

# OBJETO STRING

Este objeto nos permite crear instancias que tengan una cadena de texto

## SINTAXIS DEL OBJETO String:

```
var nombre_objeto=new String();
```

Los Objetos de tipo String tienen la siguiente **propiedad**:

- **length**; Es un valor numérico que nos indica la longitud en caracteres que tiene la cadena dada.

Los objetos de tipo String tienen todos los siguientes **métodos**:

### BÚSQUEDA

- **charAt(indice)**: Devuelve el carácter situado en la posición especificada por índice.
- **indexOf(cadena\_buscada,indice)**: Devuelve la posición de la primera ocurrencia de cadena\_buscada dentro de la cadena actual, a partir de la posición indicada por el índice.
- **lastIndexOf(cadena\_buscada,indice)**: Devuelve la posición de la última ocurrencia de cadena\_buscada dentro de la cadena actual, a partir de la posición indicada por índice y buscando hacia atrás.
- **search (rgExp)**: Busca la expresión (cadena o rgExp) pasada como parámetro y si la encuentra devuelve la posición del primer carácter de la primera coincidencia.
- **match(rgExp)**: Compara una cadena con una expresión regular y devuelve un array que contiene los resultados de la búsqueda.

### INCLUYE/EMPIEZA/TERMINA

- **includes(cadena, indice)**: Comprueba si la cadena contiene los caracteres especificados. También se puede indicar a partir de qué posición buscar. Devuelve true o false. Es case sensitive.
- **startsWith(cadena)**
- **endsWith(cadena)**: Comprueba si la cadena termina por los caracteres especificados. Se puede indicar de forma opcional al longitud de la cadena. Devuelve true o false. Es case sensitive

### COMPARACIÓN

- **localeCompare(cadena)**: Sirve para comparar dos cadenas. Devuelve 0 si son iguales, -1 si la cadena desde la que se llama va antes (alfabéticamente) a la pasada como parámetro y 1 en caso contrario.

## EXTRACCIÓN

- **slice(índice\_inicial, [índice\_final]):** Devuelve una sección de una cadena.
- **substr(índice\_inicial, [longitud]):** Obtiene una cadena que comienza en la ubicación especificada y tiene la longitud establecida.
- **substring (primerIndice,segundoIndice):** Devuelve la cadena equivalente a la subcadena que comienza en la posición indicada por primerIndice y que finaliza en la posición anterior a la indicada por segundoIndice.
- **split(separador):** Parte la cadena en un array de caracteres, si el carácter separador no se encuentra, devuelve un array con un solo elemento que coincide con la cadena original.
- **trim():** Quita los espacios en blanco y de terminador de líneas iniciales y finales de una cadena.

## OTROS

- **concat(cadena1, cadena2,...):** Une dos o más cadenas.
- **repeat(numero):** Repite el contenido de la variable tantas veces como le pase por parámetro
- **toLowerCase():** Devuelve la cadena de caracteres en minúsculas
- **toUpperCase():** Devuelve la cadena de caracteres en mayúsculas.
- **valueOf():** Convierte el objeto a su tipo primitivo ( number, boolean, string, function).
- **replace (rgExp, replaceText):**Reemplaza el texto de una cadena, usando una expresión regular o una cadena de búsqueda.
- **toString():** Devuelve una cadena que representa al objeto string
- **padStart(tamaño,relleno):** Sirve para rellenar una cadena por el principio. Se indica el tamaño final y la cadena que va a servir de relleno. Si no se indica el relleno se utilizan espacios.
- **padEnd(tamaño,relleno):** Igual que el anterior pero rellena por el final.

EJEMPLO: Pedimos una contraseña de mínimo 4 letras, la pasamos a mayúsculas y mostramos la primera letra. Si la contraseña no llega a 4 letras le avisamos del mínimo de letras y volvemos a pedir.

```
<SCRIPT language=javascript>
function pasa() {
var contra=T.value;
var nletras=contra.length;
if (nletras<4)
{
    alert("Mínimo 4 letras");
    T.value="";
    nletras=0;
}
if (nletras!=0) {
    contra=contra.toUpperCase();
    alert("La primera es: "+contra.charAt(0)); }
}
</SCRIPT>
```

## Propiedad prototype

Devuelve una referencia al prototipo correspondiente a una clase de cadena. `string.prototype`. El argumento *string* es el nombre de una cadena.

Esta propiedad se utiliza para proporcionar un conjunto base de funcionalidad a una clase de objetos. Las nuevas instancias de un objeto "heredan" el comportamiento del prototipo asignado a ese objeto.

Por ejemplo, para agregar un método al objeto **String** que devuelve el valor del último elemento de la cadena, declara la función, agrégala a **String.prototype** y después úsala.

```
function string_last( ){
    return this.charAt(this.length - 1);
}
String.prototype.last = string_last;
var myString = new String("every good boy does fine");
document.write(myString.last());

// Output: e
```

Todos los objetos intrínsecos de JavaScript tienen una propiedad **prototype** que es de solo lectura. Se pueden agregar propiedades y métodos al prototipo, pero no se puede asignar al objeto otro prototipo diferente. No obstante, se puede asignar un nuevo prototipo a los objetos definidos por el usuario.