**Functional programming** uses functions and their evaluations to create programs. Contrarily, in **Object-Oriented programming (OOP)**, objects and methods are used to represent everything

Below are a few main differences between the two

**Functional programming**

The declarative programming model is followed, which means the control flow is not described with the logic of computations. The code does not specify the flow of the steps to be completed.

**Object-Oriented programming (OOP)**

The imperative programming model is followed, which means a control flow is defined. The code itself specifies how the problem is to be solved.

**Functional programming**

The nature of data is immutable, which means that once it is defined, the state of the data can’t be changed.

**Object-Oriented programming (OOP)**

The nature of data is mutable, which means that the state of data can be changed after definition.

**Functional programming**

Recursion is used for iteration. To iterate through anything n number of times, we use a higher-order function to call the main function n number of times.

**Object-Oriented programming (OOP)**

Control flow constructs such as while or for loops are used for iteration. A state variable is updated based on its value and specified rule for iteration.