

Funciones para tratamiento de cadenas

length: método que retorna la longitud de una cadena.

```
let cadena="ABCDE";  
cadena.length; //retorna 5
```

indexOf(textoBuscado): método que retorna el índice de (o sea la posición de) la primer ocurrencia de textoBuscado en una cadena. (retorna -1 si no esta, admite un 2do parámetro entero para iniciar la búsqueda desde ahí)

```
let cadena="Tengo que hacer muchos ejercicios";  
cadena.indexOf('hacer'); //retorna 10
```

lastIndexOf(textoBuscado): método que retorna el índice de (o sea la posición de) la última ocurrencia de textoBuscado en una cadena. (retorna -1 si no esta, admite también un 2do parámetro entero para iniciar la búsqueda desde ahí)

```
let cadena="Tengo que hacer y hacer y hacer muchos ejercicios";  
cadena.lastIndexOf('hacer'); //retorna 26
```

substring(inicio, final) metodo que retorna la porción de la cadena entre las posiciones inicio y final.

```
let cadena="Tengo que hacer y hacer y hacer muchos ejercicios";  
cadena.substring(10,15); //retorna "hacer"
```

substr(inicio, largo) metodo que retorna la porción de la cadena de tamaño largo a partir de la posición inicio.

```
let cadena="Tengo que hacer y hacer y hacer muchos ejercicios";  
cadena.substr(10,21); //retorna "hacer y hacer y hacer"
```

charAt() método que sirve para obtener el caracter que se encuentra en una posicion determinada de una cadena.

```
let cadena="Tengo que hacer muchos ejercicios";  
cadena.charAt(0); //retorna 'T'  
cadena.charAt(7); //retorna 'u'
```

replace() método que retorna una cadena en la cual se ha reemplazado el primer parametro por el segundo. Atencion: este metodo distingue entre mayusculas y minusculas. Y solo realiza el reemplazo de la primer ocurrencia.

```
let cadena="Tengo que hacer y hacer y hacer muchos ejercicios";  
cadena.replace("muchos","muchisimos"); //retorna un texto que es copia
```

de cadena en la cual esta reemplazada la palabra muchos por muchisimos.

```
let cadena="Tengo que hacer y hacer y hacer muchos ejercicios";
cadena.replace("hacer","HACER"); ///retorna un texto que es copia de
cadena en la cual solo esta reemplazada la primer palabra "hacer" por
"HACER"
```

concat() método que une la cadena a la que se aplica con las cadenas ingresadas como parametros..

```
let cadena1="Tengo que hacer ";
let cadena2="y hacer ";
let cadena3="muchos ejercicios";
cadena.concat(cadena1, cadena2, cadena3); //retorna una cadena con el
siguiente contenido: "Tengo que hacer y hacer muchos ejercicios"
```

trim() método que sirve para eliminar los espacios en ambos extremos de una cadena.

```
let cadena="    Tengo que hacer muchos ejercicios    ";
cadena.trim(); //retorna "Tengo que hacer muchos ejercicios"
```

toUpperCase() método que retorna una cadena con todos sus caracteres pasados a mayúsculas.

```
let cadena="Tengo que hacer y hacer y hacer muchos ejercicios";
cadena.toUpperCase(); //retorna "TENGO QUE HACER Y HACER Y HACER MUCHOS
EJERCICIOS"
```

toLowerCase() método que retorna una cadena con todos sus caracteres pasados a minúsculas.

```
let cadena="Tengo que HACER Y HACER Y HACER MUCHOS ejercicios";
cadena.toLowerCase(); //retorna "tengo que hacer y hacer y hacer muchos
ejercicios"
```

toString() método que (aplicado a un número) retorna una cadena que representa los caracteres de cada dígito.

```
let numero=2001;
numero.toString(); //retorna "2001"
```

parseInt() método que convierte una cadena con caracteres numéricos sin símbolo decimal en un número entero.

```
let cadena="2001";
parseInt(cadena); //retorna 2001
```

parseFloat() método que convierte una cadena con caracteres numericos con símbolo decimal en un número decimal.

```
let cadena="2001.40";  
parseFloat(cadena); //retorna 2001.40
```

split() método que convierte una cadena en un arreglo, mediante un separador. Si no se ingresa separador el arreglo resultante tiene un unico elemento con la cadena completa. Si se ingresa una cadena vacia como separador el arreglo resultante tiene tantos elementos como caracteres tenga la cadena con un caracter cada uno.

```
let cadena="Tengo que hacer muchos ejercicios";  
arreglo=cadena.split(" "); //retorna el siguiente arreglo:  
arreglo[0] -> Tengo  
arreglo[1] -> que  
arreglo[2] -> hacer  
arreglo[3] -> muchos  
arreglo[4] -> ejercicios
```

```
let cadena="Tengo";  
arreglo=cadena.split(""); //retorna el siguiente arreglo:  
arreglo[0] -> T  
arreglo[1] -> e  
arreglo[2] -> n  
arreglo[3] -> g  
arreglo[4] -> o
```

```
let cadena="a,b,c,d,e";  
arreglo=cadena.split(","); //retorna el siguiente arreglo:  
arreglo[0] -> a  
arreglo[1] -> b  
arreglo[2] -> c  
arreglo[3] -> d  
arreglo[4] -> e
```