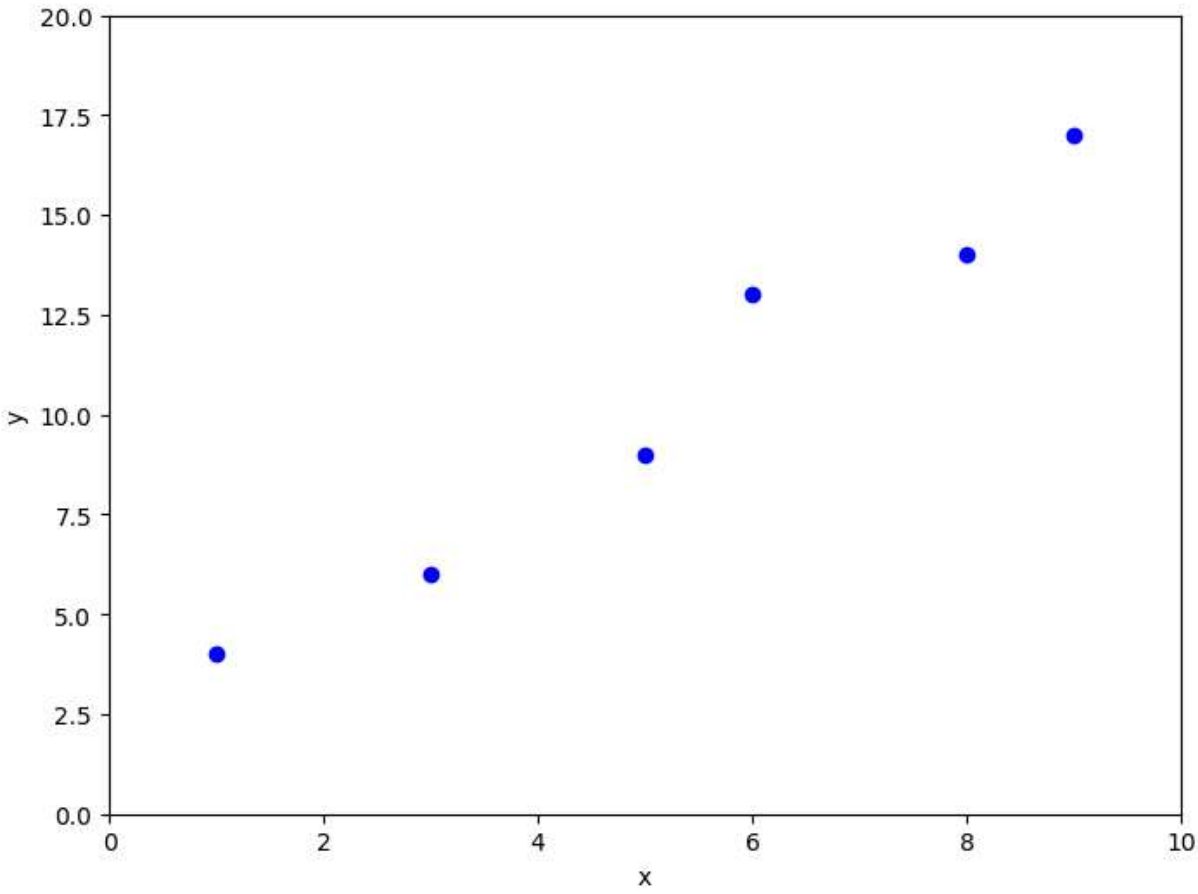
In [9]:
```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, r2_score
```

In [10]:
```
plt.figure(figsize=(8, 6))

X = np.array([1, 3, 5, 6, 8, 9]).reshape((-1, 1))
y = np.array([4, 6, 9, 13, 14, 17]).reshape((-1, 1))
print(X.shape)
print(y.shape)

plt.scatter(X, y, color = "blue", label = "Data Points")
plt.xlim(0, 10)
plt.ylim(0, 20)
plt.xlabel('x')
plt.ylabel('y')

plt.show()
```

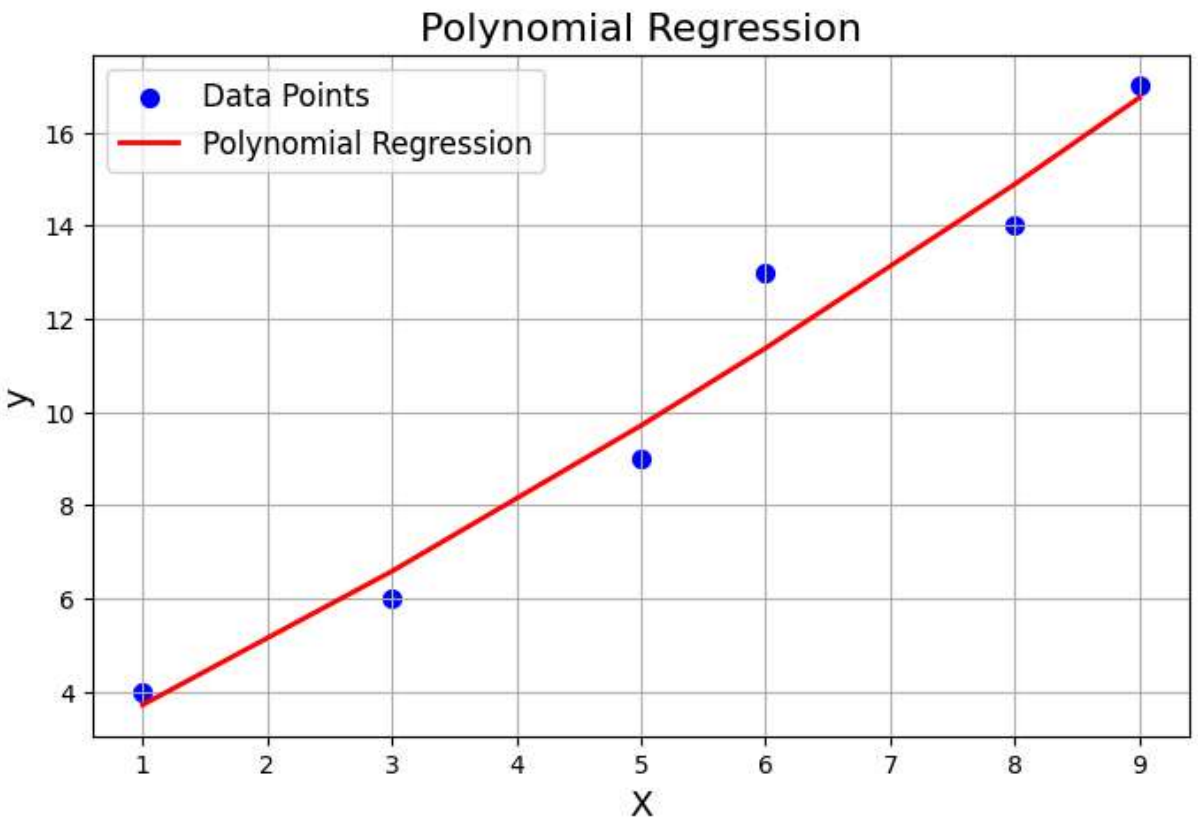
```
(6, 1)
(6, 1)
```



Polynomial Regression

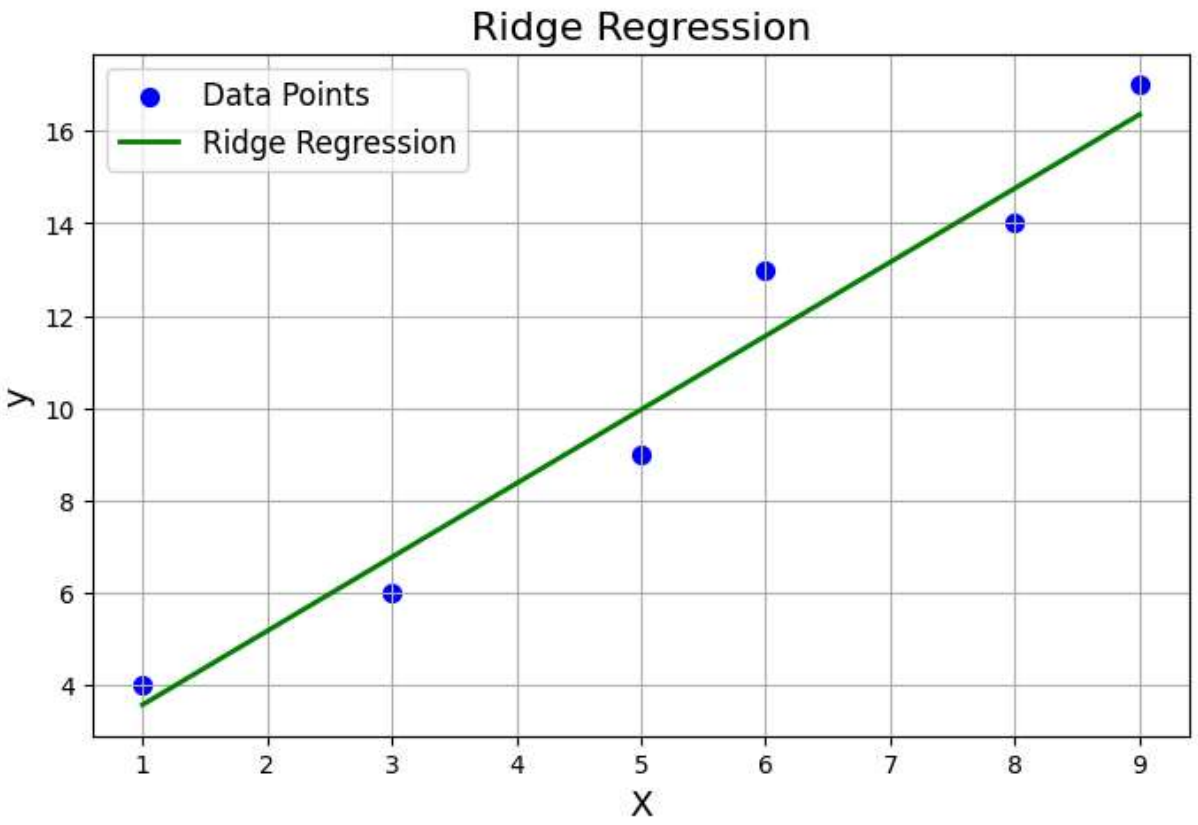
In [28]:
```
# Transform features into polynomial features
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

# Fit a polynomial regression model
model = LinearRegression()
model.fit(X_poly, y)

# Predictions
y_pred = model.predict(X_poly)

# Evaluate
print("MSE:", mean_squared_error(y, y_pred))


# Ridge Regression Plot
plt.figure(figsize=(8, 5))
plt.scatter(X, y, color="blue", label="Data Points", s=50)
plt.plot(X, y_pred, color="red", linewidth=2, label="Polynomial Regression")
plt.xlabel("X", fontsize=14)
plt.ylabel("y", fontsize=14)
plt.title("Polynomial Regression", fontsize=16)
plt.legend(fontsize=12)
plt.grid(True)
plt.show()
```

```
MSE: 0.7387209435474636
```



Ridge Regression

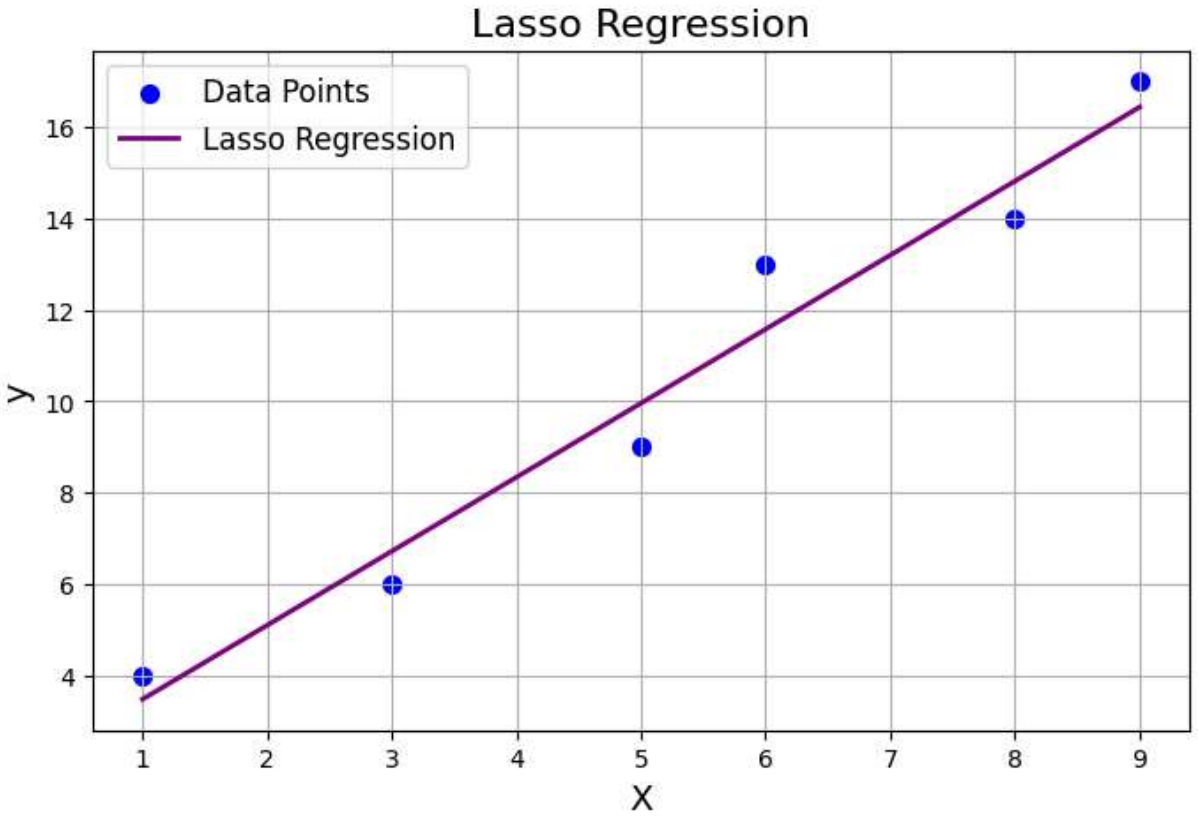
In [29]:
```
# Ridge Regression
ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X, y)
ridge_pred = ridge_model.predict(X)

# Ridge Regression Plot
plt.figure(figsize=(8, 5))
plt.scatter(X, y, color="blue", label="Data Points", s=50)
plt.plot(X, ridge_pred, color="green", linewidth=2, label="Ridge Regression")
plt.xlabel("X", fontsize=14)
plt.ylabel("y", fontsize=14)
plt.title("Ridge Regression", fontsize=16)
plt.legend(fontsize=12)
plt.grid(True)
plt.show()
```



Lasso Regression

In [30]:
```
# Lasso Regression
lasso_model = Lasso(alpha=0.1)
lasso_model.fit(X, y)
lasso_pred = lasso_model.predict(X)

# Lasso Regression Plot
plt.figure(figsize=(8, 5))
```

```
plt.scatter(X, y, color="blue", label="Data Points", s=50)
plt.plot(X, lasso_pred, color="purple", linewidth=2, label="Lasso Regression")
plt.xlabel("X", fontsize=14)
plt.ylabel("y", fontsize=14)
plt.title("Lasso Regression", fontsize=16)
plt.legend(fontsize=12)
plt.grid(True)
plt.show()
```



Model Evaluation

In [31]:
```
# Evaluate Models
print("Polynomial Regression:")
print("MSE:", mean_squared_error(y, poly_pred))
print("R^2:", r2_score(y, poly_pred))

print("\nRidge Regression:")
print("MSE:", mean_squared_error(y, ridge_pred))
print("R^2:", r2_score(y, ridge_pred))

print("\nLasso Regression:")
print("MSE:", mean_squared_error(y, lasso_pred))
print("R^2:", r2_score(y, lasso_pred))
```

```
Polynomial Regression:
MSE: 0.7387209435474636
R^2: 0.9646826640535077

Ridge Regression:
MSE: 0.7936916653727374
R^2: 0.9620545817351679

Lasso Regression:
MSE: 0.7856372549019609
R^2: 0.9624396531520975
```

In [ ]:
```
## Comparison of Polynomial Regression ,
```

In [ ]:
```
# Visualization
plt.figure(figsize=(10, 6))  # Larger figure for better visibility

# Scatter plot for actual data
plt.scatter(X, y, color="blue", label="Data Points", s=50)

# Line plots for each model
plt.plot(X, poly_pred, color="red", linewidth=2, label="Polynomial Regression")
plt.plot(X, ridge_pred, color="green", linestyle="--", linewidth=2, label="Ridge Re
plt.plot(X, lasso_pred, color="purple", linestyle=":", linewidth=2, label="Lasso Re

# Adding labels and legend
plt.xlabel("X", fontsize=14)
plt.ylabel("y", fontsize=14)
plt.title("Comparison of Regression Models", fontsize=16)
plt.legend(fontsize=12)
plt.grid(True)  # Add grid for easier comparison
plt.show()
```

In [ ]:
```

```