

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

# Datos de ejemplo
years = np.array([2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023])
popularity = np.array([30, 35, 40, 45, 50, 55, 60, 70, 75])

# Calcular la regresión lineal
slope, intercept, r_value, p_value, std_err = linregress(years, popularity)

# Imprimir resultados de la regresión
print("Coeficiente de correlación (r):", r_value)
print("Coeficiente de determinación (r^2):", r_value**2)
print("Pendiente (slope):", slope)
print("Intersección (intercept):", intercept)

# Realizar predicciones usando la ecuación de la recta
def predict(x):
    return slope * x + intercept

# Predecir la popularidad para el año 2024
predicted_popularity_2024 = predict(2024)
print(f"Predicción de popularidad para 2024: {predicted_popularity_2024:.2}")

# Generar la línea de tendencia para la gráfica
line = slope * years + intercept

# Crear la gráfica
plt.figure(figsize=(10, 6))
plt.scatter(years, popularity, color='blue', label='Datos Reales')
plt.plot(years, line, color='red', label=f'Tendencia (y = {slope:.2f}x + {intercept:.2f})')
plt.title('Tendencia de Popularidad de Lenguajes de Programación')
plt.xlabel('Año')
plt.ylabel('Popularidad (%)')
plt.legend()
plt.grid(True)
plt.show()

# Evaluar la fuerza de la relación
if r_value > 0.8:
    print("La relación es fuerte y positiva.")
elif r_value < -0.8:
    print("La relación es fuerte y negativa.")
else:
    print("La relación es débil.")

```



