# 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 reemplaza un valor por otro en una cadena. 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"); //el contenido de cadena ahora esa "Tengo que hacer y hacer y hacer muchisimos ejercicios"  let cadena="Tengo que hacer y hacer y hacer muchos ejercicios"; cadena.replace("hacer","HACER"); //el contenido de cadena ahora esa "Tengo que HACER y hacer y hacer muchos ejercicios" |

**concat**() método que une dos o mas cadenas.

|  |
| --- |
| let cadena1="Tengo que hacer ";  let cadena2="y hacer "; let cadena3="muchos ejercicios"; cadena.concat(cadena1, cadena2, cadena3); //retorna "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 separarador 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 |

**Ejercicios**

1) Hacer la función sufijo que indica si una palabra es sufijo de otra palabra/texto. Por ejemplo:

sufijo("ola", "Hola mundo") -> falso

sufijo("ola", "que hermosa ola") -> verdadero

sufijo("ola", "que hermosa ola para surfear") -> falso

2) Hacer la función prefijo que indica si una palabra es prefijo de otra palabra/texto. Por ejemplo:

prefijo("cul", "espectaculo") -> falso

prefijo("cul", "culto") -> verdadero

3) Hacer la función esParte que indica si una palabra está contenida en la otra palabra/texto. Por ejemplo:

esParte("cul", "espectaculo") -> verdadero

esParte("cul", "culto") -> verdadero

esParte("cul", "algo que no") -> falso

4) Hacer una función que determine si una palabra es palíndromo (capicúa). Por ejemplo:

esPalindromo("anilina") -> verdadero

esPalindromo("roma") -> falso

esPalindromo("dabale arroz a la zorra el abad") -> verdadero recordar que las letras mayúsculas no son iguales a las minúsculas.

5) Escriba un algoritmo que genere un histograma horizontal de 10 valores ingresados por teclado, luego de leer todos los valores.

6) Escriba una funcion que permita formatear un texto con las siguientes caracteristicas:

-se debe especificar un tamaño.

-se debe especificar un caracter de relleno.

-se debe especificar la ubicacion del texto.

7) Escriba un algoritmo que liste los numeros pares entre 1 y 25, a su lado su cuadrado y a continuacion su cubo, preguntando al usuario como desea encolumnarlos.

8) Escriba un algoritmo que liste los numeros primos entre 1 y 100, preguntando al usuario como desea encolumnarlos.

9) Escriba un algoritmo que genere la tabla de multiplicacion de los numeros naturales de 1 cifra. Preguntando al usario como desea encolumnarlos.

10) Escriba un algoritmo que genere la tabla de potencias de los 5 primeros numeros enteros positivos. Preguntando al usuario como desea encolumnarlos.