

Most common string operations

Strings are commonly used in JavaScript so we have collected the most useful operations on strings.

Create a string

There's multiple ways to do it, and all of these methods will result the same type of strings:

```
const s1 = 'I am a string';
const s2 = "So am I"; // You can use either ' or ", it's only up to
you (but ' is more common)
const s3 = `Me too`; // Template string, useful if you want to
interpolate the result of an expression
```

Length of a string

The **length** property on a string will return it:

```
console.log('Hello Cubix'.length); //11
```

Get a character at an index

It's possible to get it with the [charAt](#) method:

```
const s = 'almafa';

console.log(s.charAt(0)); // indexing from 0
console.log(s.charAt(s.length - 1)); // last character
console.log(s.charAt(100)); // if the string is shorter than the
index then it will return an empty string
```

String concatenation

The easiest way to do is the **+** operator (but you can use a template string also):

```
const s1 = 'apple';
const s2 = 'banana';
const s = s1 + s2;

console.log(s); // appplebanana
```

Strings are immutable in JavaScript, so it means that every operation on strings will result in a new string.

Substring

It's possible to do with the [slice](#) method:

```
const s = 'almafa';

console.log(s.slice(0, 4)); // first parameter: first index to
remove, second parameter: first index which will STAY in the string
console.log(s.slice(4)); // If we omit the second parameter then it
will remove everything from that index
console.log(s); // the original string won't be modified
```

Further string operations

List of string operations: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String