

Comandos y métodos específicos de Strings

Declaración de strings:

- Las cadenas de caracteres o strings se pueden definir con comillas dobles o simples.
- Es importante ser consistente en el uso de comillas simples o dobles a lo largo de un proyecto.
- Para strings con entrecomillados internos, se pueden usar comillas simples dentro de dobles o viceversa, o escapar las comillas internas con una barra invertida.
- Las comillas invertidas (backticks) permiten crear strings que admiten saltos de línea y la introducción de variables directamente dentro del string mediante `${variable}`.
- Las comillas invertidas son útiles para crear plantillas HTML dinámicas

Métodos comunes de strings:

- **.length**: Devuelve la longitud de una cadena de texto.
- **.slice(inicio, fin)**: Extrae una sección de una cadena desde el índice inicio hasta el índice fin (sin incluirlo).
- **.substring(inicio, fin)**: Similar a slice(), extrae una sección de una cadena.
- **.substr(inicio, longitud)**: Extrae una sección de la cadena, comenzando en inicio y con una longitud determinada. **Este método está obsoleto.**
- **.replace(valorABuscar, valorDeReemplazo)**: Reemplaza la primera instancia de valorABuscar con valorDeReemplazo. Para reemplazar todas las instancias, se debe utilizar una expresión regular con el flag g (global).

Manipulación de cadenas de texto:

- **.toUpperCase()**: Convierte toda la cadena a mayúsculas.
- **.toLowerCase()**: Convierte toda la cadena a minúsculas.
- **.concat(str1, str2, ...)**: Concatena dos o más cadenas.
También se puede usar el operador + para concatenar, o las comillas invertidas.
- **.trim()**: Elimina los espacios en blanco al principio y al final de una cadena.
- **.trimStart()**: Elimina los espacios en blanco al principio de una cadena.
- **.trimEnd()**: Elimina los espacios en blanco al final de una cadena
- **.charAt(posición)**: Devuelve el carácter en la posición especificada.
También se puede acceder al carácter mediante la notación de corchetes (ej., `str[posición]`)
- **.indexOf(palabra)**: Devuelve la posición de la primera ocurrencia de una palabra dentro de la cadena.

Expresiones regulares:

- Las expresiones regulares permiten realizar búsquedas más avanzadas dentro de un texto, incluyendo la búsqueda de múltiples instancias y la búsqueda insensible a mayúsculas y minúsculas
- Para usar una expresión regular global (que reemplaza todas las instancias), se utiliza el flag g

Métodos de búsqueda de cadenas:

- **.match(expresiónRegular)**: Devuelve un array con todas las coincidencias encontradas según la expresión regular.
- **.includes(substring)**: Comprueba si una cadena contiene un substring y devuelve true o false.
- **.startsWith(substring)**: Comprueba si una cadena comienza con un substring y devuelve true o false.
- **.endsWith(substring)**: Comprueba si una cadena termina con un substring y devuelve true o false.