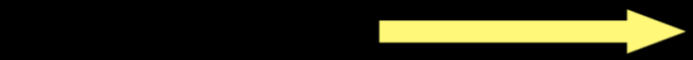
JavaScript string methods with examples

Tarak Shah



length

The length property returns the length of a string.

```
let text = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

//Use
console.log(text.length);

//Output
26
```

Tarak Shah

concat()

- concat() joins two or more strings.
- The concat() method can be used instead of the plus(+) operator to join string.

```
let text1 = "Hello";
let text2 = "World";

//Use
text1.concat(" ", text2);

//Output
Hello World
```

Tarak Shah

slice()

JS

- The slice() method extracts a part of a string and returns the extracted part as a new string.
- It takes 2 parameters: start and end position. If a parameter is negative, the position is counted from the end of the string.

```
let text = "Apple, Banana, Kiwi";

//Use
text.slice(7,13);
text.slice(-19,-14)

//Output
Banana
Apple
```

Tarak Shah

substring()

- substring() is similar to slice().
- The difference is that start and end values less than 0 are treated as 0 in substring().

```
let str = "Apple, Banana, Kiwi";

//Use
str.substring(7, 13);

//Output
Banana
```

Tarak Shah

substr()

- substr() is similar to slice().
- The difference is that the second parameter specifies the length of the extracted part.

```
let str = "Apple, Banana, Kiwi";

//Use
str.substr(15,6);

//Output
Kiwi
```

Tarak Shah

replace()

- The replace() method replaces a specified value with another value in a string.
- The replace() method replaces only the first match and this is case sensitive method.

```
let text = "Please visit YouTube!";

//Use
text.replace("YouTube", "Google");

//Output
Please visit Google!
```

Tarak Shah

replaceAll()

JS

- In 2021, JavaScript introduced the string method replaceAll().
- This method replace a specified value in whole string instead of first match.

```
let text = "I love cats, cats are very easy to love.";

//Use
text.replaceAll("cats","dogs");

//Output
I love dogs, dogs are very easy to love.
```

Tarak Shah

toLowerCase()

This method converts all characters of string into lowercase.

```
let text1 = "Hello World!";

//Use
text1.toLowerCase();

//Output
hello world!
```

Tarak Shah

toUpperCase()

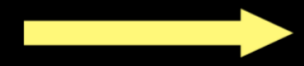
This method converts all characters of string into uppercase.

```
let text1 = "Hello World!";

//Use
text1.toUpperCase();

//Output
HELLO WORLD!
```

Tarak Shah



split()

A string can be converted to an array with the split() method.

```
let text = "a,b,c,d,e,f";

//Use
const myArray = text.split(",");
console.log(myArray[0]);
console.log(myArray[3]);

//Output
a
d
```

Tarak Shah

trim()

The trim() method removes whitespace from both sides of a string.

```
let text1 = "      Hello World!      ";

//Use
text1.trim();

//Output
Hello World!
```

Tarak Shah

trimStart()

The trimStart() method works like trim(), but removes whitespace only from the start of a string.

```
let text1 = "      Hello World!      ";

//Use
text1.trimStart();

//Output
Hello World!
```

Tarak Shah

trimEnd()

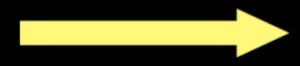
The trimEnd() method works like trim(), but removes whitespace only from the end of a string.

```
let text1 = "      Hello World!      ";

//Use
text1.trimEnd();

//Output
     Hello World!
```

Tarak Shah



charAt()

The charAt() method returns the character at a specified index (position) in a string.

```
let text1 = "Hello World!";

//Use
text.charAt(0);

//Output
H
```

Tarak Shah

charCodeAt()

The charCodeAt() method returns the unicode of the character at a specified index in a string.

```
let text1 = "Hello World!";

//Use
text.charCodeAt(0);

//Output
72
```

Tarak Shah

Tarak Shah

Full Stack Web Developer 👨‍💻 | Digital Creator 💡