



JS STRING METHODS COURSE

APRIL 2022

YONATAN BENEZRA

- DO NOT DISTRIBUTE -

- DO NOT COPY – ALL RIGHTS RESERVED TO THE
AUTHOR –

String Methods and Properties

Primitive values, such as "Ron Aharon", cannot normally have properties or methods (because they are not objects).

However, when using JavaScript, methods and properties can be applied to primitive values as well, because JavaScript treats primitive values as objects when executing methods and properties.

JavaScript String Length

The length property returns the length of a string.

Example:

```
let txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
let length = txt.length;
```

Exercise:

console.log the length of your name.

Replacing String Content

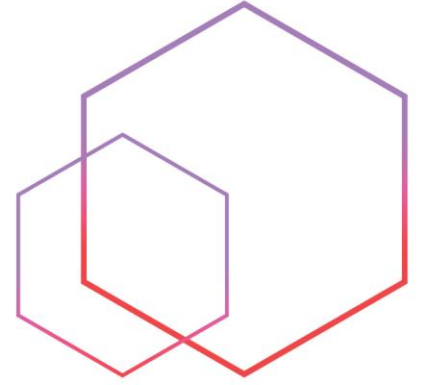
The replace() method replaces a specified value with another value in a string.

Example:

```
const greetingTemplate= "Hey _____, how are you?";  
const greeting = text.replace("_____", "Dani");
```

*Note:

The replace() method will replace the first match only.



Converting to Upper and Lower Case

A string is converted to upper case with `toUpperCase()`: method.
A string is converted to lower case with `toLowerCase()`: method.

Simple as that! Let's see some examples:

toUpperCase()

```
let lowerCasedText= "hello world!";  
let upperCasedText = lowerCasedText.toUpperCase();
```

toLowerCase()

```
let upperCasedText = "Hello World!";  
let lowerCasedText= upperCasedText.toLowerCase();
```

Go ahead, try it out!

trim()

The `trim()` method is one that you probably won't remember, but you will have fun using it when necessary.

It removes whitespace from both sides of a string:

```
let example = "    Hello World!    ";  
let trimmedText= example.trim();
```

JavaScript String Padding

ECMAScript 2017 added two String methods: `padStart()` and `padEnd()` to support padding at the beginning and at the end of a string.

Extracting String Characters

There are 3 methods for extracting string characters:

- `charAt(position)`
- `charCodeAt(position)`
- Property access `[]`

charAt()

```
let text = "HELLO WORLD";  
let char = text.charAt(0); // will return "h"
```

charCodeAt()

```
let text = "HELLO WORLD";  
let char = text.charCodeAt(0); // will return "h"
```

Property Access

```
let text = "HELLO WORLD";  
let char = text[0]; // will return "h"
```

Extracting String Portions

There are 3 methods for extracting a part of a string:

- slice(start, end)
- substring(start, end)
- substr(start, length)

slice()

slice() extracts a part of a string and returns the extracted portion in a new string.
The method takes 2 parameters: the start position, and the end position (end not included).

Example:

Slice out a portion of a string from position 7 to position 13 (13 not included):

```
let str = "Table, Katana, Rock";  
let part = str.slice(7, 13);
```

Reminder

JavaScript counts positions starting from zero.

JavaScript String substr()

substr() is similar to slice(), the difference being that the second parameter specifies the length of the extracted portion.

Example:

```
let str = "Table, Katana, Rock";  
let part = str.substr(7, 6);
```

Converting a String to an Array

If you want to work with a string as an array, you can convert it to an array using the following:

```
split()
```

A string can be converted to an array using the split() method:

```
text.split(",")    // Split on commas  
text.split(" ")    // Split on spaces  
text.split("|")    // Split on pipe
```

If the separator is omitted, the returned array will contain the whole string in index [0].

If the separator is "", the returned array will be an array of single characters

Exercise:

Given the following string:

" 123Hello45 "

create a function that returns the following:

[h,e,l,l,o]