# Polynomial Interpolation (1 POINT)

`scipy` `interpolate` `lagrange`

## Task

In this assignment, you'll use **polynomial interpolation** to generate a polynomial that fits a set of given data points. Your mission is to construct and return the polynomial that perfectly passes through all the points using **Lagrange interpolation** from the `scipy.interpolate` module.

---

## Problem

You're given a set of data points $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$, and your task is to create a function **`interpolate_polynomial`** that returns a polynomial which passes through all these points.

Using **Lagrange interpolation**, the function will construct a polynomial $P(x)$ of degree $n-1$ that exactly fits the provided points.

**Inputs:**

- A list of x-coordinates `x_values`.
- A list of corresponding y-coordinates `y_values`.

**Output:**

- A polynomial function $P(x)$ that can be used to evaluate the polynomial at any point.

## Example

You are given the following points:

$$x = [0, 1, 2], y = [1, 3, 2]$$

Your task is to interpolate a polynomial that passes through these points.

```
>>> x_values = [0, 1, 2]
>>> y_values = [1, 2, 1]
>>> P = interpolate_polynomial(x_values, y_values) # P = 1 + 2x - x^2
>>> P(1)
2.0
>>> P(0)  # Evaluating the polynomial at x = 0
1.0
>>> P(10)
-79.0
```

In this example, the function **`interpolate_polynomial`** returns a polynomial function $P(x)$. Evaluating $P(x)$ at different points yields the corresponding polynomial values.