Jaqueline Zagaste

CIS233DA

June 22, 2023

Arrays and their Methods

An array can be described as a collection of digits, images, or objects organized in rows and columns based on their types. Understanding the concept of arrays helps simplify your work by representing multiple data elements of the same type with a single name. In computer programming, an array assembles equivalent data elements stored at adjacent memory locations. This data structure helps to determine the location of each piece of data by simply using an offset to the base value. An Offset is a number that illustrates the distinction between two index numbers. There are three different types of arrays. There are indexed arrays, multidimensional arrays, and associative arrays. To create indexed arrays, you store a series of values and can access them based on their positions in the array. This is achieved by using indexes, with the first index always being 0 and subsequent indexes incrementing by one of each additional element in the array. Multidimensional arrays are composed of other arrays, forming a nested array structure. These nested arrays, often referred to as multidimensional arrays, bear resemblance to matrices or grids. Multidimensional arrays are commonly employed in programming to represent complex structures. For instance, a chessboard can be visualized as an array consisting of eight elements, with each element itself being an array comprising of eight elements. This allows for an effective model to simulate the grid like structure of a chessboard. An associative array, like an object, consists of unordered keys and corresponding values. Unlike regular arrays that use numeric indexes, associative arrays use keys to organize the stored values. Each key is a unique string and is used to access a specific value. This value can be of any data type, such as Number, Array, Object, and more. When you write code to retrieve a value associated with a key, you are essentially performing an index or lookup operation, which is the primary purpose of using associative arrays. By using associative arrays, you can access the desired array element using a string instead of a number, which is often easier to remember. However, one drawback is that these arrays are not as convenient to use in loops since they don’t rely on numeric indexes. Nevertheless, associative arrays are valuable when frequent lookups based on key values are required. For instance, if you have an array containing names and ages that you need to refer to frequently, an associative array would be a suitable choice. In object-oriented programming (OOP), a method is a procedure defined within a class that can be called by objects instantiated from that class. Each object can independently invoke the method which operates within the context of the calling object. This allows for method reuse across multiple objects created from the same class, ensuring data integrity within each object. Methods are encapsulated within classes and can be invoked separately, performing their designated tasks. Although the specific syntax and usage of methods may vary among OOP languages, the underlying principles remain consistent. Once a class and its methods are defined, an application can reference the class, import it if necessary, and create objects that can access and utilize the class's methods.

Here is a list of JavaScript methods I found interesting:

1. push(): Adds one or more elements to the end of an array.
2. shift(): Removes and returns the first element of an array, shifting all other elements down.
3. unshift(): Adds one or more elements to the beginning of an array, shifting existing elements up.
4. forEach(): Executes a provided function once for each element in the array.
5. every(): Tests whether all elements in the array pass a provided condition.
6. some(): Tests whether at least one element in the array passes a provided condition.
7. isArray(): Checks if a value is an array, returning true or false.