First-Class VS High-Orde Functions

In JavaScript, the concepts of "first-class functions" and "higher-order functions" are closely related but distinct:

1. First-Class Functions:

- **Definition:**

  This refers to the principle that functions in JavaScript are treated as "first-class citizens," meaning they can be treated like any other data type (e.g., numbers, strings, objects).

- **Implications:**
- Functions can be assigned to variables.
- Functions can be passed as arguments to other functions.
- Functions can be returned as values from other functions.
- Functions can be stored in data structures (arrays, objects).
- Functions can have properties and methods.

2. Higher-Order Functions:

- **Definition:**

  A higher-order function is a function that either:

- Takes one or more functions as arguments.
- Returns a function as its result.
- **Relationship to First-Class Functions:**
  The existence of higher-order functions in JavaScript is a direct consequence of functions being first-class citizens. Without the ability to pass functions as arguments or return them, higher-order functions would not be possible.

In summary:

- **First-class functions**

  describe a fundamental property of the JavaScript language regarding how it treats functions.

- **Higher-order functions**
  are a specific type of function that leverages this first-class nature to enable powerful functional programming paradigms like callbacks, closures, and techniques like `map`, `filter`, and `reduce`.