

Learn JavaScript With me Day-2

1. String Methods

- Strings in JavaScript have various built-in methods.
- Examples:
 - `length` to get the length of a string.
 - `charAt()` to get a character at a specific index.
 - `concat()` to concatenate two or more strings.

2. Trim Method

- `trim()` removes whitespace from both ends of a string.
- Example:

```
let str = "  Hello World!  ";  
console.log(str.trim()); // "Hello World!"
```

3. Strings are Immutable in JS

- Strings in JavaScript cannot be changed directly; you can only create new strings by combining or modifying existing ones.
- Example:

```
let str = "Hello";  
  
str[0] = "J"; // This will not change the string.  
  
console.log(str); // "Hello"
```

4. toUpperCase() and toLowerCase()

- `toUpperCase()`: Converts a string to uppercase.
- `toLowerCase()`: Converts a string to lowercase.
- Example:

```
let str = "Hello";  
  
console.log(str.toUpperCase()); // "HELLO"  
  
console.log(str.toLowerCase()); // "hello"
```

5. Methods with Arguments – indexOf

- `indexOf(substring)`: Returns the index of the first occurrence of a substring, or `-1` if not found.
- Example:

```
let str = "Hello World";

console.log(str.indexOf("World")); // 6

console.log(str.indexOf("world")); // -1
```

6. Method Chaining

- Method chaining allows calling multiple methods on an object in a single statement.
- Example:

```
let str = "  Hello World  ";

console.log(str.trim().toUpperCase()); // "HELLO WORLD"
```

7. Slice Method

- `slice(start, end)`: Extracts a part of a string and returns it as a new string.
- Negative indexes count from the end.
- Example:

```
let str = "Hello World";

console.log(str.slice(0, 5)); // "Hello"

console.log(str.slice(-5)); // "World"
```

8. Replace & Repeat Method

- `replace(searchValue, newValue)`: Replaces the first occurrence of a string.
- `repeat(n)`: Repeats the string `n` times.
- Example:

```
let str = "Hello World";

console.log(str.replace("World", "Everyone")); // "Hello Everyone"

console.log(str.repeat(2)); // "Hello WorldHello World"
```

9. Array (Data Structure)

- Arrays in JavaScript are used to store multiple values in a single variable.
- Declared using square brackets: `[]`.
- Example:

```
let arr = [1, 2, 3, 4];  
  
console.log(arr[0]); // 1
```

10. Visualizing Arrays

- Arrays store values at indexed positions starting from 0.
- Example:

```
let arr = ["apple", "banana", "cherry"];  
  
console.log(arr); // ["apple", "banana", "cherry"]
```

11. Creating Arrays

- Arrays can be created using literal syntax `[]` or the `new Array()` constructor.
- Example:

```
let arr = new Array(3); // Creates an empty array of length 3
```

12. Arrays are Mutable

- Arrays in JavaScript can be changed even if declared with `const`.
- Example:

```
const arr = [1, 2, 3];  
  
arr[0] = 100; // Allowed, because arrays are mutable
```

13. Array Methods

- Some common array methods:
 - `push()`: Adds an element to the end of the array.
 - `pop()`: Removes the last element.
 - `shift()`: Removes the first element.
 - `unshift()`: Adds an element to the beginning.
 - `join()`: Joins all elements into a string.

14. indexOf() & includes() Method

- `indexOf(value)`: Returns the index of the first occurrence of a value in an array, or `-1` if not found.
- `includes(value)`: Returns `true` if the value exists in the array, `false` otherwise.
- Example:

```
let arr = [1, 2, 3, 4];

console.log(arr.indexOf(3)); // 2

console.log(arr.includes(5)); // false
```

15. Concatenation & Reverse

- `concat()`: Merges two or more arrays.
- `reverse()`: Reverses the array in place.
- Example:

```
let arr1 = [1, 2];

let arr2 = [3, 4];

console.log(arr1.concat(arr2)); // [1, 2, 3, 4]

console.log(arr1.reverse()); // [2, 1]
```

16. Slice in Arrays

- `slice(start, end)`: Extracts a section of an array and returns a new array.
- Example:

```
let arr = [1, 2, 3, 4];

console.log(arr.slice(1, 3)); // [2, 3]
```

17. Splice in Arrays

- `splice(start, deleteCount, item1, item2, ...)`: Adds/removes items from an array.
- Example:

```
let arr = [1, 2, 3, 4];

arr.splice(1, 2, "a", "b"); // Removes 2 elements from index 1
                             // and adds "a", "b"
```

```
console.log(arr); // [1, "a", "b", 4]
```

18. Sort in Arrays

- `sort()`: Sorts the array in place.
- By default, it sorts elements as strings.
- Example:

```
let arr = [10, 1, 3, 20];
```

```
console.log(arr.sort()); // [1, 10, 20, 3] (lexical sorting)
```

19. Arrays References

- Arrays are reference types in JavaScript, meaning changes to an array in one variable affect all references to it.
- Example:

```
let arr1 = [1, 2, 3];
```

```
let arr2 = arr1;
```

```
arr2[0] = 100;
```

```
console.log(arr1); // [100, 2, 3]
```

20. Constant Arrays

- Arrays declared with `const` cannot be reassigned, but their elements can be modified.
- Example:

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

```
arr[0] = 10; // Allowed
```

```
arr = [4, 5, 6]; // Error
```

21. Nested Arrays

- Arrays within arrays are called nested arrays.
- Example:

```
let nestedArr = [[1, 2], [3, 4]];
```

```
console.log(nestedArr[0][1]); // 2
```