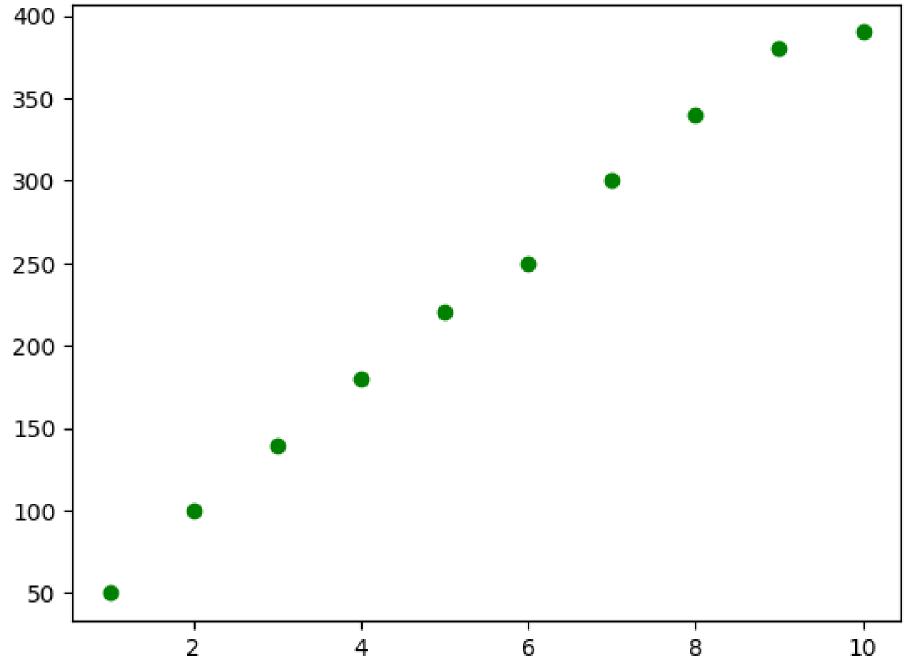
	Weight
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10

```
y=df.iloc[:,-1]
y
```

```
0     50
1    100
2    140
3    180
4    220
5    250
6    300
7    340
8    380
9    390
Name: Price, dtype: int64
```

```
plt.scatter(x,y,color='g')
```

```
<matplotlib.collections.PathCollection at 0x7c585ba2ed10>
```



```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

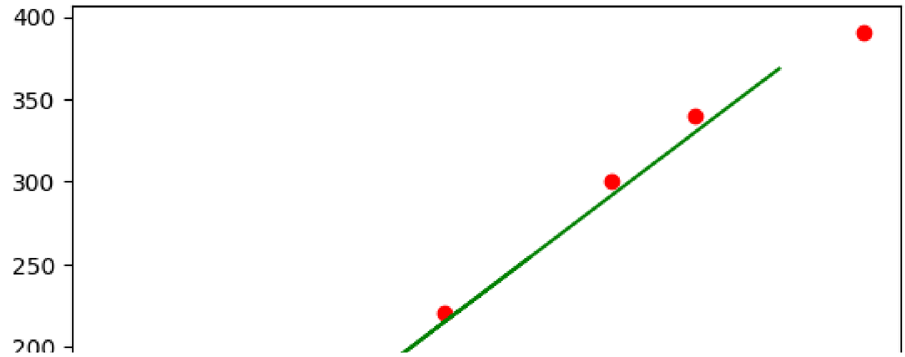
	Weight
0	1
7	8
2	3
9	10
4	5
3	4
6	7

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
```

```
array([368.53960396,  99.8019802 , 253.36633663])
```

```
plt.scatter(x_train,y_train,color='r')
plt.plot(x_test,y_pred,color='g')
```

```
[<matplotlib.lines.Line2D at 0x7c58516f5780>]
```



```
df1=pd.DataFrame({'Actual_value':y_test,'Predicted_value':y_pred,'Differece':y_test-y_pred})
df1
```

	Actual_value	Predicted_value	Differece
8	380	368.539604	11.460396
1	100	99.801980	0.198020
5	250	253.366337	-3.366337

Next steps: Generate code with df1 View recommended plots

```
print("Slope is",model.coef_)
print("Constant is",model.intercept_)
```

```
Slope is [38.39108911]
Constant is 23.01980198019794
```

```
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,mean_squared_error,r2_score
print("MAE is", mean_absolute_error(y_test,y_pred))
print("MAPE is", mean_absolute_percentage_error(y_test,y_pred))
print("MSE is", mean_squared_error(y_test,y_pred))
print("r2_score", r2_score(y_test,y_pred))
data=mean_squared_error(y_test,y_pred)
print("RMSE is",np.sqrt(data))
```

```
MAE is 5.008250825082503
MAPE is 0.015201493833594003
MSE is 47.570703852562964
r2_score 0.9963655659195835
RMSE is 6.8971518652675
```