

# M1 - 3 - JavaScript Strings

## Métodos string

Las funciones o métodos string, son como su propio nombre indica métodos que utilizaremos para tratar y manipular las cadenas de texto.

### length

`length` es una propiedad que nos permite conocer la longitud de un string, es decir, el número de caracteres que forman parte de la cadena de texto.

```
>> 'hola'.length
← 4
```

### toUpperCase() y toLowerCase()

Ambos métodos nos convierten o transforman toda nuestra cadena a mayúsculas o a minúsculas. `toUpperCase()` para convertirlo a mayúsculas y `toLowerCase()` a minúsculas.

```
>> 'hola'.toUpperCase()
← "HOLA"
>> 'HOLA'.toLowerCase()
← "hola"
>>
```

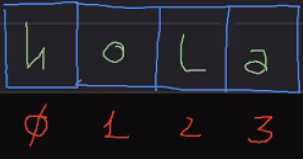
### indexOf()

`indexOf()` nos devuelve la posición en la que se encuentra, dentro de nuestra cadena de texto, una letra o un texto en concreto. Si le pasamos una cadena de texto, nos devolverá la posición donde se encuentra el primer carácter del texto que le pasamos por paréntesis. Lo que tenemos que tener en cuenta es que en los strings (cadenas de texto) el primer carácter tiene una posición 0, siempre se empieza contando que la primera posición va a tener el valor 0.

```
>> 'hola'.indexOf('a')
← 3
```

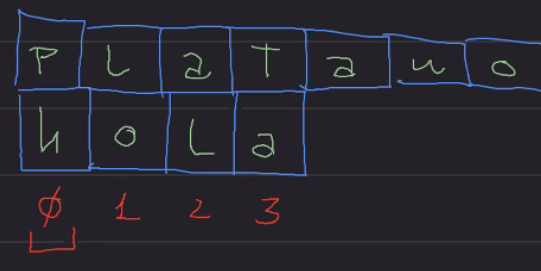
Y si no encuentra el valor que le pasamos entre paréntesis nos devuelve un -1

```
>> 'hola'.indexOf('a')
< 3
>> 'hola'.indexOf('la')
< 2
>> 'hola'.indexOf('zz')
< -1
>> |
```



Otra de las cosas que tenemos que tener en cuenta es que si por ejemplo, estamos buscando la posición de la letra 'a' en una cadena de texto y dicha cadena tiene más de un 'a' sólo nos devolverá la posición de la primera. Lo que podemos hacer en este caso, es indicar también a partir de qué posición queremos que busque dicho carácter.

```
>> 'hola'.indexOf('a')
< 3
>> 'hola'.indexOf('la')
< 2
>> 'hola'.indexOf('zz')
< -1
>> 'platano'.indexOf('a')
< 2
>> 'platano'.indexOf('a', 3)
< 4
```



## substring()

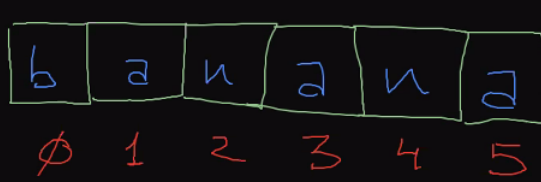
**substring()** es un método que nos permite crear una cadena de texto a partir de otra cadena de texto, entre paréntesis le vamos a indicar dos valores el primero se corresponde con la posición en la que empiezo, y la segunda es la posición en la que acabo pero esa posición no se incluye. Recordar que el primer carácter de un string tiene una posición 0.

```
>> 'banana'.substring(2,3)
< "n"

>> 'banana'.substring(2,4)
< "na"

>> 'banana'.substring(2,5)
< "nan"

>> 'banana'.substring(2,6)
< "nana"
```



Si solo le pasamos un número por parámetro, le indicamos que queremos copiar una cadena que vaya desde la posición que le indicamos entre paréntesis hasta el final de la cadena desde la que partimos.

patata.substring(2,2)

```
>> 'banana'.substring(2)
< "ana"
```

## replace()

A través del método `replace()` podemos sustituir un carácter o varios por otros, entre paréntesis le indicamos primero el o los caracteres que queremos sustituir, segundo los caracteres o carácter por el que los queremos sustituir.

```
>> 'hola'.replace('a', 'e')
< "hole"

>> 'banana'.replace('a', 'e')
< "benana"
```

## toString()

El método `toString()` podremos pasar de un valor numérico a un valor string.

```
>> let numero = 12
< undefined

>> numero
< 12

>> numero = numero.toString()
< "12"

>> numero
< "12"

>>
```