

2. Strings

Strings in JavaScript are objects that represent a sequence of characters. They are used to store and manipulate text.

A JS string could have zero or more characters. Those characters are written inside, between, a single or double quotes.

```
let name1 = "Peter Smith";  
let name2 = 'Kat Smith';
```

You can use quotes inside a string, as long as they don't match the quotes surrounding the string:

```
let answer1 = "It's alright";      (output) → It's alright  
let answer3 = 'He is called "Johnny"'; (output) → He is called "Johnny"
```

Strings are indexed

Strings are considered an array of characters, therefore, each character in a string has index. The first character in a string, the leftmost character, has an index of zero, and the index number increases as we count up from left to right of the string:

Kat Smith									
	↑	↑	↑	↑	↑	↑	↑	↑	↑
Index number	0	1	2	3	4	5	6	7	8

To access to a character in a string, we simply type the string name and type the index number in between a squared brackets right next to the string name:

```
let name = 'Kat Smith';  
name[2]      (output) → t  
name[30]     (output) → undefined
```

Strings methods and properties

Methods are function belonging to an object. The context of a method (this value) equals the object the method belongs to. Methods are functions stored as object properties. Methods can be accessed with the following syntax:

Object.methodName();

On the other hand, methods used to obtain information about an object are known as **accessor methods or properties**. Properties can be accessed as:

Object.propertyName;

Checking the length of a string

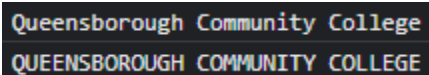
One property that you can use with strings is the **.length**, which returns the number of characters contained in the string object.

```
let text = "Hello World!";  
let msgLength = text.length;  
msgLength  
(output) → 12
```

Converting to Upper and Lower Case

A string can be converted to upper case with **toUpperCase()** or to lower case with **toLowerCase()** method.

```
let myString = "Queensborough Community College";  
console.log(myString);  
let myUpperString = myString.toUpperCase();  
console.log(myUpperString);
```



```
Queensborough Community College  
QUEENSBOROUGH COMMUNITY COLLEGE
```

Searching for a character in a string

JS uses the method **search()** to find characters within a string. It returns the location of the first searched character. The characters that we are looking for are placed in between the parenthesis. Also, for the count of the character position, the first character position has an index of zero.

```
let myString = "Queensborough Community College";
console.log(myString);
let stringLocation = myString.search('Co')
console.log(stringLocation);
```

```
Queensborough Community College
14
```

Remove white spaces

Method **trim()** removes the white spaces at the beginning and end of a string but not the space in between word.

```
let name = "    Willim Smith";
name.trim();
```

→

```
name
'    Willim Smith'
'Willim Smith'
```

Combining methods

We can also combine methods by using a dot in between methods.

```
let name = "    Hello        World        ";
name.toUpperCase().trim();
```

→

```
'HELLO        WORLD'
```

Method with argument

Subtracting characters

The method uses to subtract characters from a string is **substr(x,y)**. The **substr(x,y)** method has two parameters, x = the position of the character, within the string, and y = the number of character subtracted from the position.

Syntax

substr(position of character, number of character to be subtracted from the position)

```
let myString = "Queensborough Community College";
let subWord = myString.substr(6,5);
```

```
subWord
'borou'
```

```
let myString = "Queensborough Community College";
console.log(myString);
let stringLocation = myString.search('Co')
let substrLocation = myString.substr(stringLocation,5);
console.log(substrLocation);
```

```
Queensborough Community College
Commu
```

Replacing string content

The **replace(x,y)** method replaces a specified value with another value in a string. The method has two attributes, x = the character/s to be replaced and y = the replace characters.

```
let myString = "Queensborough Community College";
console.log(myString);
let replaceString = myString.replace('Community', 'PARK');
console.log(replaceString);
```

```
Queensborough Community College
Queensborough PARK College
```

```
let myString = "Queensborough Community College";
console.log(myString);
let replaceString = myString.replace('Co', 'PARK');
console.log(replaceString);
```

```
Queensborough Community College
Queensborough PARKmmunity College
```

Extracting characters from a string

The **slice()** method extracts a section of a string and returns it as a new string without modifying the original string.

```
const str = 'The quick brown fox jumps over the lazy dog.';
newString = str.slice(31);
```

```
newString
'the lazy dog.'
```

```
newString = str.slice(4,19);
```

```
newString
'quick brown fox'
```

```
newString = str.slice(-6);
```

```
newString
'y dog.'
```

```
newString = str.slice(-12, -5);
```

```
newString
'he lazy'
```

```
newString = str.slice(-5, -12);
```

```
newString
''
```