In [6]:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
df = pd.read_csv('Data/PolyData(1).csv')
df

```

Out[6]:

	Unnamed: 0	x	y
0	0	-0.216619	2.113105
1	1	2.945493	10.795517
2	2	-2.818077	4.346195
3	3	-1.641737	3.622927
4	4	0.200467	3.759674
...	...	...	...
195	195	0.057998	2.350656
196	196	-2.936630	6.285578
197	197	2.644792	11.962454
198	198	2.009540	6.082032
199	199	-1.916395	2.883002

200 rows × 3 columns

In [8]:

```
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

```

In [9]:

```
# Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin = LinearRegression()

lin.fit(X, y)

```

Out[9]:

```
LinearRegression()
```

In [10]:

```
# Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree = 2)
X_poly = poly.fit_transform(X)

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)

```

Out[10]:

```
LinearRegression()
```

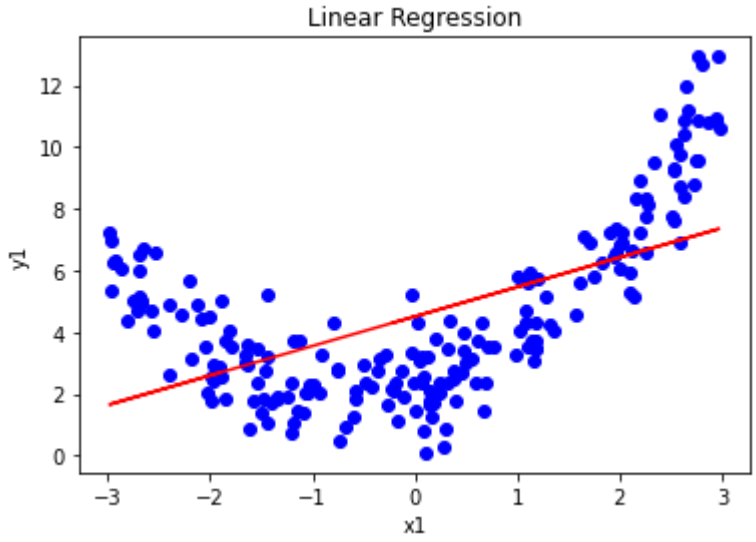
In [11]:

```
# Visualising the Linear Regression results
plt.scatter(X, y, color = 'blue')

plt.plot(X, lin.predict(X), color = 'red')
plt.title('Linear Regression')
plt.xlabel('x1')
plt.ylabel('y1')

plt.show()

```



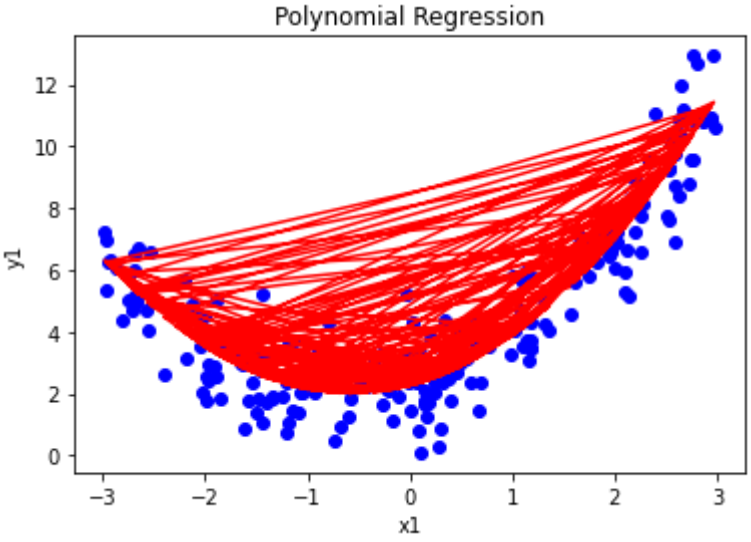
In [12]:

```
# Visualising the Polynomial Regression results
plt.scatter(X, y, color = 'blue')

plt.plot(X, lin2.predict(poly.fit_transform(X)), color = 'red')
plt.title('Polynomial Regression')
plt.xlabel('x1')
plt.ylabel('y1')

plt.show()

```



In [ ]:

```

```