JavaScript String Methods

2nd edition

by

Ankush Bhagat

Character Access

1. charAt(index)

Returns the character at a specified index.

```
let str = "Hello";
console.log(str.charAt(1)); // "e"
```

2. charCodeAt(index)

Returns the Unicode value of the character at a specified index.

```
let str = "Hello";
console.log(str.charCodeAt(1)); // 101 (Unicode of 'e')
```

3. codePointAt(position)

Returns the code point value of the character at the specified position.

```
let str = "𝌆";
console.log(str.codePointAt(0)); // 119558
```

String Comparison

4. localeCompare(string)

Compares two strings in the current locale.

```
let str1 = "a";
let str2 = "b";
console.log(str1.localeCompare(str2)); // -1 (a comes before b)
```

String Searching

5. includes(searchString[, position])

Checks if the string contains the specified substring

```
let str = "Hello world";
console.log(str.includes("world")); // true
```

6. indexOf(searchValue[, fromIndex])

Returns the position of the first occurrence of a specified value.

```
let str = "Hello world";
console.log(str.indexOf("o")); // 4
```

7. lastIndexOf(searchValue[, fromIndex])

Returns the position of the last occurrence of a specified value.

```
let str = "Hello world";
console.log(str.lastIndexOf("o")); // 7
```

8. startsWith(searchString[, position])

Checks if the string starts with the specified substring.

```
let str = "Hello world";
console.log(str.startsWith("Hello")); // true
```

9. endsWith(searchString[, length])

Checks if the string ends with the specified substring.

```
let str = "Hello world";
console.log(str.endsWith("world")); // true
```

10. match(regexp)

Searches for a match against a regular expression and returns the matches.

```
let str = "abc123";
console.log(str.match(/\d+/)); // ["123"]
```

11. matchAll(regexp)

Returns an iterator for all matched groups in a string.

```
let str = "abc123def456";
let matches = str.matchAll(/\d+/g);
for (const match of matches) {
  console.log(match[0]); // "123", "456"
}
```

12. search(regexp)

Executes a search for a match between a regular expression and the string.

```
let str = "Hello world";
console.log(str.search(/world/)); // 6
```

13. replace(searchFor, replaceWith)

Replaces occurrences of a specified value or a regular expression.

```
let str = "Hello world";
console.log(str.replace("world", "JavaScript")); // "Hello JavaScript"
```

14. replaceAll(searchFor, replaceWith)

Replaces all occurrences of a specified value or a regular expression.

```
let str = "apple banana apple";
console.log(str.replaceAll("apple", "orange")); // "orange banana orange"
```

15. padStart(targetLength[, padString])

Pads the current string from the start to reach the target length.

```
let str = "5";
console.log(str.padStart(3, "0")); // "005"
```

16. padEnd(targetLength[, padString])

Pads the current string from the end to reach the target length.

```
let str = "5";
console.log(str.padEnd(3, "0")); // "500"
```

String Modification

17. slice(beginIndex[, endIndex])

Extracts a section of the string.

```
let str = "Hello world";
console.log(str.slice(0, 5)); // "Hello"
```

18. substring(indexStart[, indexEnd])

Extracts a part of the string between two indices.

```
let str = "Hello world";
console.log(str.substring(0, 5)); // "Hello"
```

19. substr(start[, length])

Returns a portion of the string starting at the specified index and optionally up to the given length.

```
let str = "Hello world";
console.log(str.substr(6, 5)); // "world"
```

20. split([separator[, limit]])

Splits a string into an array of substrings.

```
let str = "apple, banana, orange";
console.log(str.split(", ")); // ["apple", "banana", "orange"]
```

21. concat(string1, string2, ...)

Combines two or more strings.

```
let str1 = "Hello";
let str2 = "World";
console.log(str1.concat(" ", str2)); // "Hello World"
```

22. toLowerCase()

Converts the string to lowercase.

```
let str = "HELLO WORLD";
console.log(str.toLowerCase()); // "hello world"
```

23. toUpperCase()

Converts the string to uppercase.

```
let str = "hello world";
console.log(str.toUpperCase()); // "HELLO WORLD"
```

24. trim()

Removes whitespace from both ends of the string.

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

25. trimStart() or trimLeft()

Removes whitespace from the beginning of the string.

```
let str = "   Hello world   ";
console.log(str.trimStart()); // "Hello world   "
```

26. trimEnd() or trimRight()

Removes whitespace from the end of the string.

```
let str = "   Hello world   ";
console.log(str.trimEnd()); // "   Hello world"
```

27. normalize([form])

Returns the Unicode normalization form of the string.

```
let str = "\u00F1"; // ñ
console.log(str.normalize("NFC")); // "ñ"
```

String Encoding and Decoding

28. toString()

Returns the string representation of an object.

```
let str = new String("Hello");
console.log(str.toString()); // "Hello"
```

29. valueOf()

Returns the primitive value of a string object.

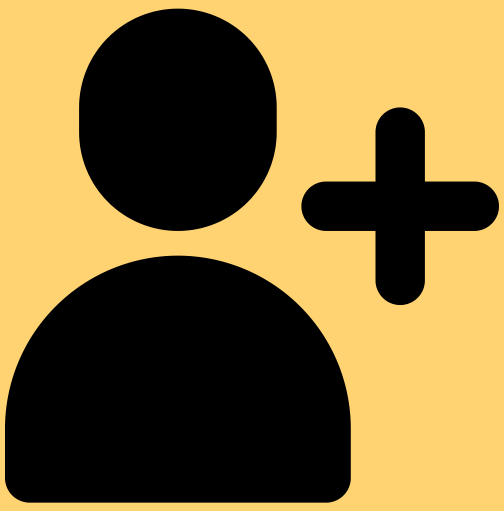
```
let str = new String("Hello");
console.log(str.valueOf()); // "Hello"
```

Raw String Methods

30. String.raw()

A template string tag that returns the raw string, escaping backslashes.

```
let str = String.raw`Hello\nWorld`;
console.log(str); // "Hello\\nWorld" (backslashes aren't interpreted)
```



Follow for more