**Arrays and Nested Arrays**

**Arrays**

Arrays are list-like objects

▪ Arrays are a reference type, the variable points to an address in memory Elements are numbered from 0 to length - 1

▪ Creating an array using an array literal

let numbers = [10, 20, 30, 40, 50];

▪Neither the length of a JavaScript array or the types of its elements are fixed

▪ An array's length can be changed at any time

▪ Data can be stored at non-contiguous locations in the array

▪ JavaScript arrays are not guaranteed to be dense



**Accessing Elements**

▪ Array elements are accessed using their index

▪ Accessing indexes that do not exist in the array returns undefined

▪ Arrays can be iterated using for-of loop

**Arrays Indexation**

▪ Setting values via non-integers using bracket notation (or dot notation) creates object properties instead of array elements (will be discussed in later lesson).



**Destructuring Syntax**

▪Expression that unpacks values from arrays or objects, into distinct variables



▪ The rest operator can also be used to collect function parameters into an array

**Mutator Methods**

1. **Pop**

▪ Removes the last element from an array and returns that element

▪ This method changes the length of the array

1. **Push**

▪ The push() method adds one or more elements to the end of an array and returns the new length of the array

1. **Shift**

The shift() method removes the first element from an array and returns that removed element

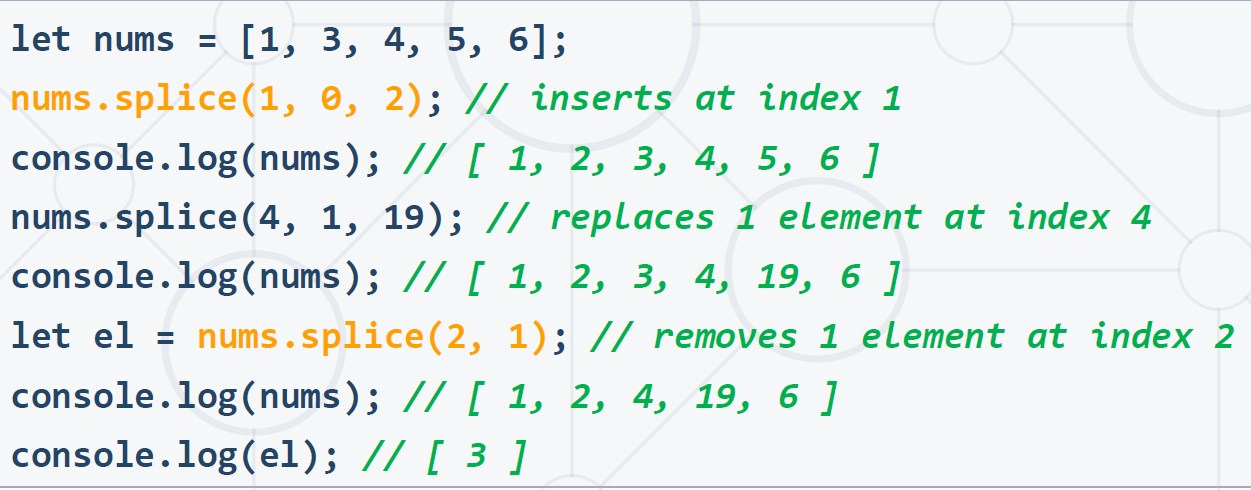
▪ This method changes the length of the array

1. **Unshift**

The unshift() method adds one or more elements to the beginning of an array and returns the new length of the array

1. **Splice**

Changes the contents of an array by removing or replacing existing elements and/or adding new elements



1. **Fill**

Fills all the elements of an array from a start index to an end index with a static value

A screenshot of a computer code

AI-generated content may be incorrect.

1. **Reverse**

▪ Reverses the array

▪ The first array element becomes the last, and the last array element becomes the first

A close up of numbers

AI-generated content may be incorrect.

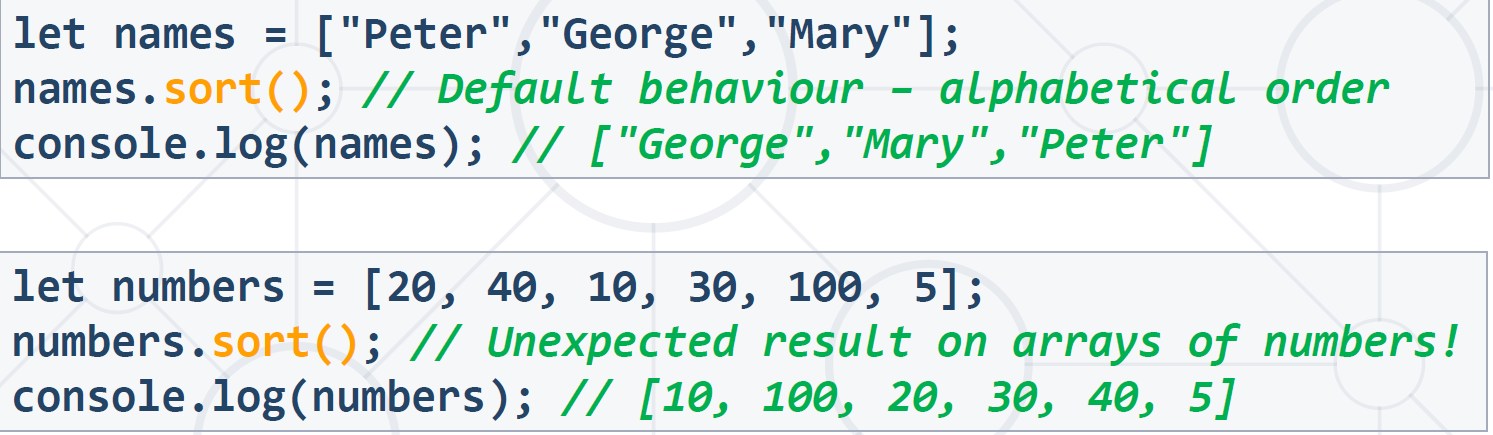
**Sorting arrays**

▪ The sort() method sorts the items of an array

▪ Depending on the provided compare function, sorting can be alphabetic or numeric, and either ascending (up) or descending (down)

▪ By default, the sort() function sorts the values as strings in alphabetical and ascending order

▪ If you want to sort numbers or other values, you need to provide the correct compare function!



**Compare functions**

▪ A function receiving two parameters, e.g. a and b

▪ Returns either a positive number, a negative number, or zero

▪ If result < 0, a is sorted before b

▪ If result > 0, a is sorted after b

▪ If result = 0, a and b are equal (no change)



▪ The localeCompare() method is used to compare any two characters without regard for the case used

▪ It's a string method so it can't be used directly on an array

▪ Pass localeCompare() as the comparison function

A close up of a logo

AI-generated content may be incorrect.

**Accessor Methods**

1. **Join**

▪ Creates and returns a new string by concatenating all of the elements in an array (or an array-like object), separated by commas or a specified separator string

A screenshot of a computer code

AI-generated content may be incorrect.

1. **Concat**

▪ The concat() method is used to merge two or more arrays

▪ This method does not change the existing arrays, but instead returns a new array

A white background with text and numbers

AI-generated content may be incorrect.

1. **Slice**

▪ The slice() method returns a shallow copy of a portion of an array into a new array object selected from begin to end (end not included)

▪ The original array will not be modified

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Includes**

▪ Determines whether an array contains a certain element, returning true or false as appropriate

A white paper with green and orange text

AI-generated content may be incorrect.

1. **IndexOf**

▪ The indexOf() method returns the first index at which a given element can be found in the array

▪ Output is -1 if element is not present

A screenshot of a computer code

AI-generated content may be incorrect.

**Iteration Methods**

1. **ForEach**

▪ The forEach() method executes a provided function once for each array element

▪ Converting a for loop to forEach:



1. **Map**

▪ Creates a new array with the results of calling a provided function on every element in the calling array.

A screenshot of a computer code

AI-generated content may be incorrect.

1. **Some**

▪ The some() method tests whether at least one element in the array passes the test implemented by the provided function

▪ It returns a Boolean value

A screen shot of a computer

AI-generated content may be incorrect.

1. **Find**

▪ Returns the first found value in the array, if an element in the array satisfies the provided testing function or undefined if not found.

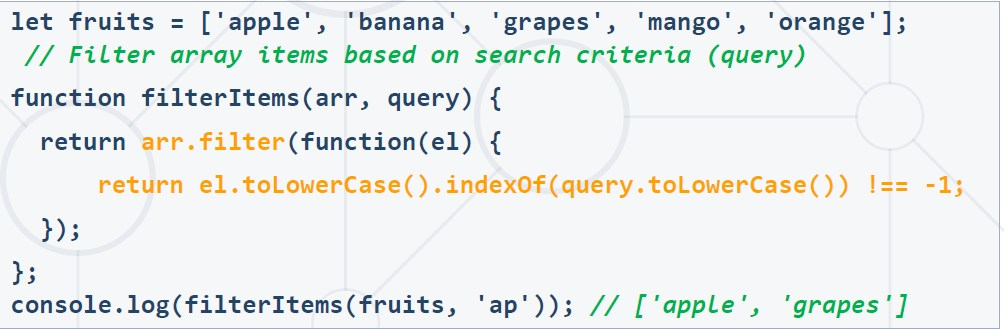


1. **Filter**

▪ Creates a new array with filtered elements only

▪ Calls a provided callback function once for each element in an array

▪ Does not mutate the array on which it is called



1. **Reduce**

▪ The reduce() method executes a reducer function on each element of the array, resulting in a single output value

A screenshot of a computer

AI-generated content may be incorrect.

**Reducer function**

▪ The reducer function takes four arguments:

▪ Accumulator

▪ Current Value

▪ Current Index (Optional)

▪ Source Array (Optional)

▪ Your reducer function's returned value is assigned to the accumulator

▪ Accumulator's value - the final, single resulting value

A screenshot of a computer code

AI-generated content may be incorrect.