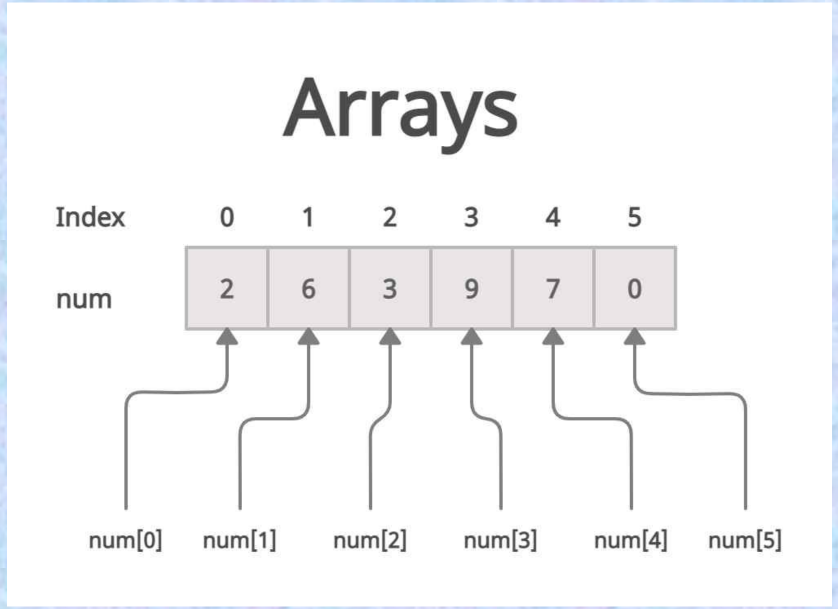

Arrays

**Ashutosh Jha,
Technical Officer,
C-DAC Patna**

Agenda

- Introductions
- Initializing Arrays
- Multi-dimensional Array
- Rest Parameters
- Adding, Updating and Reading Array Elements
- Sorting and Searching
- Array to String
- Copying Arrays

Arrays

- Data structure that stores a collection of elements, such as numbers or strings, in a single variable.
- Each element in the array is assigned a unique index that indicates its position in the collection.
- Store and manipulate large amounts of data efficiently
- Organize data in a logical and structured manner
- Perform various operations, such as sorting, searching, and filtering

Array Initialization

- An array can be created using either literal syntax or constructor syntax.
  - Literal syntax: const array_name = [item1, item2, …];
  - Constructor syntax: const array_name = new Array(item1, item2, …);
- An array element can be accessed or modified using index notation.
  - Index notation: array_name[index];
  - Index starts from 0 for the first element and goes up to array.length - 1 for the last element.

Accessing Elements

- We can access array elements using their index. The first element in an array has an index of 0, the second has an index of 1, and so on. For example:

```
const myArray = [1, 2, 3, 4, 5];
const firstElement = myArray[0]; // equals 1
 const thirdElement = myArray[2]; // equals 3
```

- We can also use negative indexes to access elements from the end of the array. For example:

```
const myArray = [1, 2, 3, 4, 5];
const lastElement = myArray[-1]; // equals 5
const secondLastElement = myArray[-2]; // equals 4
```

2 Dimensional Array

- A two dimensional array is an array of arrays, which means each element of the array is another array.
- A two dimensional array can be created using nested square brackets or nested constructors.
  - Nested square brackets:

    ```
    const matrix = [[1, 2], [3, 4], [5, 6]];
    ```
  - Nested constructors:

    ```
    const matrix = new Array(new Array(1, 2), new
    Array(3, 4), new Array(5, 6));
    ```

2 Dimensional Array

- A two dimensional array element can be accessed or modified using nested index notation.
- Nested index notation:
    - Matrix[row][column];

```
const myArray = [[1, 2, 3], [4, 5, 6], [7, 8, 9]];
const firstElement = myArray[0][0]; // equals 1
const secondElement = myArray[0][1]; // equals 2
const fifthElement = myArray[1][1]; // equals 5
```

- Here, myArray[0][0] accesses the first element of the first subarray, which is 1. Similarly, myArray[0][1] accesses the second element of the first subarray, which is 2. Finally, myArray[1][1] accesses the second element of the second subarray, which is 5.

JavaScript Variable Arguments

- Variable arguments allow a function to accept any number of arguments.
- This is useful when we don't know how many arguments the function will need to work with.
- we use the `...`syntax to declare a variable argument
  `function myFunction(...args) { // code here }`
- This function can now accept any number of arguments, which will be stored in the args array.
- The rest parameter syntax is a way to declare a function that can accept any number of arguments as an array.

JavaScript Variable Arguments

```
function sum(...nums) {
  let total = 0;
  nums.forEach(num => {
    total += num;
  });
  return total;
}
```

Array Methods

- Array methods are built-in functions that can perform various operations on arrays
- common array methods:
  - push (): add elements to the end of an array
  - pop (): remove elements from the end of an array

```
const fruits = [“apple”, “banana”];
fruits.push(“orange”); // [“apple”, “banana”, “orange”]
fruits.pop(); // [“apple”, “banana”]
```

Array Methods

- unshift(): add elements to the end beginning of an array
- shift(): remove elements from the beginning of an array

```
const fruits = [“apple”, “banana”];
fruits.shift(); // [“banana”]
fruits.unshift(“orange”); // [“orange”, “banana”]
```

Array Methods

- splice(): The splice() method is used to add or remove elements from an array. It can be used to remove elements, insert new elements, or replace existing elements. The method modifies the original array and returns the removed elements (if any).

```
const fruits = ['apple', 'banana', 'cherry', 'kiwi'];
const removed = fruits.splice(2, 1, 'orange', 'pear');
console.log(fruits); // output: ['apple', 'banana',
'orange', 'pear', 'kiwi']
console.log(removed); // output: ['cherry']
```

splice(index, howManyToRemove, item1, item2, ...)

Array Methods

- slice(): The slice() method is used to extract a section of an array and return it as a new array. The method does not modify the original array.

```
const numbers = [1, 2, 3, 4, 5];

const sliced = numbers.slice(1, 4);

console.log(sliced); // output: [2, 3, 4]
```

- concat(): The concat() method is used to merge two or more arrays into a new array. The method does not modify the original arrays.

```
const arr1 = [1, 2, 3];
const arr2 = [4, 5, 6];
const merged = arr1.concat(arr2);
console.log(merged); // output: [1, 2, 3, 4, 5, 6]
```

Array Methods

- join(): The join() method is used to join all elements of an array into a string. The method returns a new string and does not modify the original array.

```
const fruits = ['apple', 'banana', 'cherry'];
const joined = fruits.join(', ');
console.log(joined); // output: "apple, banana, cherry"
```

- sort(): The sort() method is used to sort the elements of an array. By default, it sorts the elements in ascending order. The method modifies the original array.

```
const fruits = ['banana', 'apple', 'cherry'];
fruits.sort();
console.log(fruits); // output: ['apple', 'banana', 'cherry']
```

Array Methods

- reverse(): The reverse() method is used to reverse the order of the elements in an array. The method modifies the original array.

```
const numbers = [1, 2, 3];
numbers.reverse();
console.log(numbers); // output: [3, 2, 1]
```

- indexOf(): The indexOf() method is used to find the index of the first occurrence of a specified element in an array. If the element is not found, it returns -1.

```
const fruits = ['apple', 'banana', 'cherry'];

const index = fruits.indexOf('banana');

console.log(index); // output: 1
```

Array Methods

- lastIndexOf(): The lastIndexOf() method is used to find the index of the last occurrence of a specified element in an array. If the element is not found, it returns -1.

```
const numbers = [1, 2, 3, 2];

const index = numbers.lastIndexOf(2);

console.log(index); // output: 3
```

- includes(): The includes() method is used to determine whether an array includes a certain element. It returns true if the element is found in the array, false otherwise.

```
const numbers = [1, 2, 3];
const includesTwo = numbers.includes(2);
console.log(includesTwo); // output: true
```

Array Methods

- findIndex(): The findIndex() method is used to find the index of the first element in an array that satisfies a provided testing function. If no element satisfies the testing function, it returns -1.

```
const numbers = [1, 2, 3, 4, 5];
const evenIndex = numbers.findIndex(num => num % 2 === 0);
console.log(evenIndex); // output: 1
```

Array conversion

- Array conversion is a way to convert an array into a different data type or vice versa.
- Some common ways to convert arrays are:
    - toString () or join (): convert an array into a string using a separator (default is comma).
    - Example:

```
const arr = [1 ,2 ,3];
const str = arr.toString(); // "1 ,2 ,3"
const str2 = arr.join(“-”); // "1-2-3"
```

Array conversion

- split (): convert a string into an array using a delimiter.
  - Example:
  - const str = "hello world";
  - const arr = str.split(" "); // ["hello", "world"]
- Array.from () or Array.of (): convert other data types into arrays using a mapping function (optional).
  - Example:
    const set = new Set([1 ,2 ,3]);
    const arr = Array.from(set); //[1 ,2 ,3]
    const num = Array.of(5); //5

Copying array

- Array copying is a way to create a new array with the same elements as an existing array.
- There are two types of array copying: shallow copy and deep copy.
- Shallow copy means that only the first level of elements are copied, and any nested arrays or objects are still referenced by both arrays.
- Example:
```
const arr = [1 ,2 ,[3 ,4]];
const shallowCopy = arr.slice(); // using slice method
shallowCopy[0] = 5; // changes only shallowCopy
shallowCopy[2][0] = 6; // changes both arr and shallowCopy
```

Copying array

- Deep copy means that all levels of elements are copied, and any nested arrays or objects are disconnected from the original array2.
- Example:
  ```
  const arr = [1 ,2 ,[3 ,4]];
  const deepCopy = JSON.parse(JSON.stringify(arr));
  // using JSON methods
  deepCopy[0] = 5; // changes only deepCopy
  deepCopy[2][0] = 6; // changes only deepCopy
  ```

Write your own program

- Write a function that takes an array of numbers as an argument and returns the average of those numbers.
- Write a function that takes an array of strings as an argument and returns a new array with all strings capitalized.
- Write a function that takes a two dimensional array of numbers as an argument and returns the sum of all elements in it.

Recap

- In this session, we learned about arrays and how to use them in JavaScript.
- We covered how to initialize, access, modify, loop through, and search for elements in one dimensional and two dimensional arrays.
- We also learned about variable arguments and how to use them with functions using arguments object or rest parameter syntax.
- We explored some common array methods that can manipulate, combine, sort, or find elements in arrays.
- We also learned how to convert arrays into strings or other data types using various methods like toString (), join (), split (), Array.from (), etc.
- Finally, we learned how to copy arrays using shallow copy or deep copy techniques and what are the differences between them.

Coming up…

In the next session, we will learn about object-oriented programming (OOP) and how to use classes and objects in JavaScript.

Questions

Copyright © C-DAC Patna

Thank You!!!

https://www.linkedin.com/company/c-dac-patna

https://twitter.com/CDAC_Patna

https://www.facebook.com/PatnaCDAC/

https://www.youtube.com/c/CDACPatna