Nicolas Gonzalez

June 20, 2023

CIS233DA 17130

Professor McHardy

Arrays and their Methods

1. An array is an object that allows the storing of data under one single variable name. Usually, an array will start from index 0 and move upwards depending on the size of it. They are used in web pages to make loops and coding simpler and more compact. There are two different types of arrays, multi-dimensional and single-dimensional.

A single-dimensional array is sequential and contains elements of a single type. A multi-dimensional array is an array of arrays which allows for more versatility in its usage. Suppose you have 5 IDs for different students, instead of making 5 different variables, you can just use an array instead. It also makes loops easier to use as you can use a control variable to access the data stored in the array.

1. A method is a block of code that is under a name that can be called at any point in a program. Unless the method is void, there is always a value that is returned at the end of the method. While similar to a function, a method is usually associated with an object.
2. **push**: Adds one or more elements to the end of an array.

**pop**: Removes the last element from an array and returns it.

**shift**: Removes the first element from an array and returns it, shifting all other elements down by one.

**unshift**: Adds one or more elements to the beginning of an array, shifting existing elements up by one.

**concat:** Combines two or more arrays into a new array.

**slice:** Extracts a portion of an array and returns a new array with the selected elements.

**splice:** Changes the contents of an array by removing, replacing, or adding elements at a specific position.

**join:** Joins all elements of an array into a string.

**indexOf:** Returns the first index at which a given element can be found in an array, or -1 if it is not present.

**forEach:** Executes a provided function once for each array element.

**map:** Creates a new array with the results of calling a provided function on every element in the original array.

**filter:** Creates a new array with all elements that pass a given test implemented by a provided function.

**reduce:** Applies a function against an accumulator and each element in the array to reduce it to a single value.

**sort:** Sorts the elements of an array in place and returns the sorted array.

**reverse:** Reverses the order of the elements in an array.

**includes:** Determines whether an array includes a certain element, returning true or false.

**every:** Checks if all elements in an array pass a test implemented by a provided function.

**some:** Checks if at least one element in an array passes a test implemented by a provided function.

**find:** Returns the first element in an array that satisfies a provided testing function.

**findIndex:** Returns the index of the first element in an array that satisfies a provided testing function.