Hema Coding School

YouTube Link: https://www.youtube.com/@HemaCodingSchool

Class 06: Arrays

What are Arrays?

Arrays are data structures in JavaScript used to store multiple values in a single variable. Each value in an array is called an element, and elements in an array are indexed starting from 0.

Declaration and Initialization:

Using square brackets **[]**

```
var jewelleryBox = ['Ring', 'Necklace', 'Earrings', 'Bracelet'];
```

Accessing Array Elements

```
// Accessing elements from the jewelry box
console.log("First item:", jewelleryBox[0]);  // Outputs: "Ring"
console.log("Third item:", jewelleryBox[2]);  // Outputs: "Earrings"
```

Array Methods:

The essential array methods in JavaScript:

1. **push()**: Adds one or more elements to the end of an array and returns the new length of the array.
2. **pop()**: Removes the last element from an array and returns that element.
3. **shift()**: Removes the first element from an array and returns that element.
4. **unshift()**: Adds one or more elements to the beginning of an array and returns the new length of the array.
5. **concat()**: Combines two or more arrays and returns a new array.
6. **indexOf()**: Returns the first index at which a given element can be found in the array, or -1 if it is not present.
7. **join()**: Joins all elements of an array into a string.
8. **slice()**: Returns a shallow copy of a portion of an array into a new array.
9. **splice()**: Adds or removes elements from an array at a specified index.

1. **push()**: Adds one or more elements to the end of an array and returns the new length of the array.

```
// Adding an item to the jewelry box
var newLength = jewelleryBox.push('Watch');
console.log(jewelleryBox); // Outputs: ['Ring', 'Necklace', 'Earrings', 'Bracelet', 'Watch']
console.log(newLength); // Outputs: 5
```

2. **pop()**: Removes the last element from an array and returns that element.

```
// Removing an item from the jewelry box
var removedItem = jewelleryBox.pop();
console.log(jewelleryBox); // Outputs: ['Ring', 'Necklace', 'Earrings']
console.log(removedItem); // Outputs: 'Bracelet'
```

3. **shift()**: Removes the first element from an array and returns that element.

```
// shift
var removedItem = jewelleryBox.shift();
console.log(jewelleryBox); // Outputs: ['Necklace', 'Earrings']
console.log(removedItem); // Outputs: 'Ring'
```

4. **unshift()**: Adds one or more elements to the beginning of an array and returns the new length of the array.

```
//unshift

var newLength = jewelleryBox.unshift('Pendant', 'Brooch');
console.log(jewelleryBox); // Outputs: ['Pendant', 'Brooch', 'Necklace', 'Earrings']
console.log(newLength); // Outputs: 4
```

5. **concat()**: Combines two or more arrays and returns a new array.

```
//concat

var additionalItems = ['Watch', 'Anklet'];
var combinedArray = jewelleryBox.concat(additionalItems);
console.log(combinedArray); // Outputs: ['Pendant', 'Brooch', 'Necklace', 'Earrings', 'Watch', 'Anklet']
```

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

```
//indexOf
var index = jewelleryBox.indexOf('Necklace');
```

```
console.log(index); // Outputs: 2
```

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

```
//join
var joinedString = jewelleryBox.join(', ');
console.log(joinedString); // Outputs: 'Pendant, Brooch, Necklace, Earrings'
```

8. **slice()**: Returns a shallow copy of a portion of an array into a new array.

```
//slice
var slicedArray = jewelleryBox.slice(1, 3);
console.log(slicedArray); // Outputs: ['Brooch', 'Necklace']
```

9. **splice()**: Adds or removes elements from an array at a specified index.

```
//splice
// Adding 'Anklet' at index 2
jewelleryBox.splice(2, 0, 'Anklet');
console.log(jewelleryBox); // Outputs: ['Pendant', 'Brooch', 'Anklet', 'Necklace', 'Earrings']

// Removing 'Brooch' from index 1
jewelleryBox.splice(1, 1);
console.log(jewelleryBox); // Outputs: ['Pendant', 'Anklet', 'Necklace', 'Earrings']
```

Loop in Array:

```
// Looping in array

var jewelleryBox = ['Ring', 'Necklace', 'Earrings', 'Bracelet'];

// Using a basic for loop to iterate through the array
for (var i = 0; i < jewelleryBox.length; i++) {
   console.log(jewelleryBox[i]);
}
// Outputs:
```

```
// Ring
// Necklace
// Earrings
// Bracelet
```

Conditional Statement:

```
//conditional statement
var jewelleryBox = ['Ring', 'Necklace', 'Earrings', 'Bracelet'];

// Using a for loop with a conditional statement to filter items
for (var i = 0; i < jewelleryBox.length; i++) {
    if (jewelleryBox[i].length > 5) {
        console.log(jewelleryBox[i] + ' is a long item.');
    } else {
        console.log(jewelleryBox[i] + ' is a short item.');
    }
}
// Outputs:
// Ring is a short item.
// Necklace is a long item.
// Earrings is a long item.
// Bracelet is a long item.
```

Multiple data type in Array:

```
var kitchenDraw = ['Recipe Note', 1.5, new Date(), ['Spoon A', 'Spoon B', 'Spoon C'], true];
```

Task:

Accessing Array Elements

Create an array of your favorite colors and then log the first and last elements of the array.

```
var favoriteColors = ['Blue', 'Green', 'Red', 'Yellow', 'Purple'];

console.log("First color:", favoriteColors[0]);  // Outputs: "Blue"
console.log("Last color:", favoriteColors[favoriteColors.length - 1]);  // Outputs: "Purple"
```

Adding and Removing Elements

Create an array of fruits and then add a new fruit at the end. After that, remove the first fruit from the array.

```
var fruits = ['Apple', 'Banana', 'Orange'];
// Adding a new fruit at the end
```

```
fruits.push('Grapes');
console.log("Fruits after adding:", fruits);  // Outputs: ["Apple", "Banana", "Orange", "Grapes"]
// Removing the first fruit
fruits.shift();
console.log("Fruits after removing first fruit:", fruits);  // Outputs: ["Banana", "Orange", "Grapes"]
```

Using the slice() Method

Create an array of animals and then use the **slice()** method to extract a portion of the array without modifying the original array.

```
var animals = ['Elephant', 'Lion', 'Tiger', 'Giraffe', 'Zebra'];
// Use slice() to extract a portion of the array
var extractedAnimals = animals.slice(1, 4);
console.log("Extracted animals:", extractedAnimals);  // Outputs: ["Lion", "Tiger", "Giraffe"]
console.log("Original array:", animals);  // Outputs: ["Elephant", "Lion", "Tiger", "Giraffe", "Zebra"]
```

Using the splice() Method

Create an array of cities and then use the **splice()** method to remove two cities from the array and add a new city in their place.

```
var cities = ['New York', 'London', 'Paris', 'Tokyo', 'Berlin'];
// Use splice() to remove and add cities
cities.splice(1, 2, 'Sydney');
console.log("Updated cities array:", cities);  // Outputs: ["New York", "Sydney", "Tokyo", "Berlin"]
```

Interview Questions:

What is an array in JavaScript?

- **Answer**: An array is a data structure that can hold multiple values in a single variable. The values can be of any type, including numbers, strings, objects, and even other arrays.

How do you add and remove elements from the end of an array?

- **Answer**: You can use the push method to add elements and the pop method to remove elements from the end of an array.

How do you add and remove elements from the beginning of an array?

- **Answer**: You can use the unshift method to add elements and the shift method to remove elements from the beginning of an array.

Explain the difference between slice and splice methods.

- **Answer**: The slice method returns a shallow copy of a portion of an array into a new array object, while the splice method changes the contents of an array by removing, replacing, or adding elements.

How do you flatten a multidimensional array in JavaScript?

- **Answer**: You can use the flat method to flatten a multidimensional array.

```
var array = [1, [2, [3, [4, 5]]]];
var flattenedArray = array.flat(Infinity); // Infinity to flatten all levels
console.log(flattenedArray); // Output: [1, 2, 3, 4, 5]
```