

## ANEXO MODULO 3 – OBJETOS ESTANDAR

### CONSTRUCTORES PARA LOS TIPOS DE DATOS

**Number(valor):** Este constructor crea objetos para almacenar valores numéricos. Acepta números y también cadenas de caracteres con números. Si el valor especificado por el atributo no se puede convertir en un número, el constructor devuelve el valor NaN (No es un Número).

**String(valor):** Este constructor crea objetos para almacenar cadenas de caracteres. Acepta una cadena de caracteres o cualquier valor que pueda convertirse en una cadena de caracteres, incluidos números.

**Boolean(valor):** Este constructor crea objetos para almacenar valores booleanos. Acepta los valores true y false. Si el valor se omite o es igual a 0, NaN, null, undefined, o una cadena de caracteres vacía, el valor almacenado en el objeto es false, en caso contrario es true.

**Array(valores):** Este constructor crea objetos para almacenar arrays. Si se proveen múltiples valores, el constructor crea un array con esos valores, pero si solo se provee un valor, y ese valor es un número entero, el constructor crea un array con la cantidad de elementos que indica el valor del atributo y almacena el valor undefined en cada índice.

### OBJETO STRING - PROPIEDADES Y METODOS

**length:** Esta propiedad devuelve un entero que representa la cantidad de caracteres en la cadena.

**toLowerCase():** Este método convierte los caracteres a letras minúsculas.

**toUpperCase():** Este método convierte los caracteres a letras mayúsculas.

**trim():** Este método elimina espacios en blanco a ambos lados de la cadena de caracteres.

**substr(comienzo, extensión):** Este método devuelve una nueva cadena de caracteres con caracteres extraídos de la cadena original. El atributo comienzo indica la posición del primer carácter a leer, y el atributo extensión determina cuántos caracteres queremos incluir. Si no se especifica la extensión, el método devuelve todos los caracteres hasta el final de la cadena.

**substring(comienzo, final):** Este método devuelve una nueva cadena de caracteres con caracteres extraídos de la cadena original. Los atributos comienzo y final son números enteros que determinan las posiciones del primer y último carácter a incluir.

**split(separador, limite):** Este método divide la cadena de caracteres y devuelve un array con todas las partes. El atributo separador indica el carácter en el que se va a cortar la cadena y el atributo limite es un número entero que determina el número máximo de partes. Si no especificamos un límite, la cadena se cortará cada vez que se encuentre el separador.

**startsWith(valor):** Este método devuelve true si el texto especificado por el atributo valor se encuentra al comienzo de la cadena de caracteres.

**endsWith(valor):** Este método devuelve true si el texto especificado por el atributo valor se encuentra al final de la cadena de caracteres.

**includes**(buscar, posición): Este método busca el valor del atributo buscar dentro de la cadena y devuelve true o false de acuerdo con el resultado. El atributo buscar es el texto que queremos buscar, y el atributo posición determina el índice en el que queremos comenzar la búsqueda. Si el atributo posición no se especifica, la búsqueda comienza desde el inicio de la cadena.

**indexOf**(valor, posición): Este método devuelve el índice en el que el texto especificado por el atributo valor se encuentra por primera vez. El atributo posición determina el índice en el que queremos comenzar la búsqueda. Si el atributo posición no se especifica, la búsqueda comienza desde el inicio de la cadena. El método devuelve el valor -1 si ninguna coincidencia es encontrada.

**lastIndexOf**(valor, posición): Este método devuelve el índice en el que se encuentra por primera vez el texto especificado por el atributo valor. A diferencia de **indexOf**(), este método realiza una búsqueda hacia atrás, desde el final de la cadena. El atributo posición determina el índice en el que queremos comenzar la búsqueda. Si no se especifica el atributo posición, la búsqueda comienza desde el final de la cadena. El método devuelve el valor -1 si no se encuentra ninguna coincidencia.

**replace**(expresión, reemplazo): Este método reemplaza la parte de la cadena de caracteres que coincide con el valor del atributo expresión por el texto especificado por el atributo reemplazo. El atributo expresión se puede especificar como una cadena de caracteres o como una expresión regular para buscar un texto con un formato particular.

## OBJETOS ARRAY - PROPIEDADES Y METODOS

**length**: Esta propiedad devuelve un número entero que representa la cantidad de valores en el array.

**push**(valores): Este método agrega uno o más valores al final del array y devuelve la nueva extensión del array. También contamos con un método similar llamado **unshift**(), que agrega los valores al comienzo del array.

**pop**(): Este método elimina el último valor del array y lo devuelve. También contamos con un método similar llamado **shift**(), que elimina el primer valor del array.

**concat**(array): Este método concatena el array con el array especificado por el atributo y devuelve un nuevo array con el resultado. Los arrays originales no se modifican.

**splice**(índice, remover, valores): Este método agrega o elimina valores de un array y devuelve un nuevo array con los elementos eliminados. El atributo índice indica el índice en el que vamos a introducir las modificaciones, el atributo remover es el número de valores que queremos eliminar desde el índice, y el atributo valores es la lista de valores separados por coma que queremos agregar al array desde el índice. Si solo queremos agregar valores, el atributo remover se puede declarar con el valor 0, y si solo queremos eliminar valores, podemos ignorar el último atributo.

**slice**(comienzo, final): Este método copia los valores en las posiciones indicadas por los atributos dentro de un nuevo array. Los atributos comienzo y final indican los índices del primer y último valor a copiar. El último valor no se incluye en el nuevo array.

**indexOf**(valor, posición): Este método devuelve el índice en el que se encuentra por primera vez el valor especificado por el atributo valor. El atributo posición determina el índice en el que queremos comenzar la búsqueda. Si el atributo posición no se especifica, la búsqueda comienza desde el inicio del array. El método devuelve el valor -1 si no se encuentra ninguna coincidencia.

**lastIndexOf(valor, posición):** Este método devuelve el índice en el que el valor especificado por el atributo valor se encuentra por primera vez. A diferencia de **indexOf()**, este método realiza una búsqueda de atrás hacia adelante. El atributo posición determina el índice en el que queremos comenzar la búsqueda. Si no se especifica el atributo posición, la búsqueda comienza desde el final del array. El método devuelve el valor -1 si no se encuentra ninguna coincidencia.

**filter(función):** Este método envía los valores del array a una función uno por uno y devuelve un nuevo array con todos los valores que aprueba la función. El atributo función es una referencia a una función o una función anónima a cargo de validar los valores. La función recibe tres valores: el valor a evaluar, su índice y una referencia al array. Después de procesar el valor, la función debe devolver un valor booleano para indicar si es válido o no.

**every(función):** Este método envía los valores del array a una función uno por uno y devuelve true cuando la función aprueba todos los valores. El atributo función es una referencia a una función o una función anónima a cargo de evaluar los valores. La función recibe tres valores: el valor a evaluar, su índice, y una referencia al array. Después de procesar el valor, la función debe devolver un valor booleano indicando si es válido o no. También contamos con un método similar llamado **some()** que devuelve true si la función aprueba al menos uno de los valores.

**join(separador):** Este método crea una cadena de caracteres con todos los valores en el array. El atributo separador especifica el carácter o la cadena de caracteres que queremos incluir entre los valores.

**reverse():** Este método invierte el orden de los valores en el array.

**sort(función):** Este método ordena los valores en el array. El atributo función es una referencia a una función o una función anónima a cargo de comparar los valores. La función recibe dos valores desde el array y debe devolver un valor booleano indicando su orden. Si no se especifica el atributo, el método ordena los elementos alfabéticamente y en orden ascendente.

**map(función):** Este método envía los valores del array a una función uno por uno y crea un nuevo array con los valores que devuelve la función. El atributo función es una referencia a una función o una función anónima a cargo de procesar los valores. La función recibe tres valores: el valor a procesar, su índice y una referencia al array.

## OBJETOS DATE - PROPIEDADES Y METODOS

**Date(valor):** Este constructor crea un valor en milisegundos para representar una fecha basada en los valores provistos por el atributo. El atributo valor se puede declarar como una cadena de caracteres o como los componentes de una fecha separados por comas, en este orden: año, mes, día, horas, minutos, segundos y milisegundos. Si no especificamos ningún valor, el constructor crea un objeto con la fecha actual del sistema.

**getFullYear():** Este método devuelve un número entero que representa el año (un valor de 4 dígitos).

**getMonth():** Este método devuelve un número entero que representa el mes (un valor de 0 a 11).

**getDate():** Este método devuelve un número entero que representa el día del mes (un valor de 1 a 31).

**getDay():** Este método devuelve un número entero que representa el día de la semana (un valor de 0 a 6).

**getHours():** Este método devuelve un número que representa las horas (un valor de 0 a 23).

**getMinutes():** Este método devuelve un número entero que representa los minutos (un valor de 0 a 59).

**getSeconds():** Este método devuelve un número entero que representa los segundos (un valor de 0 a 59).

**getMilliseconds():** Este método devuelve un número entero que representa los milisegundos (un valor de 0 a 999).

**getTime():** Este método devuelve un número entero en milisegundos que representa el intervalo desde el 1 de enero de 1970 hasta la fecha.

**setFullYear(año):** Este método especifica el año (un valor de 4 dígitos). También puede recibir valores para especificar el mes y el día.

**setMonth(mes):** Este método especifica el mes (un valor de 0 a 11). También puede recibir valores para especificar el día.

**setDate(día):** Este método especifica el día (un valor de 1 a 31).

**setHours(horas):** Este método especifica la hora (un valor de 0 a 23). También puede recibir valores para especificar los minutos y segundos.

**setMinutes(minutos):** Este método especifica los minutos (un valor de 0 a 59). También puede recibir un valor para especificar los segundos.

**setSeconds(segundos):** Este método especifica los segundos (un valor de 0 a 59). También puede recibir un valor para especificar los milisegundos.

**setMilliseconds(milisegundos):** Este método especifica los milisegundos (un valor de 0 a 999)

**toString():** Este método convierte una fecha en una cadena de caracteres. El valor que devuelve se expresa en inglés americano y con el formato "Wed Jan 04 2017 22:32:48 GMT-0500 (EST)".

**toDateString():** Este método convierte una fecha en una cadena de caracteres, pero solo devuelve la parte de la fecha, no la hora. El valor se expresa en inglés americano y con el formato "Wed Jan 04 2017".

**toTimeString():** Este método convierte una fecha en una cadena de caracteres, pero solo devuelve la hora. El valor se expresa en inglés americano y con el formato "23:21:55 GMT-0500 (EST)".

## OBJETO MATH - PROPIEDADES Y METODOS

**PI:** Esta propiedad devuelve el valor de PI.

**E:** Esta propiedad devuelve la constante Euler.

**LN10:** Esta propiedad devuelve el logaritmo natural de 10. El objeto también incluye las propiedades **LN2** (algoritmo natural de 2), **LOG2E** (algoritmo base 2 de E) y **LOG10E** (algoritmo base 10 de E).

**SQRT2:** Esta propiedad devuelve la raíz cuadrada de 2. El objeto también incluye la propiedad **SQRT1\_2** (la raíz cuadrada de 1/2).

**ceil(valor):** Este método redondea un valor hacia arriba al siguiente entero y devuelve el resultado.

**floor(valor):** Este método redondea un valor hacia abajo al siguiente entero y devuelve el resultado.

**round(valor):** Este método redondea un valor al entero más cercano y devuelve el resultado.

**trunc(valor):** Este método elimina los dígitos de la fracción y devuelve un entero.

**abs(valor):** Este método devuelve el valor absoluto de un número (invierte valores negativos para obtener un número positivo).

**min(valor):** Este método devuelve el número más pequeño de una lista de valores separados por comas.

**max(valores):** Este método devuelve el número más grande de una lista de valores separados por comas.

**random():** Este método devuelve un número al azar en un rango entre 0 y 1.

**pow(base, exponente):** Este método devuelve el resultado de elevar la base a la potencia del exponente.

**exp(exponente):** Este método devuelve el resultado de elevar E a la potencia del exponente. El objeto también incluye el método **expm1()**, que devuelve el mismo resultado menos 1.

**sqrt(valor):** Este método devuelve la raíz cuadrada de un valor.

**log10(valor):** Este método devuelve el logaritmo base 10 de un valor. El objeto también incluye los métodos **log()** (devuelve el logaritmo base E), **log2()** (devuelve el logaritmo base 2) y **log1p()** (devuelve el logaritmo base E de 1 más un número).

**sin(valor):** Este método devuelve el seno de un número. El objeto también incluye los métodos **asin()** (devuelve el arcoseno de un número), **sinh()** (devuelve el seno hiperbólico de un número) y **asinh()** (devuelve el arcoseno hiperbólico de un número).

**cos(valor):** Este método devuelve el coseno de un número. El objeto también incluye los métodos **acos()** (devuelve el arcocoseno de un número), **cosh()** (devuelve el coseno hiperbólico de un número), y **acosh()** (devuelve el arcocoseno hiperbólico de un número).

**tan(valor):** Este método devuelve la tangente de un número. El objeto también incluye los métodos **atan()** (devuelve la arcotangente de un número), **atan2()** (devuelve la arcotangente o el cociente de dos números), **tanh()** (devuelve la tangente hiperbólica de un número), y **atanh()** (devuelve la arcotangente hiperbólica de un número).

## OBJETO WINDOW - PROPIEDADES Y METODOS

**location:** Esta propiedad contiene un objeto Location con información acerca del origen del documento. También se puede usar como una propiedad para declarar o devolver la URL del documento (por ejemplo, `window.location = "https://www.azulschool.net/`).

**history:** Esta propiedad contiene un objeto History con propiedades y métodos para manipular el historial de navegación.

**navigator:** Esta propiedad contiene un objeto Navigator con información acerca de la aplicación y el dispositivo.

**document:** Esta propiedad contiene un objeto Document, que provee acceso a los objetos que representan los elementos HTML en el documento.

**innerWidth**—Esta propiedad devuelve el ancho de la ventana en píxeles.

**innerHeight**—Esta propiedad devuelve la altura de la ventana en píxeles.

**scrollX**—Esta propiedad devuelve el número de píxeles en los que el documento se ha desplazado horizontalmente.

**scrollY**—Esta propiedad devuelve el número de píxeles en los que el documento se ha desplazado verticalmente.

**alert(valor)**—Este método muestra una ventana emergente en la pantalla que muestra el valor entre paréntesis.

**confirm(mensaje)**—Este método es similar a **alert()**, pero ofrece dos botones, Aceptar y Cancelar, para que el usuario elija qué hacer. El método devuelve true o false, según a la respuesta del usuario.

**prompt(mensaje)**—Este método muestra una ventana emergente con un campo de entrada para permitir al usuario introducir un valor. El método devuelve el valor que inserta el usuario.

**setTimeout(función, milisegundos)**—Este método ejecuta la función especificada en el primer atributo cuando haya pasado el tiempo especificado por el segundo atributo. El objeto Window también ofrece el método **clearTimeout()** para cancelar este proceso.

**setInterval(función, milisegundos)**—Este método es similar a **setTimeout()**, pero llama a la función constantemente. El objeto Window también ofrece el método **clearInterval()** para cancelar el proceso.

**open(URL, ventana, parámetros)**—Este método abre un documento en una nueva ventana. El atributo URL es la URL del documento que queremos abrir, el atributo ventana es el nombre de la ventana donde queremos mostrar el documento (si el nombre no se especifica o la ventana no existe, el documento se abre en una nueva ventana), y el atributo parámetros es una lista de parámetros de configuración separados por comas que configuran las características de la ventana (por ejemplo, “resizable=no,scrollbars=no”). El objeto Window también ofrece el método **close()** para cerrar una ventana abierta con este método.

## OBJETO DOCUMENT - PROPIEDADES Y METODOS

**forms:** Esta propiedad devuelve un array con referencias a todos los objetos Element que representan los elementos <form> en el documento.

**images:** Esta propiedad devuelve un array con referencias a todos los objetos Element que representan los elementos <img> en el documento.

**links:** Esta propiedad devuelve un array con referencias a todos los objetos Element que representan los elementos <a> en el documento.

**getElementById(id):** Este método devuelve una referencia al objeto Element que representa el elemento identificado con el valor especificado por el atributo (el valor asignado al atributo id).

**getElementsByClassName**(clase): Este método devuelve un array con referencias a los objetos Element que representan los elementos identificados con la clase especificada por el atributo (el valor asignado al atributo clase).

**getElementsByName**(nombre): Este método devuelve un array con referencias a los objetos Element que representan los elementos identificados con el nombre especificado por el atributo (el valor asignado al atributo name)

**getElementsByTagName**(tipo): Este método devuelve un array con referencias a los objetos Element que representan el tipo de elementos especificados por el atributo. El atributo es el nombre que identifica a cada tipo de elemento, como h1, p, img, div, etc.

**querySelector**(selector): Este método devuelve una referencia al objeto Element que representa el elemento que coincide con el selector especificado por el atributo. El método devuelve el primer elemento en el documento que coincide con el selector CSS.

**querySelectorAll**(selectores): Este método devuelve un array con referencias a los objetos Element que representan los elementos que coinciden con los selectores especificados por el atributo. Se pueden declarar uno o más selectores separados por comas.

#### OBJETOS ELEMENT - PROPIEDADES Y METODOS

**clientWidth**: Esta propiedad devuelve el ancho del elemento, incluido el relleno.

**clientHeight**: Esta propiedad devuelve la altura del elemento, incluido el relleno.

**offsetTop**: Esta propiedad devuelve el número de píxeles que se ha desplazado el elemento desde la parte superior de su contenedor.

**offsetLeft**: Esta propiedad devuelve el número de píxeles que se ha desplazado el elemento desde el lado izquierdo de su contenedor.

**offsetWidth**: Esta propiedad devuelve el ancho del elemento, incluidos el relleno y el borde.

**offsetHeight**: Esta propiedad devuelve la altura del elemento, incluidos el relleno y el borde.

**scrollTop**: Esta propiedad devuelve el número de píxeles en los que el contenido del elemento se ha desplazado hacia arriba.

**scrollLeft**: Esta propiedad devuelve el número de píxeles en los que el contenido del elemento se ha desplazado hacia la izquierda.

**scrollWidth**: Esta propiedad devuelve el ancho del contenido del elemento.

**scrollHeight**: Esta propiedad devuelve la altura del contenido del elemento.

**className**: Esta propiedad declara o devuelve el valor del atributo class.

**classList**: Esta propiedad devuelve un array con la lista de las clases asignadas al elemento.

#### METODOS DEL OBJETO CLASSLIST

**add**(clase): Este método agrega una clase al elemento.

**remove**(class): Este método agrega una clase del elemento.

**toggle**(clase): Este método agrega o elimina una clase dependiendo del estado actual. Si la clase ya se ha asignado al elemento, la elimina, y en caso contrario la agrega.

**contains**(clase): Este método detecta si se ha asignado la clase al elemento o no, y devuelve true o false respectivamente.

**innerHTML**: Esta propiedad declara o devuelve el contenido de un elemento.

**outerHTML**: Esta propiedad declara o devuelve un elemento y su contenido. A diferencia de la propiedad **innerHTML**, esta propiedad no solo reemplaza el contenido, sino también el elemento.

**insertAdjacentHTML**(ubicación, contenido): Este método inserta contenido en una ubicación determinada por el atributo ubicación. Los valores disponibles son **beforebegin** (antes del elemento), **afterbegin** (dentro del elemento, antes del primer elemento hijo), **beforeend** (dentro del elemento, después del último elemento hijo) y **afterend** (después del elemento)

**style**: Contiene un objeto llamado Styles que a su vez incluye propiedades para modificar los estilos de los elementos.

## ATRIBUTOS DEL OBJETO STYLE

**color**: Esta propiedad declara el color del contenido del elemento.

**background**: Esta propiedad declara los estilos del fondo del elemento. También podemos trabajar con cada estilo de forma independiente usando las propiedades asociadas **backgroundColor**, **backgroundImage**, **backgroundRepeat**, **backgroundPosition** y **backgroundAttachment**.

**border**: Esta propiedad declara los estilos del borde del elemento. Podemos modificar cada estilo de forma independiente con las propiedades asociadas **borderColor**, **borderStyle**, y **borderWidth**, o modificar cada borde individualmente usando las propiedades asociadas **borderTop** (**borderTopColor**, **borderTopStyle**, y **borderTopWidth**), **borderBottom** (**borderBottomColor**, **borderBottomStyle**, y **borderBottomWidth**), **borderLeft** (**borderLeftColor**, **borderLeftStyle**, y **borderLeftWidth**), y **borderRight** (**borderRightColor**, **borderRightStyle**, y **borderRightWidth**).

**margin**: Esta propiedad declara el margen del elemento. También podemos usar las propiedades asociadas **marginBottom**, **marginLeft**, **marginRight**, y **marginTop**.

**padding**: Esta propiedad declara el relleno del elemento. También podemos usar las propiedades asociadas **paddingBottom**, **paddingLeft**, **paddingRight**, y **paddingTop**.

**width**: Esta propiedad declara el ancho del elemento. Existen dos propiedades asociadas para declarar el ancho máximo y mínimo de un elemento: **maxWidth** y **minWidth**.

**height**: Esta propiedad declara la altura del elemento. Existen dos propiedades asociadas para declarar la altura máxima y mínima de un elemento: **maxHeight** y **minHeight**.



**visibility:** Esta propiedad determina si el elemento es visible o no.

**display:** Esta propiedad define el tipo de caja usado para presentar el elemento.

**position:** Esta propiedad define el tipo de posicionamiento usado para posicionar el elemento.

**top:** Esta propiedad especifica la distancia entre el margen superior del elemento y el margen superior de su contenedor.

**bottom:** Esta propiedad especifica la distancia entre el margen inferior del elemento y el margen inferior de su contenedor.

**left:** Esta propiedad especifica la distancia entre el margen izquierdo del elemento y el margen izquierdo de su contenedor.

**right:** Esta propiedad especifica la distancia entre el margen derecho del elemento y el margen derecho de su contenedor.

**cssFloat:** Esta propiedad permite al elemento flotar hacia un lado o el otro.

**clear:** Esta propiedad recupera el flujo normal del documento, impidiendo que el elemento siga flotando hacia la izquierda, la derecha, o ambos lados.

**overflow:** Esta propiedad especifica cómo se va a mostrar el contenido que excede los límites de la caja de su contenedor.

**zIndex:** Esta propiedad define un índice que determina la posición del elemento en el eje z.

**font:** Esta propiedad declara los estilos de la fuente. También podemos declarar los estilos individuales usando las propiedades asociadas **fontFamily**, **fontSize**, **fontStyle**, **fontVariant** y **fontWeight**.

**textAlign:** Esta propiedad alinea el texto dentro del elemento.

**verticalAlign:** Esta propiedad alinea elementos **Inline** verticalmente.

**textDecoration:** Esta propiedad resalta el texto con una línea. También podemos declarar los estilos de forma individual asignando los valores true o false a las propiedades **textDecorationBlink**, **textDecorationLineThrough**, **textDecorationNone**, **textDecorationOverline**, y **textDecorationUnderline**.

## METODOS PARA CREAR UN OBJETO ELEMENT

**createElement(nombre):** Este método crea un nuevo objeto Element del tipo especificado por el atributo nombre.

**appendChild(elemento):** Este método inserta el elemento representado por un objeto Element como hijo de un elemento existente en el documento.

**removeChild(elemento):** Este método elimina un elemento hijo de otro elemento. El atributo debe ser una referencia del hijo a eliminarse.