In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
train_data = pd.read_csv("C:\\Users\\nacha\\OneDrive\\Desktop\\train_data.csv")
test_data = pd.read_csv("C:\\Users\\nacha\\OneDrive\\Desktop\\test_data.csv")
```

In [3]:
```
train_data
```

Out[3]:

	x	y
0	24.0	21.549452
1	50.0	47.464463
2	15.0	17.218656
3	38.0	36.586398
4	87.0	87.288984
...	...	...
695	58.0	58.595006
696	93.0	94.625094
697	82.0	88.603770
698	66.0	63.648685
699	97.0	94.975266

700 rows × 2 columns

In [4]:
```
test_data
```

Out[4]:

	x	y
0	77	79.775152
1	21	23.177279
2	22	25.609262
3	20	17.857388
4	36	41.849864
...	...	...
295	71	68.545888
296	46	47.334876
297	55	54.090637
298	62	63.297171
299	47	52.459467

300 rows × 2 columns

In [5]:
```
print(train_data.isnull().sum())  # Check for missing values in the training dataset
print(test_data.isnull().sum())   # Check for missing values in the test dataset
```

```
x    0
y    1
dtype: int64
x    0
y    0
dtype: int64
```

In [6]:
```
train_data.fillna(train_data.mean(), inplace=True)
test_data.fillna(test_data.mean(), inplace=True)
```

In [7]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

X_train = train_data[['x']]     # Features
y_train = train_data['y']       # Target variable

model = LinearRegression()
model.fit(X_train, y_train)
```

Out[7]:
```
LinearRegression()
```

In [8]:
```
X_test = test_data[['x']]     # Features in the test dataset
y_test = test_data['y']       # True target values in the test dataset

y_pred = model.predict(X_test)   # Predictions on the test dataset

mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

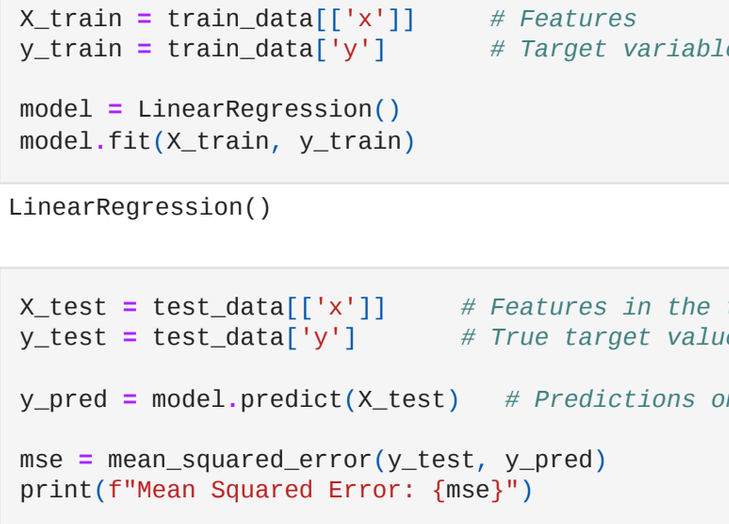
```
Mean Squared Error: 770.6956321737713
```

The Mean Squared Error (MSE) value of 770.70 for your simple linear regression model indicates the average squared difference between the predicted values and the actual values in your dataset.

In summary, an MSE of 770.70 suggests that there is room for improvement in your linear regression model's predictive accuracy.

In [9]:
```
import matplotlib.pyplot as plt

# Assuming y_test and y_pred are your actual and predicted values, respectively
plt.scatter(X_test, y_test, color='green', label='Actual Data')  # Scatter plot of actual data points
plt.plot(X_test, y_pred, color='red', label='Linear Regression')  # Regression line
plt.xlabel('X')  # Replace 'X' with your actual feature name
plt.ylabel('Y')  # Replace 'Y' with your actual target variable name
plt.title('Linear Regression Model')
plt.legend()
plt.show()
```



In [ ]:
```
```