JS - Array Manipulation

1. Basic Array Operations

Arrays in JavaScript are used to store multiple values in a single variable. You can perform various operations on arrays like adding, removing, or accessing elements.

```
// Creating Arrays
let arr1 = [1, 2, 3, 4, 5];
let arr2 = ['apple', 'banana', 'cherry'];

// Accessing Elements (indexing starts at 0)
let firstElement = arr1[0]; // 1
let lastElement = arr1[arr1.length - 1]; // 5

// Modifying Elements
arr1[1] = 20; // arr1 becomes [1, 20, 3, 4, 5]

// Array Length
let length = arr1.length; // 5

/*
| Operation          | Example                 | Description                                      |
|--------------------|-------------------------|--------------------------------------------------|
| Creating Arrays    | let arr = [1, 2, 3];    | Arrays are created using square brackets [].     |
| Accessing Elements | arr[0]                  | Access array elements by index (0-based).        |
| Modifying Elements | arr[0] = 10;            | Modify array elements by index.                  |
| Length             | arr.length              | Get the length (number of elements) of an array. |
*/
```

2. Common Array Methods

JavaScript provides several built-in methods to manipulate arrays. Below are some of the most commonly used array methods:

2.1. `push()` & `pop()`

The `push()` method adds one or more elements to the end of an array, and `pop()` removes the last element from an array.

```
let arr = [1, 2, 3];
arr.push(4); // arr becomes [1, 2, 3, 4]
let popped = arr.pop(); // removes 4, arr becomes [1, 2, 3]
```

2.2. `shift()` & `unshift()`

The `shift()` method removes the first element from an array, while `unshift()` adds one or more elements to the beginning of an array.

```
let arr = [1, 2, 3];
arr.shift(); // removes 1, arr becomes [2, 3]
arr.unshift(0); // arr becomes [0, 2, 3]
```

2.3. `concat()`

The `concat()` method is used to merge two or more arrays into a single array.

```
let arr1 = [1, 2];
let arr2 = [3, 4];
let merged = arr1.concat(arr2); // [1, 2, 3, 4]
```

2.4. `join()`

The `join()` method joins all elements of an array into a single string, with an optional separator.

```
let arr = ['apple', 'banana', 'cherry'];
let joinedStr = arr.join(', '); // "apple, banana, cherry"
```

2.5. `slice()`

The `slice()` method returns a shallow copy of a portion of an array, selected from start index to end index (non-inclusive).

```
let arr = [1, 2, 3, 4, 5];
let slicedArr = arr.slice(1, 4); // [2, 3, 4]
```

2.6. `splice()`

The `splice()` method changes the contents of an array by removing or replacing existing elements and/or adding new elements in place.

```
let arr = [1, 2, 3, 4, 5];
arr.splice(2, 2, 'a', 'b'); // removes 3 and 4, arr becomes [1, 2, 'a', 'b', 5]
```

2.7. `forEach()`

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

```
let arr = [1, 2, 3];
arr.forEach(function(element) {
    console.log(element); // Logs each element: 1, 2, 3
});
```

2.8. `map()`

The `map()` method creates a new array populated with the results of calling a provided function on every element in the array.

```
let arr = [1, 2, 3];
let doubled = arr.map(function(num) {
    return num * 2; // [2, 4, 6]
});
```

2.9. `filter()`

The `filter()` method creates a new array with all elements that pass the test implemented by the provided function.

```
let arr = [1, 2, 3, 4, 5];
let evenNumbers = arr.filter(function(num) {
    return num % 2 === 0; // [2, 4]
});
```

2.10. `reduce()`

The `reduce()` method applies a function against an accumulator and each element in the array to reduce it to a single value.

```
let arr = [1, 2, 3, 4];
let sum = arr.reduce(function(accumulator, currentValue) {
    return accumulator + currentValue; // 10
}, 0);
```

2.11. `find()`

The `find()` method returns the first element in an array that satisfies the provided testing function.

```
let arr = [1, 2, 3, 4];
let found = arr.find(function(num) {
    return num > 2; // 3
});
```

3. Summary of Array Methods

Method	Example	Description
push()	arr.push(4)	Adds one or more elements to the end of an ar
pop()	arr.pop()	Removes the last element of an array.
shift()	arr.shift()	Removes the first element of an array.
unshift()	arr.unshift(0)	Adds one or more elements to the beginning.
concat()	arr.concat([6, 7])	Merges two or more arrays into a new array.
join()	arr.join(', ')	Joins all elements into a string with a separ
slice()	arr.slice(1, 3)	Returns a portion of the array as a new array
splice()	arr.splice(2, 1, 'a')	Adds or removes elements from an array.
forEach()	arr.forEach(fn)	Executes a function for each element in the a
map()	arr.map(fn)	Creates a new array with the results of calli
filter()	arr.filter(fn)	Creates a new array with elements that pass a
reduce()	arr.reduce(fn, 0)	Reduces the array to a single value.
find()	arr.find(fn)	Returns the first element that passes a test.