

3.- UTILIZACIÓN DE LOS OBJETOS PREDEFINIDOS DE JAVASCRIPT.

OBJETIVOS

- Conocer cuáles son los principales objetos predefinidos de JavaScript.
- Comprender las propiedades y métodos de los principales objetos de JavaScript.
- Aprender a generar código HTML desde sentencias JavaScript.
- Manipular y gestionar la creación y apariencia de las ventanas del navegador, además de la comunicación entre ellas.

2.1.- Introducción

Un objeto es una colección de:

- Propiedades (valores).
- Métodos (funciones).
- Eventos (acciones)

Para acceder a una propiedad o método de un objeto hay que utilizar el punto
(.)

```
Objeto.propiedad;  
Objeto.método([argumentos]);
```

2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

En la jerarquía de objetos del navegador, el objeto principal es **window**, que representa la ventana abierta en el navegador. Dentro de esta ventana se encuentran diversos objetos importantes:



2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

Acceso a propiedad y métodos

```
window.nombrePropiedad  
window.nombreMétodo( [parámetros] )
```

Un objeto **window** también se podrá referenciar mediante la palabra **self**

```
self.nombrePropiedad  
self.nombreMétodo( [parámetros] )
```

Como contiene el resto de objeto podemos omitir su nombre:

```
nombrePropiedad  
nombreMétodo( [parámetros] )
```

2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

Propiedades:

Propiedades del objeto Window	
Propiedad	Descripción
<u>closed</u>	Devuelve un valor <code>Boolean</code> indicando cuando una ventana ha sido cerrada o no.
<u>console</u>	Retorna una referencia a un objeto <code>Console</code> , que tiene métodos para mostrar información al usuario a través de la consola del ordenador. (Ej: <code>console.log("Hola")</code>)
<u>document</u>	Devuelve el objeto <code>document</code> de la ventana.
<u>frameElement</u>	Devuelve el elemento <code>iframe</code> en el que está insertado el documento, null si es la ventana principal.
<u>frames</u>	Devuelve un array con todos <code>iframes</code> de la ventana actual.
<u>history</u>	Devuelve el objeto <code>history</code> de la ventana.
<u>innerHeight</u>	Altura útil de la ventana. No incluye barras de herramientas, estado, desplazamiento.
<u>innerWidth</u>	Anchura útil de la ventana
<u>length</u>	Devuelve el nº de <code>iframes</code> que hay en una ventana. (También <code>frames.length</code>)
<u>localStorage</u>	Permite almacenar datos localmente de forma permanente. (<i>Se verá en otra unidad</i>)
<u>location</u>	Devuelve la URL de la barra de direcciones de la ventana.

2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

Propiedades:

Propiedades del objeto Window	
Propiedad	Descripción
<u>name</u>	Ajusta o devuelve el nombre de una ventana.
<u>navigator</u>	Devuelve el objeto navigator de una ventana.
<u>opener</u>	Devuelve el objeto window que abrió la ventana actual.
<u>outerHeight</u>	Altura total de la ventana.
<u>outerWidth</u>	Anchura total de la ventana.
<u>pageXOffset</u>	Devuelve los píxeles que está desplazado horizontalmente el documento actual desde la esquina superior izquierda.
<u>pageYOffset</u>	Devuelve los píxeles que está desplazado verticalmente el documento actual desde la esquina superior izquierda.
<u>parent</u>	Devuelve el objeto window padre de la ventana actual.
<u>screen</u>	Devuelve el objeto screen de la ventana.
<u>screenLeft</u>	Muestra el nº de píxeles desde el borde izquierdo de la pantalla al borde izquierdo de la ventana. (Excepto Firefox)
<u>screenTop</u>	Muestra el nº de píxeles desde el borde superior de la pantalla al borde superior de la ventana. (Excepto Firefox)

2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

Propiedades:

Propiedades del objeto Window	
Propiedad	Descripción
<u>screenY</u>	Muestra el nº de píxeles desde el borde superior de la pantalla al borde superior de la ventana. (Firefox)
<u>sessionStorage</u>	Almacena datos localmente durante la sesión. (<i>Se verá en otra unidad</i>)
<u>scrollX</u>	Alias de <u>pageXOffset</u> .
<u>scrollY</u>	Alias de <u>pageYOffset</u> .
<u>self</u>	Devuelve la ventana actual.
<u>status</u>	Obtiene o establece el texto mostrado en la barra de estado.
<u>top</u>	Devuelve el objeto <u>window</u> de nivel superior.

2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

Métodos:

Métodos del objeto Window	
Método	Descripción
<code>alert(mens)</code>	Muestra una ventana emergente de alerta con el mensaje y un botón de aceptar.
<code>atob(cadena)</code>	Decodifica una cadena codificada con <code>btoa(cadena)</code> ;
<code>blur()</code>	Elimina el foco de la ventana actual.
<code>btoa(cadena)</code>	Codifica una cadena en base-64 .
<code>clearInterval(id)</code>	Resetea el cronómetro ajustado con <code>setInterval()</code> identificado por <code>id</code> .
<code>clearTimeout(id)</code>	Resetea el cronómetro ajustado con <code>setTimeout()</code> identificado por <code>id</code> .
<code>close()</code>	Cierra la ventana actual.
<code>confirm(mens)</code>	Muestra una ventana emergente con un mensaje, un botón de aceptar y un botón de cancelar.
<code>focus()</code>	Coloca el foco en la ventana actual.
<code>getComputedStyle()</code>	Obtiene las propiedades y valores CSS actuales de un elemento concreto.

2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

Métodos:

Métodos del objeto Window	
Método	Descripción
<code>isNaN(n)</code>	Verdad si <code>n</code> es un NaN (se verá en el objeto Number).
<code>matchMedia(mQS)</code>	Obtiene un objeto MediaQueryList que representa el resultado del Media Query Search especificado como argumento.
<code>moveBy(tn, tm)</code>	Mueve la ventana que tiene el foco <code>n</code> píxeles horizontalmente y <code>m</code> píxeles verticalmente.
<code>moveTo(x, y)</code>	Mueve la ventana que tiene el foco a la coordenada <code>(x, y)</code> .
<code>open(url, nom, car)</code>	Abre una nueva ventana de navegación.
<code>print()</code>	Imprime la página.
<code>prompt(mens, val)</code>	Muestra una ventana de diálogo para introducir datos con un mensaje <code>mens</code> y un valor <code>val</code> por defecto.
<code>resizeBy(tn, tm)</code>	Modifica el tamaño de la ventana en <code>n</code> píxeles de ancho y <code>m</code> píxeles de alto.
<code>resizeTo(n, m)</code>	Establece el tamaño de la ventana en <code>n</code> píxeles de ancho y <code>m</code> píxeles de alto.
<code>scrollBy(tx, ty)</code>	Mueve el contenido en la ventana que tiene el foco <code>n</code> píxeles horizontalmente y <code>m</code> píxeles verticalmente.
<code>scrollTo(x, y)</code>	Desplaza el contenido de la ventana a las coordenadas <code>(x, y)</code> de la misma.
<code>setInterval(ex, t)</code>	Evaluá una expresión <code>ex</code> después de <code>t</code> milisegundos. Devuelve un identificador.
<code>setTimeout(ex, t)</code>	Evaluá una expresión <code>ex</code> después de <code>t</code> milisegundos. Devuelve un identificador.
<code>stop()</code>	Realiza la función de parada de la ventana del navegador.

2.2.- Objetos de más alto nivel.

2.2.1.- Objeto Window

Crear ventanas:

Un script no creará nunca la ventana principal de un navegador pero un script que se está ejecutando en una ventana principal, sí que podrá crear o abrir nuevas sub-ventanas mediante el método open().

```
var subVentana=window.open("nueva.html","nueva","height=800,width=600");
```

La variable **subVentana**, hace referencia a la nueva ventana. Para cerrar esa ventana:

✓ **subVentana.close()**

2.2.- Objetos de más alto nivel.

2.2.2.- Objeto Location



El objeto **location** contiene información referente a la URL actual.

Propiedad	Descripción
hash	Cadena que contiene el nombre del enlace, dentro de la URL.
host	Cadena que contiene el nombre del servidor y el número del puerto, dentro de la URL.
hostname	Cadena que contiene el nombre de dominio del servidor (o la dirección IP), dentro de la URL.
href	Cadena que contiene la URL completa.
pathname	Cadena que contiene el camino al recurso, dentro de la URL.
port	Cadena que contiene el número de puerto del servidor, dentro de la URL.
protocol	Cadena que contiene el protocolo utilizado (incluyendo los dos puntos), dentro de la URL.
search	Cadena que contiene la información pasada en una llamada a un script, dentro de la URL.

2.2.- Objetos de más alto nivel.

2.2.2.- Objeto Location

Métodos del objeto Location

<code>assign()</code>	Carga un nuevo documento.
<code>reload()</code>	Vuelve a cargar la URL especificada en la propiedad <code>href</code> del objeto <code>location</code> .
<code>replace()</code>	Reemplaza el historial actual mientras carga la URL especificada en cadenaURL.

2.2.- Objetos de más alto nivel.

2.2.2.- Objeto Navigator

- ✓ El objeto **Navigator** contiene información sobre el navegador.

Propiedad	Descripción
appCodeName	Cadena que contiene el nombre en código del navegador.
appName	Cadena que contiene el nombre del cliente.
appVersion	Cadena que contiene información sobre la versión del cliente.
cookieEnabled	Determina si las cookies están o no habilitadas en el navegador.
language	Devuelve el idioma por defecto del navegador. "es" si es español o "en" si es inglés.
onLine	Indica si el navegador está offline u online. O sea que se ha desconectado de Internet usando la opción del navegador.
platform	Cadena con la plataforma sobre la que se está ejecutando el programa cliente.
userAgent	Cadena que contiene la cabecera completa del agente enviado en una petición HTTP . Contiene la información de las propiedades appCodeName y appVersion.

2.2.- Objetos de más alto nivel.

2.2.2.- Objeto Navigator

Métodos del objeto Navigator

Método	Descripción
<code>javaEnabled()</code>	Devuelve true si el cliente permite la utilización de Java, en caso contrario, devuelve false.

2.2.- Objetos de más alto nivel.

2.2.3.- Objeto screen

- ✓ El objeto **screen** representa una pantalla, usualmente la misma donde la ventana actual está siendo visualizada

Propiedad	Descripción
width	Devuelve el ancho de la pantalla en píxeles
height	Devuelve la altura total de la pantalla en píxeles
availWidth	Devuelve el ancho disponible para la ventana del navegador, excluyendo la barra de tareas
availHeight	Devuelve la altura disponible para la ventana del navegador, excluyendo la barra de tareas
colorDepth	Devuelve el número de bits utilizados para representar el color en un solo píxel en la pantalla
orientation	Información sobre la orientación de la pantalla

2.2.- Objetos de más alto nivel.

2.2.3.- Objeto screen

Métodos del objeto screen

Métodos	Descripción
<code>lockOrientation()</code>	Bloquea la orientación de la pantalla
<code>unlockOrientation()</code>	Desbloquea la orientación de la pantalla

2.2.- Objetos de más alto nivel.

2.2.4.- Objeto document

- ✓ El objeto **document** proporciona a los scripts, el acceso a todos los elementos HTML.
- ✓ Cada documento cargado en una ventana del navegador, será un objeto de tipo **document**.

Colecciones del objeto Document

Colección	Descripción
anchors []	Es un array que contiene todos los hiperenlaces del documento.
forms []	Es un array que contiene todos los formularios del documento.
images []	Es un array que contiene todas las imágenes del documento.
links []	Es un array que contiene todos los enlaces del documento.

2.2.- Objetos de más alto nivel.

2.2.4.- Objeto document

Propiedades del objeto Document	
Propiedad	Descripción
<code>activeElement</code>	Devuelve el elemento del DOM que tiene el foco.
<code>baseURI</code>	Devuelve el valor de la URI. Sólo IE.
<code>body</code>	Devuelve el <code>body</code> de un documento.
<code>characterSet</code>	Devuelve el conjunto de caracteres en el que está codificado el documento.
<code>cookie</code>	Devuelve todos los nombres/valores de las cookies asociadas al documento.
<code>doctype</code>	Devuelve la declaración <code>DOCTYPE</code> del documento.
<code>documentElement</code>	Devuelve el elemento <code><html></code> del documento.
<code>documentMode</code>	Devuelve el valor usado por el navegador para renderizar el documento. Sólo IE. (5, 7, 8, 9, 10, 11). Si no se especifica <code>DOCTYPE</code> IE devuelve 5.
<code>documentURI</code>	Establece o devuelve el valor de la URI. Todos excepto IE.
<code>domain</code>	Devuelve el nombre del dominio del servidor que llamó al documento.

2.2.- Objetos de más alto nivel.

2.2.4.- Objeto document

Propiedad	Descripción
<code>head</code>	Devuelve el elemento <code>head</code> del documento.
<code>inputEncoding</code>	Lo mismo que <code>characterSet</code> .
<code>lastModified</code>	Devuelve la fecha y la hora en la que el documento fue modificado.
<code>readyState</code>	Devuelve el estado de carga de un documento: <ul style="list-style-type: none">• <code>uninitialized</code> – No comenzó la descarga• <code>loading</code> – Se está descargando• <code>loaded</code> – Ha sido descargado• <code>interactive</code> – Descargado lo suficiente para que el usuario interactúe• <code>complete</code> – Totalmente descargado
<code>referrer</code>	Devuelve la URL del documento que descargó el documento actual.
<code>title</code>	Devuelve o establece el título del documento.
<code>URL</code>	Devuelve la URL completa del documento.

2.2.- Objetos de más alto nivel.

2.2.4.- Objeto document

Métodos del objeto Document	
Método	Descripción
<code>addEventListener()</code>	Añade un gestor de eventos al documento.
<code>adoptNode()</code>	Adopta un nodo de otro documento. (Y todos sus descendientes) El nodo es borrado del documento original.
<code>close()</code>	Cierra el flujo abierto previamente con <code>document.open()</code> .
<code>createAttribute()</code>	Crea un atributo de un nodo.
<code>createComment()</code>	Crea un nodo comentario con un texto.
<code>createDocumentFragment()</code>	Crea un nodo fragmento de documento vacío. Sirve para extraer, modificar trozos del documento original y/o añadir posteriormente al mismo.
<code>createElement()</code>	Crea un nodo elemento.
<code>createTextNode()</code>	Crea un nodo de texto.
<code>getElementById(id)</code>	Devuelve el elemento con dicho <code>id</code> .

2.2.- Objetos de más alto nivel.

2.2.4.- Objeto document

Método	Descripción
<code>getElementsByClassName (className)</code>	Devuelve los elementos con dicho <code>className</code> .
<code>getElementsByName (name)</code>	Devuelve los elementos con dicho <code>name</code> .
<code>getElementsByTagName (tagName)</code>	Devuelve los elementos con dicho <code>tagName</code> .
<code>hasFocus ()</code>	Indica si el documento tiene el foco.
<code>importNode (nodo , descendientes)</code>	Importa un nodo de otro documento sin borrarlo del original. Si el parámetro descendientes es <code>true</code> importa también sus descendientes.
<code>normalize ()</code>	Borra los nodos texto vacíos y une los adyacentes.
<code>open ()</code>	Abre un flujo de salida para ordenes <code>write</code> y/o <code>writeln</code> que se visualizan en el documento cuando se cierra con <code>close()</code> .
<code>querySelector ()</code>	Devuelve el primer elemento cuyo selector concuerda con el especificado.
<code>querySelectorAll ()</code>	Lo mismo que la anterior pero crea un nodo lista con todos los que concuerdan.

2.2.- Objetos de más alto nivel.

2.2.4.- Objeto document

Método	Descripción
<code>removeEventListener()</code>	Borra un gestor de eventos añadido previamente con <code>addEventListener()</code> .
<code>write(expresión)</code>	Escribe expresiones HTML y/o código JavaScript en el documento.
<code>writeln(expresión)</code>	Como <code>write()</code> pero añade al final un salto de línea.

2.2.- Objetos de más alto nivel.

2.2.4.- Objeto document

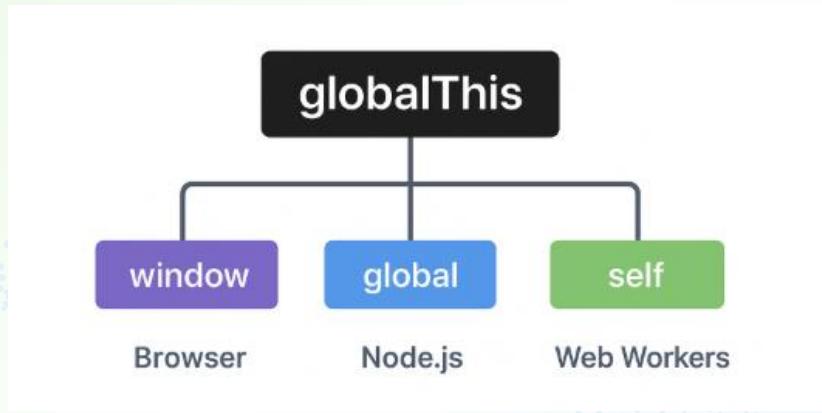
- ✓ Las colecciones nos permiten acceder a conjuntos de objetos de manera sencilla.

Colecciones del objeto Document	
Colección	Descripción
<code>anchors</code>	Contiene todas las anclas del documento (zonas etiquetadas con <code></code>).
<code>embeds</code>	Contiene todos los elementos embebidos del documento (etiqueta <code><embed></code>).
<code>forms</code>	Contiene todos los formularios del documento (etiqueta <code><form ...></code>).
<code>images</code>	Contiene todas las imágenes del documento (etiqueta <code></code>).
<code>links</code>	Contiene todos los enlaces del documento (etiqueta <code></code>).
<code>scripts</code>	Contiene todos los script del documento (etiqueta <code><script></code>).

2.3.- Objeto global estándar.

2.3.1.- Objeto globalThis

El objeto `globalThis` es equivalente a `window`, pero proporciona una forma estándar de acceder al objeto global, independientemente de si se está en navegador, Node.js o Web Worker.



2.3.- Objeto global estándar.

2.3.1.- Objeto globalThis



Navegador

```
console.log(window === globalThis); //true
```



Node.js

```
console.log(global === globalThis); //true
```



Web Workers

```
console.log(self === globalThis); //true
```

2.4.- Objetos nativos en Javascript.

Los objetos nativos no dependen del navegador

- Date
- Math
- Number
- Boolean
- String

Sintaxis para crear un objeto:

```
var / let / const miObjeto = new Clase();
```

2.4.- Objetos Nativos en JavaScript.

2.4.1.- Objeto Date

El objeto **Date** se utiliza para trabajar con fechas y horas.

Instanciar:

```
const fechaHoraEspecifica = new Date(2024, 8, 22, 14, 30, 0); // Año,  
mes, día, hora, minuto, segundo  
  
const fechaActual = new Date();  
  
const fechaEspecifica = new Date('2024-09-23');  
  
const fechaMilisegundo = new Date (1212121212);
```

2.4.- Objetos Nativos en JavaScript.

2.4.1.- Objeto Date

Propiedades:

Propiedad	Descripción
constructor	Devuelve la función que creó el objeto Date.
prototype	Te permitirá añadir propiedades y métodos a un objeto.

2.4.- Objetos Nativos en JavaScript.

2.4.1.- Objeto Date

Métodos:

Método	Descripción
getDate()	Devuelve el día del mes (de 1-31).
getDay()	Devuelve el día de la semana (de 0-6).
getFullYear()	Devuelve el año (4 dígitos).
getHours()	Devuelve la hora (de 0-23).
getMilliseconds()	Devuelve los milisegundos (de 0-999).
getMinutes()	Devuelve los minutos (de 0-59).
getMonth()	Devuelve el mes (de 0-11).
getSeconds()	Devuelve los segundos (de 0-59).

2.4.- Objetos Nativos en JavaScript.

2.4.1.- Objeto Date

Métodos:

Método	Descripción
getTime()	Devuelve los milisegundos desde media noche del 1 de Enero de 1970.
getTimezoneOffset()	Devuelve la diferencia de tiempo entre GMT y la hora local, en minutos.
getUTCDate()	Devuelve el día del mes en base a la hora UTC (de 1-31).
getUTCDay()	Devuelve el día de la semana en base a la hora UTC (de 0-6).
getUTCFullYear()	Devuelve el año en base a la hora UTC (4 dígitos).
setDate()	Ajusta el día del mes del objeto (de 1-31).
setFullYear()	Ajusta el año del objeto (4 dígitos).
setHours()	Ajusta la hora del objeto (de 0-23).

2.4.- Objetos Nativos en JavaScript.

2.4.2.- Objeto Math

El objeto **Math** permite realizar operaciones matemáticas. **Math** no es un constructor, por tanto, no permite instanciar un objeto.

Crear:

```
const numPi = Math.PI; //devuelve el número PI
```

```
const raizC= Math.sqrt(16); //devuelve la raíz cuadrada de 16
```

2.4.- Objetos Nativos en JavaScript.

2.4.2.- Objeto Math

Propiedades:

Propiedad	Descripción
E	Devuelve el número Euler (aproximadamente 2.718).
LN2	Devuelve el logaritmo neperiano de 2 (aproximadamente 0.693).
LN10	Devuelve el logaritmo neperiano de 10 (aproximadamente 2.302).
LOG2E	Devuelve el logaritmo base 2 de E (aproximadamente 1.442).
LOG10E	Devuelve el logaritmo base 10 de E (aproximadamente 0.434).
PI	Devuelve el número PI (aproximadamente 3.14159).
SQRT2	Devuelve la raíz cuadrada de 2 (aproximadamente 1.414).

2.4.- Objetos Nativos en JavaScript.

2.4.2.- Objeto Math

Métodos:

Método	Descripción
abs(x)	Devuelve el valor absoluto de x.
acos(x)	Devuelve el arcocoseno de x, en radianes.
asin(x)	Devuelve el arcoseno de x, en radianes.
atan(x)	Devuelve el arcotangente de x, en radianes con un valor entre -PI/2 y PI/2.
atan2(y,x)	Devuelve el arcotangente del cociente de sus argumentos.
ceil(x)	Devuelve el número x redondeado al alta hacia el siguiente entero.
cos(x)	Devuelve el coseno de x (x está en radianes).
floor(x)	Devuelve el número x redondeado a la baja hacia el anterior entero.
log(x)	Devuelve el logaritmo neperiano (base E) de x.

2.4.- Objetos Nativos en JavaScript.

2.4.2.- Objeto Math

Métodos:

Método	Descripción
max(x,y,z,...,n)	Devuelve el número más alto de los que se pasan como parámetros.
min(x,y,z,...,n)	Devuelve el número más bajo de los que se pasan como parámetros.
pow(x,y)	Devuelve el resultado de x elevado a y.
random()	Devuelve un número al azar entre 0 y 1.
round(x)	Redondea x al entero más próximo.
sin(x)	Devuelve el seno de x (x está en radianes).
sqrt(x)	Devuelve la raíz cuadrada de x.
tan(x)	Devuelve la tangente de un ángulo.

2.4.- Objetos Nativos en JavaScript.

2.4.3.- Objeto Number

El objeto **Number** no se suele instanciar; normalmente asignamos valores numéricos a una variable.

Crear:

```
const num=new Number(23);
```

```
const num2=23;
```

Si el parámetro que se pasa al constructor no se puede convertir devuelve el valor **NaN** (Not a Number).

```
const num=new Number("a23"); // devuelve NaN
```

2.4.- Objetos Nativos en JavaScript.

2.4.3.- Objeto Number

Propiedades:

Propiedad	Descripción
constructor	Devuelve la función que creó el objeto Number.
MAX_VALUE	Devuelve el número más alto disponible en JavaScript.
MIN_VALUE	Devuelve el número más pequeño disponible en JavaScript.
NEGATIVE_INFINITY	Representa a infinito negativo (se devuelve en caso de overflow).
POSITIVE_INFINITY	Representa a infinito positivo (se devuelve en caso de overflow).
prototype	Permite añadir nuestras propias propiedades y métodos a un objeto.

2.4.- Objetos Nativos en JavaScript.

2.4.3.- Objeto Number

Métodos:

Método	Descripción
toExponential(x)	Convierte un número a su notación exponencial.
toFixed(x)	Formatea un número con x dígitos decimales después del punto decimal.
toPrecision(x)	Formatea un número a la longitud x.
toString()	Convierte un objeto Number en una cadena. • Si se pone 2 como parámetro se mostrará el número en binario. • Si se pone 8 como parámetro se mostrará el número en octal. • Si se pone 16 como parámetro se mostrará el número en hexadecimal.
valueOf()	Devuelve el valor primitivo de un objeto Number.

2.4.- Objetos Nativos en JavaScript.

2.4.4.- Objeto Boolean

El objeto **Boolean** se utiliza para convertir un valor no booleano, en un valor booleano (true o false).

```
const bool=new Boolean(1);
const semaforo =true;
console.log(bool.toString()); //devuelve "true"
console.log(semaforo.valueOf()); //devuelve true
```

2.4.- Objetos Nativos en JavaScript.

2.4.4.- Objeto Boolean

Propiedades:

Propiedad	Descripción
constructor	Devuelve la función que creó el objeto Boolean.
prototype	Te permitirá añadir propiedades y métodos a un objeto.

Métodos:

Método	Descripción
toString()	Convierte un valor Boolean a una cadena y devuelve el resultado.
valueOf()	Devuelve el valor primitivo de un objeto Boolean.

2.4.- Objetos Nativos en JavaScript.

2.4.5.- Objeto String

Instanciación:

- **Podemos utilizar dos maneras:**

- const texto = new String ("JavaScript");
- const texto= "JavaScript";

- **Podemos utilizar comillas dobles o simples:**

- const texto = '<input type="checkbox" name="coche" />Audi A6';
- const texto= "<input type='checkbox' name='coche' />Audi A6';

- **Podemos utilizar caracteres de escape:**

- const texto = '<input type=\"checkbox\" name=\"coche\" />Audi A6';

2.4.- Objetos Nativos en JavaScript.

2.4.5.- Objeto String

- Podemos concatenar cadenas muy largas con +=:

- let texto = ""
• texto+= "JavaScript";
• texto+= " , lenguaje interpretado";

- Podemos concatenar cadenas con variables con el operador +:

- const nombreEquipo = prompt("Introduce el nombre de tu equipo favorito");
• const mensaje = "El "+ nombreEquipo + " ha sido el campeón de la liga";

- Interpolación de variables en cadenas mediante los Template Strings:

- const mensaje= `El \${nombreEquipo} ha sido el campeón de la liga`;

2.4.- Objetos Nativos en JavaScript.

2.4.5.- Objeto String

Propiedades:

Propiedad	Descripción
length	Contiene la longitud de una cadena.

Métodos:

Método	Descripción
charAt()	Devuelve el carácter especificado por la posición que se indica entre paréntesis.
charCodeAt()	Devuelve el Unicode del carácter especificado por la posición que se indica entre paréntesis.
concat()	Une una o más cadenas y devuelve el resultado de esa unión.
fromCharCode()	Convierte valores Unicode a caracteres.
indexOf()	Devuelve la posición de la primera ocurrencia del carácter buscado en la cadena.
lastIndexOf()	Devuelve la posición de la última ocurrencia del carácter buscado en la cadena.

2.4.- Objetos Nativos en JavaScript.

2.4.5.- Objeto String

Métodos:

Método	Descripción
match()	Busca una coincidencia entre una expresión regular y una cadena y devuelve las coincidencias o null si no ha encontrado nada.
replace()	Busca una subcadena en la cadena y la reemplaza por la nueva cadena especificada.
search()	Busca una subcadena en la cadena y devuelve la posición dónde se encontró.
slice()	Extrae una parte de la cadena y devuelve una nueva cadena.
split()	Divide una cadena en un array de subcadenas.
substr()	Extrae los caracteres de una cadena, comenzando en una determinada posición y con el número de caracteres indicado.
substring()	Extrae los caracteres de una cadena entre dos índices especificados.
toLowerCase()	Convierte una cadena en minúsculas.
toUpperCase()	Convierte una cadena en mayúsculas.

2.4.- Objetos Nativos en JavaScript.

2.4.5.- Objeto String

Métodos:

Método	Descripción
includes()	Devuelve true si una cadena contiene una subcadena
localeCompare()	Devuelve un número que indica si la cadena de referencia viene antes, después o es equivalente a la cadena dada
padEnd()	Rellena la cadena actual desde el final, con otra cadena
padStart()	Rellena la cadena actual desde el principio con otra cadena
replaceAll()	Busca una subcadena en la cadena y reemplaza todas subcadenas
repeat()	Devuelve una cadena con una serie de copias de esa cadena
startsWith()	Devuelve true si es una cadena comienza con la subcadena especificada
trimEnd()	Elimina los espacios que hay al final de una cadena
trimStart()	Elimina los espacios que hay al comienzo de una cadena

2.5.- Objetos globales avanzados en fechas.

2.5.2.- Objeto Intl.DateTimeFormat()

El objeto `Intl.DateTimeFormat()` es una API de internacionalización de JavaScript que permite dar formato a fechas y horas de manera localizada, adaptándose al idioma de un país o región.

Sintaxis:

```
new Intl.DateTimeFormat([locales [, opciones]])
```

- **locales** (opcional): una cadena o un array de cadenas que indican el idioma y la región. Ejemplo: 'es-ES', 'en-US', ...
- **opciones** (opcional): objeto que especifica el estilo de fecha y hora a mostrar.

2.5.- Objetos globales avanzados en fechas.

2.5.1.- Objeto Intl.DateTimeFormat()

Opciones más comunes

Opción	Valores posibles	Descripción
dateStyle	“short”, “médium”, “long”, “full”	Define el estilo de la fecha.
timeStyle	“short”, “médium”, “long”, “full”	Define el estilo de la hora.
hour12	True /false	Indica si usar formato de 12 horas o 24 horas.
timeZone	“UTC”, “Europe/Madrid”	Define la zona horaria de la fecha.

2.5.- Objetos globales avanzados en fechas.

2.5.1.- Objeto Intl.DateTimeFormat()

Ejemplo

```
const fecha = new Date();

// Formato corto en español
const corto = new Intl.DateTimeFormat ('es-ES', { dateStyle: 'short' });
console.log (corto.format(fecha)); // Ej: "18/09/25"
```

2.5.- Objetos globales avanzados en fechas.

2.5.2.- Temporal

La API Temporal es una propuesta moderna de JavaScript para gestionar fechas y horas de forma segura y precisa, evitando los problemas tradicionales de Date, como:

- Diferencias de zona horaria complicadas.
- Cálculos de fechas poco precisos.
- Formatos inconsistentes entre navegadores.

Temporal permite trabajar con instantes, fechas, horas, zonas horarias y duraciones de manera clara y confiable.

2.5.- Objetos globales avanzados en fechas.

2.5.2.- Temporal

Opciones más comunes

Objeto Temporal	Descripción
Temporal.PlainDate	Representa solo una fecha (día, mes, año)
Temporal.PlainTime	Representa solo la hora(hora, minutos, segundos)
Temporal.PlainDateTime	Combina fecha y hora, sin zona horaria
Temporal.ZonedDateTime	Fecha y hora con zona horaria
Temporal.Duration	Representa una duración(días, horas, minutos, segundos)
Temporal.Instant	Punto específico en el tiempo UTC

2.5.- Objetos globales avanzados en fechas.

2.5.2.- Temporal

Ejemplos//

// Solo fecha

```
const fecha = new Temporal.PlainDate(2025, 9, 18);
```

// Solo hora

```
const hora = new Temporal.PlainTime(14, 30, 0);
```

// Fecha y hora

```
const fechaHora = new Temporal.PlainDateTime(2025, 9, 18, 14, 30, 0);
```

// Fecha y hora con zona horaria

```
const zona = Temporal.ZonedDateTime.from('2025-09-18T14:30:00-05:00[America/New_York]');
```