📘JavaScript Strings – Complete Notes

🕰️1. What is a String in JavaScript?

- A **string** is a sequence of characters (letters, numbers, symbols, spaces).
- Strings are used to represent text in JavaScript.
- Strings are enclosed in:
- **Single quotes** → `'Hello'`
- **Double quotes** → `"Hello"`
- **Backticks (Template literals)** → `` `Hello` ``

👉Example:

```javascript
let single = 'Hello';
let double = "World";
let backtick = `Hello World`;

console.log(single);   // Hello
console.log(double);   // World
console.log(backtick); // Hello World
```

🕰️2. String Creation

```javascript
let str1 = "JavaScript";
let str2 = 'Programming';
let str3 = `String Example`; // template literal
```

🕰️3. String Properties

- **length** → returns number of characters.

```javascript
let txt = "Hello";
console.log(txt.length); // 5
```

🕰️4. String Access

- **Bracket notation** → `str[index]`
- **charAt(index)** → returns character at position

```
let str = "JavaScript";

console.log(str[0]);       // J
console.log(str.charAt(4)); // S
console.log(str[str.length - 1]); // t
```

🕰️5. Escape Characters

Some characters need `\` (backslash):

```
let str = "He said, \"JavaScript is awesome!\"";
let newLine = "Hello\nWorld";
let tab = "Hello\tWorld";

console.log(str);     // He said, "JavaScript is awesome!"
console.log(newLine); // Hello (newline) World
console.log(tab);     // Hello   World
```

🕰️6. String Concatenation

Joining strings:

```
let first = "Hello";
let second = "World";

// Using +
let result = first + " " + second;
console.log(result); // Hello World

// Using +=
first += " Everyone";
console.log(first); // Hello Everyone

// Using template literals
```

```
let name = "Umesh";
console.log(`Hi ${name}, welcome to JavaScript!`);
```

🕰️7. String Methods (Most Important)

JavaScript has many **built-in string methods**:

🔗Case Conversion

```
let text = "JavaScript";
console.log(text.toUpperCase()); // JAVASCRIPT
console.log(text.toLowerCase()); // javascript
```

🔗Trimming

```
let str = "   hello world   ";
console.log(str.trim()); // "hello world"
console.log(str.trimStart()); // "hello world   "
console.log(str.trimEnd());   // "   hello world"
```

🔗Searching

```
let str = "I love JavaScript";

console.log(str.indexOf("love"));   // 2
console.log(str.lastIndexOf("a"));  // 10
console.log(str.includes("Java"));  // true
console.log(str.startsWith("I"));   // true
console.log(str.endsWith("Script")); // true
```

🔗Extracting Parts

```
let str = "Hello World";

console.log(str.slice(0, 5)); // Hello
console.log(str.substring(0, 5)); // Hello
console.log(str.substr(6, 5)); // World (deprecated, avoid in new code)
```

🔗Replace

```
let txt = "I like Java";
console.log(txt.replace("Java", "JavaScript")); // I like JavaScript
console.log(txt.replace(/JAVA/i, "Python"));    // I like Python
console.log("apple apple".replaceAll("apple", "orange")); // orange orange
```

🔗Split (Convert to Array)

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

🔗Repeat

```
console.log("Hi ".repeat(3)); // Hi Hi Hi
```

🔗Padding

```
let num = "5";
console.log(num.padStart(4, "0")); // 0005
console.log(num.padEnd(4, "*"));   // 5***
```

🕰️8. Template Literals (Backticks ``)

- Support multi-line strings
- Embed variables with `${ }`

```
let name = "Umesh";
let age = 20;

let intro = `My name is ${name}.
I am ${age} years old.
I love JavaScript!`;

console.log(intro);
```

🕰️9. String Comparison

```
console.log("apple" === "apple");  // true
console.log("apple" < "banana");   // true (dictionary order)
console.log("A" < "a");            // true (because ASCII value of A < a)
```

🕰️10. Looping Over Strings

```
let str = "JS";

for (let char of str) {
  console.log(char);
}
// Output:
// J
// S
```

🕰️11. String Objects vs Primitives

```
let str1 = "hello"; // string primitive
let str2 = new String("hello"); // string object

console.log(typeof str1); // string
console.log(typeof str2); // object
```

👉Usually, we use **string literals** (not `new String()`).

🕰️12. Useful Real-Life Examples

🔗Reverse a String

```
let str = "hello";
let reversed = str.split("").reverse().join("");
console.log(reversed); // olleh
```

🔗Count Characters

```javascript
let str = "programming";
console.log(str.length); // 11
```

🔗Check Palindrome

```javascript
function isPalindrome(str) {
  let reversed = str.split("").reverse().join("");
  return str === reversed;
}

console.log(isPalindrome("madam")); // true
console.log(isPalindrome("hello")); // false
```

📝Summary

👈String Basics → Quotes, escape characters
👈Properties → `.length`
👈Access → `[index]`, `.charAt()`
👈 Methods → `toUpperCase()`, `slice()`, `replace()`, `split()`, `trim()`, `includes()`, `repeat()`, etc.
👈Template Literals → `` `Hello ${name}` ``
👈String Comparison → Lexicographic (dictionary order)
👈String Objects vs Literals
👈Real-world problems → reverse, palindrome, searching