Day :12 JavaScript Notes.

Strings in JavaScript

Strings in JavaScript are used to store and manipulate text. A string can be enclosed in **single quotes (')**, **double quotes (")**, or **backticks (``` `` ```)**. Backticks are especially useful when working with template literals (like embedding variables or expressions).

Escape Sequences in Strings

When you want to include special characters in a string, you can use escape sequences:

- \n – Newline
- \t – Tab space
- \' – Single quote
- \" – Double quote
- \` – Backtick
- \\ – Backslash

Example:

```
let text = "He said: \"Hello there!\"\nLet's learn JavaScript.";

console.log(text);
```

Common String Operations

1. Length of a String

The .length property returns the number of characters in a string:

```
let str = "Hello";

console.log(str.length);  // Output: 5
```

---

2. Accessing Characters

You can access characters using index notation or the charAt() method:

```
console.log(str[0]);        // Output: 'H'
console.log(str.charAt(1)); // Output: 'e'
```

---

3. Changing Case

JavaScript allows converting strings to uppercase or lowercase:

```
console.log("hello".toUpperCase()); // Output: "HELLO"
console.log("WORLD".toLowerCase()); // Output: "world"
```

String Search Methods

indexOf(substr, position)

This method returns the index of the **first occurrence** of a substring, optionally starting from a given position. If not found, it returns -1.

```
let str = "Hello, world!";
console.log(str.indexOf("o"));     // Output: 4
console.log(str.indexOf("o", 5));  // Output: 8
console.log(str.indexOf("xyz"));   // Output: -1
```

---

lastIndexOf(substr, position)

Searches the string **from right to left** and returns the index of the **last occurrence** of a substring. You can optionally provide a position to start searching backwards from.

```
console.log(str.lastIndexOf("o"));     // Output: 8
console.log(str.lastIndexOf("o", 7));  // Output: 4
console.log(str.lastIndexOf("xyz"));   // Output: -1
```

---

includes(substr, position)

Checks whether the given substring exists in the string. It's **case-sensitive** and optionally accepts a starting position.

```
let str2 = "JavaScript is awesome!";
console.log(str2.includes("Script"));    // true
console.log(str2.includes("script"));    // false
console.log(str2.includes("is", 18));    // false
```

startsWith(substr, position)

Checks if the string starts with the given substring, optionally from a specific index.

```
console.log(str2.startsWith("Java"));        // true
console.log(str2.startsWith("Script"));      // false
console.log(str2.startsWith("Script", 4));  // true
```

endsWith(substr, length)

Checks whether the string ends with the given substring. You can specify how many characters from the start should be considered.

```
console.log(str2.endsWith("awesome!"));     // true
console.log(str2.endsWith("is", 13));       // true (checks first 13 characters)
console.log(str2.endsWith("Java"));         // false
```

Extracting Substrings

substring(startIndex, endIndex)

Extracts characters from a string **between** the startIndex and endIndex (excluding endIndex).

```
console.log("JavaScript".substring(0, 4));  // Output: "Java"
```

substr(start, length)

Extracts a substring starting from a given index and for the specified number of characters.

```
console.log("Hello".substr(1, 3)); // Output: "ell"
```

⚠ Note: substr() is considered **legacy** and may be less recommended in modern JavaScript, but it's still widely used and supported.
