Características de JavaScript

¿Qué es JavaScript?

- Lenguaje de Script en la web
- Lenguaje interpretado
- Utilizado para dar un comportamiento dinámico a las páginas.
- Gramática sencilla, soporta programación orientada a objetos.
- Cualquier navegador soporta JS.
- JS puede modificar elementos HTML y CSS.
- Objetos utilizan herencia basada en prototipos.
- Débilmente tipado.
- Desarrollado en 1995 por Brendan Eich de Netscape con el nombre de Mocha
- ECMAScript en su especificación estándar.

Integración del código con las etiquetas HTML

Hay tres formas de incluir código JS en una página HTML.

- **Estilo interno:** Se utilizan las etiquetas <script></script>, se recomienda incluir en la cabecera del documento aunque se puede añadir en cualquier parte. Se utiliza cuando se definen instrucciones que se referencian desde cualquier parte del código o cuando se definen funciones con fragmentos de código genéricos.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html;
      charset=utf-8">
    <title>Hola Mundo</title>
  </head>
  <body>
    <script>
      alert('Hola mundo en JavaScript')
    </script>
  </body>
</html>
```

- **Estilo externo:**
  - JS en un archivo externo, se puede utilizar el mismo código de JS para diferentes documentos HTML , lo que mejora el mantenimiento.
  - Aprovecha mejor la caché, carga más rápido.

- El mismo código escrito entre <script></script> se puede almacenar en un fichero .js.
- Se accede a los ficheros .js mediante el documento HTML/XHTML y las etiquetas <script></script>
- No hay numero limite de ficheros .js que se pueden enlazar al HTML
- Utilizar el atributo **src** de la etiqueta **<script>** para especificar el fichero que contiene el código JS.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html;
      charset=utf-8">
    <title>Hola Mundo</title>
    <script type="text/javascript" src="HolaMundo.js">
    </script>
  </head>
  <body></body>
</html>
```

El fichero `HolaMundo.js` debe contener:

```
alert('Hola Mundo en JavaScript')
```

- **Estilo en línea:**
  - Consiste en insertar fragmentos de JS dentro de atributos de etiquetas HTML de la página.
  - Controlar lo que pasa a un elemento HTML en concreto.
  - Desventaja: Mantenimiento y modificación más tediosa.
  - Solo para casos especiales.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html;
      charset=utf-8">
  </head>
  <body>
    <p onclick="alert('funcion');">
      Hola mundo
    </p>
  </body>
</html>
```

Colocación idónea del JavaScript

Antes se situaba en la parte del <head>, pero tiene dos problemas:
- El script no tiene acceso al DOM del doc.
- Retrasa la carga de la página.

Actualmente se sitúa al final del <body>, pero tiene otro problema:
- La pag HTML puede estar disponible antes que el JScript.

Se podría optar por utilizar una colocación mixta.

El lenguaje JavaScript: Sintaxis

Javascript especifica aspectos como la definición de comentarios, nombre de variables, separación entre diferentes instrucciones, mayúsculas y minúsculas.

- Comentarios en el código:
  - No se interpretan.
  - Dos forma de insertar:
    - Doble barra “//”
    - ”/*” Inicio, “*/”final.
- Tabulación y saltos de línea:
  - JS ignora espacios, tabulaciones y saltos de línea.
  - Tabulación para mejorar la presentación.
- Punto y coma:
  - Al final de instrucciones.
- Palabras reservadas:
  - Palabras que no se pueden utilizar como nombre de variables.
- Etiqueta <noscript>:
  - Muestra contenido alternativo cuando JS está desactivado, puede ser utilizado para mandar un mensaje para que se habiliten las funciones de JS.

Tipos de datos

Especifica qué tipo de valor se guardará en una determinada variable.
Los tres tipos primitivos son:
- Números
  - Se le llama number, formato llamado double.
- Cadenas de texto
  - Los llamados *string*, cualquier carácter Unicode, entre comillas sobres o simples.
- Valores booleanos
  - Los conocidos *boolean,* aceptan valores *true o false.*

En JS, menos los datos primitivos, TODO son objetos.

Variables

- Declaración de variables:
  - Se declaran mediante *var* y el nombre de la variable.
  - Se pueden declarar una seguida de la otra separadas por comas.
  - Ámbito global (no como *let y const* que es local)
- Inicialización de variables:
  - Se le puede asignar un valor directamente:
    - `var mi_variable_1 = 30;`
  - A través de un calculo:
    - `var mi_variable_2 = mi_variable_1 + 10;`
  - Solicitándolo al usuario:
    - `var mi_variable_3 = prompt('Introduce un valor:');`
- Los tipos de datos son dinámicos. Una variable puede cambiar de tipo:
  - `var x = 7;`
  - `x = true;`
  - `x = "hola";`
  - `x = ["lunes", "martes", "miércoles"];`
  - `x = {nombre: "Carlos", apellidos: "Huertas", DNI: "123456789-X", edad: 27};`

Operadores

Para construir expresiones con las que realizar cálculos más complejos.
JS utiliza cinco tipos de operadores principales:

- Aritméticos:
  - Cálculos elementales entre variables numéricas:

Operador	Nombre
+	Suma
-	Resta
*	Multiplicación
**	Exponenciación
/	División
%	Módulo (Resto)
++	Incremento
--	Decremento

- Lógicos:

Operador	Nombre
&&	Y
||	O
!	No / Negación

- Asignación:

Operador	Nombre
+=	Suma y asigna
-=	Resta y asigna
*=	Multiplica y asigna
/*	Divide y asigna
%=	Módulo y asigna

- Comparación:

Operador	Nombre
<	Menor que
<=	Menor o igual que
==	Igual
>	Mayor que
>=	Mayor o igual que
!=	Diferente
===	Estrictamente igual
!==	Estrictamente diferente

- Condicionales:

Operador	Nombre
?:	Condicional

  - 1 - EJEMPLO:

```
<script type="text/javascript">
  var dividendo = prompt("Introduce el dividendo: ");
  var divisor = prompt("Introduce el divisor: ");
  var resultado;
  divisor != 0 ? resultado = dividendo/divisor :
    alert("No es posible la división por cero");
    alert("El resultado es: " + resultado);
</script>
```

  - 2 - EJEMPLO:

    - Ejercicio: Desarrolla un programa JavaScript que calcule el área y el perímetro de un rectángulo.

```
<script>
  var lado1 = prompt("Introduce la longitud del lado 1:", "");
  var lado2 = prompt("Introduce la longitud del lado 2:", "");

  area = lado1 * lado2;
  perimetro = parseInt(lado1) + parseInt(lado1) +
       parseInt(lado2) + parseInt(lado2);

  document.write("El área del rectángulo es: " + area);
  document.write("<br>");
  document.write("El perímetro del rectángulo es: " +
       perimetro);
</script>
```

El objeto Date -Métodos:

1. getFullYear()
   - Devuelve el año (4 dígitos).
   - Ejemplo: console.log(new Date().getFullYear()); // 2023

2. getMonth()
   - Devuelve el mes (0-11).
   - Ejemplo: console.log(new Date().getMonth()); // 9 (para octubre)

3. getDate()
   - Devuelve el día del mes (1-31).
   - Ejemplo: console.log(new Date().getDate()); // 19

4. getDay()
   - Devuelve el día de la semana (0=Domingo, 1=Lunes, ...).
   - Ejemplo: console.log(new Date().getDay()); // 3 (para miércoles)

5. getHours()
   - Devuelve la hora (0-23).
   - Ejemplo: console.log(new Date().getHours()); // 15

6. getMinutes()
   - Devuelve los minutos (0-59).
   - Ejemplo: console.log(new Date().getMinutes()); // 30

7. getSeconds()
   - Devuelve los segundos (0-59).
   - Ejemplo: console.log(new Date().getSeconds()); // 45

8. getMilliseconds()
   - Devuelve los milisegundos (0-999).
   - Ejemplo: console.log(new Date().getMilliseconds()); // 500

9. getTime()
   - Devuelve el tiempo en milisegundos desde el 1 de enero de 1970 (UTC).
   - Ejemplo: console.log(new Date().getTime()); // 1645164645500 (un valor en milisegundos)

10. getTimezoneOffset()
    - Devuelve la diferencia en minutos entre la hora local y UTC.
    - Ejemplo: console.log(new Date().getTimezoneOffset()); // -240 (para horario de verano en UTC-4)

11. toString()
    - Devuelve una representación en cadena de la fecha y la hora.

- Ejemplo: console.log(new Date().toString()); // Wed Oct 19 2023 15:30:45 GMT-0400 (hora de verano oriental)

12. toDateString()
    - Devuelve la parte de fecha de la representación en cadena.
    - Ejemplo: console.log(new Date().toDateString()); // Wed Oct 19 2023

13. toTimeString()
    - Devuelve la parte de tiempo de la representación en cadena.
    - Ejemplo: console.log(new Date().toTimeString()); // 15:30:45 GMT-0400 (hora de verano oriental)

14. toLocaleString()
    - Devuelve la representación en cadena de la fecha y la hora en formato local.
    - Ejemplo: console.log(new Date().toLocaleString()); // 19/10/2023, 15:30:45

15. toLocaleDateString()
    - Devuelve la parte de fecha de la representación en cadena en formato local.
    - Ejemplo: console.log(new Date().toLocaleDateString()); // 19/10/2023

16. toLocaleTimeString()
    - Devuelve la parte de tiempo de la representación en cadena en formato local.
    - Ejemplo: console.log(new Date().toLocaleTimeString()); // 15:30:45

17. toISOString()
    - Devuelve una representación en formato ISO de la fecha y la hora.
    - Ejemplo: console.log(new Date().toISOString()); // 2023-10-19T19:30:45.500Z

18. toJSON()
    - Devuelve una cadena JSON que representa la fecha y la hora.
    - Ejemplo: console.log(new Date().toJSON()); // "2023-10-19T19:30:45.500Z"

19. toUTCString()
    - Devuelve una representación en cadena en formato UTC.
    - Ejemplo: console.log(new Date().toUTCString()); // Wed, 19 Oct 2023 19:30:45 GMT

20. toGMTString()
    - Devuelve una representación en cadena en formato GMT (obsoleto).
    - Ejemplo: console.log(new Date().toGMTString()); // Wed, 19 Oct 2023 19:30:45 GMT

21. valueOf()
    - Devuelve el valor primitivo de la fecha (equivalente a getTime()).
    - Ejemplo: console.log(new Date().valueOf()); // 1645164645500

22. setFullYear()
    - Establece el año.
    - Ejemplo: const fecha = new Date(); fecha.setFullYear(2024); console.log(fecha.getFullYear()); // 2024

23. setMonth()
   - Establece el mes (0-11).
   - Ejemplo: const fecha = new Date(); fecha.setMonth(5); console.log(fecha.getMonth()); // 5

24. setDate()
   - Establece el día del mes.
   - Ejemplo: const fecha = new Date(); fecha.setDate(15); console.log(fecha.getDate()); // 15

25. setHours()
   - Establece la hora (0-23).
   - Ejemplo: const fecha = new Date(); fecha.setHours(14); console.log(fecha.getHours()); // 14

26. setMinutes()
   - Establece los minutos (0-59).
   - Ejemplo: const fecha = new Date(); fecha.setMinutes(45); console.log(fecha.getMinutes()); // 45

27. setSeconds()
   - Establece los segundos (0-59).
   - Ejemplo: const fecha = new Date(); fecha.setSeconds(30); console.log(fecha.getSeconds()); // 30

28. setMilliseconds()
   - Establece los milisegundos (0-999).
   - Ejemplo: const fecha = new Date(); fecha.setMilliseconds(750); console.log(fecha.getMilliseconds()); // 750

29. setTime()
   - Establece el tiempo en milisegundos desde el 1 de enero de 1970.
   - Ejemplo: const fecha = new Date(); fecha.setTime(1645168000000); console.log(fecha.getTime()); // 1645168000000

30. setUTCFullYear()
   - Establece el año en formato UTC.
   - Ejemplo: const fecha = new Date(); fecha.setUTCFullYear(2024); console.log(fecha.getUTCFullYear()); // 2024

31. setUTCMonth()
   - Establece el mes en formato UTC.
   - Ejemplo: const fecha = new Date(); fecha.setUTCMonth(5); console.log(fecha.getUTCMonth()); // 5

32. setUTCDate()
   - Establece el día del mes en formato UTC.

- Ejemplo: const fecha = new Date(); fecha.setUTCDate(15);
console.log(fecha.getUTCDate()); // 15

33. setUTCHours()
- Establece la hora en formato UTC.
- Ejemplo: const fecha = new Date(); fecha.setUTCHours(14);
console.log(fecha.getUTCHours()); // 14

34. setUTCMinutes()
- Establece los minutos en formato UTC.
- Ejemplo: const fecha = new Date(); fecha.setUTCMinutes(45);
console.log(fecha.getUTCMinutes()); // 45

35. setUTCSeconds()
- Establece los segundos en formato UTC.
- Ejemplo: const fecha = new Date(); fecha.setUTCSeconds(30);
console.log(fecha.getUTCSeconds()); // 30

36. setUTCMilliseconds()
- Establece los milisegundos en formato UTC.
- Ejemplo: const fecha = new Date(); fecha.setUTCMilliseconds(750);
console.log(fecha.getUTCMilliseconds()); // 750

37. toDateString()
- Devuelve la fecha en formato de cadena amigable para humanos (obsoleto).
- Ejemplo: console.log(new Date().toDateString()); // Wed Oct 19 2023

38. toGMTString()
- Devuelve una representación en cadena en formato GMT (obsoleto).
- Ejemplo: console.log(new Date().toGMTString()); // Wed, 19 Oct 2023 19:30:45 GMT

39. toLocaleDateString()
- Devuelve la fecha en formato local.
- Ejemplo: console.log(new Date().toLocaleDateString()); // 19/10/2023

40. toLocaleTimeString()
- Devuelve la hora en formato local.
- Ejemplo: console.log(new Date().toLocaleTimeString()); // 15:30:45

Objeto Math - Métodos

1. Math.abs()
   - Devuelve el valor absoluto de un número.
   - Ejemplo: console.log(Math.abs(-5)); // 5

2. Math.ceil()
   - Redondea un número hacia arriba al entero más cercano.
   - Ejemplo: console.log(Math.ceil(3.2)); // 4

3. Math.floor()
   - Redondea un número hacia abajo al entero más cercano.
   - Ejemplo: console.log(Math.floor(3.8)); // 3

4. Math.round()
   - Redondea un número al entero más cercano.
   - Ejemplo: console.log(Math.round(3.5)); // 4

5. Math.max()
   - Devuelve el número más grande de una lista de argumentos.
   - Ejemplo: console.log(Math.max(2, 6, 4, 9, 1)); // 9

6. Math.min()
   - Devuelve el número más pequeño de una lista de argumentos.
   - Ejemplo: console.log(Math.min(2, 6, 4, 9, 1)); // 1

7. Math.pow()
   - Eleva un número a una potencia.
   - Ejemplo: console.log(Math.pow(2, 3)); // 8

8. Math.sqrt()
   - Calcula la raíz cuadrada de un número.
   - Ejemplo: console.log(Math.sqrt(25)); // 5

9. Math.random()
   - Genera un número decimal aleatorio entre 0 (inclusive) y 1 (exclusivo).
   - Ejemplo: console.log(Math.random());

10. Math.sin()
    - Calcula el seno de un ángulo en radianes.
    - Ejemplo: console.log(Math.sin(Math.PI / 2)); // 1 (seno de 90 grados)

11. Math.cos()
    - Calcula el coseno de un ángulo en radianes.
    - Ejemplo: console.log(Math.cos(0)); // 1 (coseno de 0 radianes)

12. Math.tan()
    - Calcula la tangente de un ángulo en radianes.
    - Ejemplo: console.log(Math.tan(Math.PI / 4)); // 1 (tangente de 45 grados)

13. Math.asin()
    - Calcula el arco seno de un número, devolviendo el ángulo en radianes.
    - Ejemplo: console.log(Math.asin(1)); // 1.5707963267948966 (arco seno de 1)

14. Math.acos()
    - Calcula el arco coseno de un número, devolviendo el ángulo en radianes.
    - Ejemplo: console.log(Math.acos(0)); // 1.5707963267948966 (arco coseno de 0)

15. Math.atan()
    - Calcula el arco tangente de un número, devolviendo el ángulo en radianes.
    - Ejemplo: console.log(Math.atan(1)); // 0.7853981633974483 (arco tangente de 1)

16. Math.log()
    - Calcula el logaritmo natural de un número.
    - Ejemplo: console.log(Math.log(Math.E)); // 1 (logaritmo natural de Euler)

17. Math.exp()
    - Calcula el valor de la función exponencial e^x.
    - Ejemplo: console.log(Math.exp(2)); // 7.3890560989306495 (e elevado a la potencia 2)

Objeto Number - Métodos

1. Number.isNaN()
   - Verifica si un valor es NaN (No es un número).
   - Ejemplo: console.log(Number.isNaN(5)); // false

2. Number.parseFloat()
   - Convierte una cadena en un número de punto flotante.
   - Ejemplo: console.log(Number.parseFloat("3.14")); // 3.14

3. Number.parseInt()
   - Convierte una cadena en un número entero.
   - Ejemplo: console.log(Number.parseInt("42")); // 42

4. Number.isFinite()
   - Verifica si un valor es un número finito (no es NaN ni infinito).
   - Ejemplo: console.log(Number.isFinite(7)); // true

5. Number.isInteger()
   - Verifica si un valor es un número entero.
   - Ejemplo: console.log(Number.isInteger(3.0)); // true

6. Number.isSafeInteger()
   - Verifica si un valor es un número entero seguro (dentro del rango seguro de representación).
   - Ejemplo: console.log(Number.isSafeInteger(9007199254740991)); // true

7. Number.MIN_SAFE_INTEGER
   - Devuelve el valor mínimo de un número entero seguro.
   - Ejemplo: console.log(Number.MIN_SAFE_INTEGER); // -9007199254740991

8. Number.MAX_SAFE_INTEGER
   - Devuelve el valor máximo de un número entero seguro.
   - Ejemplo: console.log(Number.MAX_SAFE_INTEGER); // 9007199254740991

9. Number.MAX_VALUE
   - Devuelve el número más grande representable en JavaScript.
   - Ejemplo: console.log(Number.MAX_VALUE);

10. Number.MIN_VALUE
    - Devuelve el número positivo más pequeño (mayor que cero) representable en JavaScript.
    - Ejemplo: console.log(Number.MIN_VALUE);

11. Number.NaN
    - Representa el valor NaN (No es un número).
    - Ejemplo: console.log(Number.NaN);

12. Number.POSITIVE_INFINITY
    - Representa el infinito positivo.
    - Ejemplo: console.log(Number.POSITIVE_INFINITY);

13. Number.NEGATIVE_INFINITY
    - Representa el infinito negativo.
    - Ejemplo: console.log(Number.NEGATIVE_INFINITY);

14. Number.toExponential()
    - Convierte un número en una cadena en notación exponencial.
    - Ejemplo: console.log((1234).toExponential(2)); // "1.23e+3"

15. Number.toFixed()
    - Formatea un número con un número fijo de decimales.
    - Ejemplo: console.log((3.14159).toFixed(2)); // "3.14"

16. Number.toPrecision()
    - Formatea un número con una longitud de dígitos total.

- Ejemplo: console.log((1234.5678).toPrecision(4)); // "1235"

17. Number.toString()
    - Convierte un número en una cadena con una base específica (decimal por defecto).
    - Ejemplo: console.log((10).toString(2)); // "1010"

Objeto String - Métodos

1. String.length
   - Devuelve la longitud de una cadena de texto.
   Ejemplo: "Hola, mundo".length

2. String.charAt()
   - Devuelve el carácter en la posición especificada de una cadena.
   Ejemplo: "Hola".charAt(1)

3. String.charCodeAt()
   - Devuelve el valor Unicode del carácter en la posición especificada.
   Ejemplo: "Hola".charCodeAt(1)

4. String.concat()
   - Combina dos o más cadenas de texto y devuelve una nueva cadena.
   Ejemplo: "Hola, ".concat("mundo")

5. String.indexOf()
   - Devuelve la posición de la primera aparición de una subcadena en una cadena.
   Ejemplo: "Hola, mundo".indexOf("mundo")

6. String.lastIndexOf()
   - Devuelve la posición de la última aparición de una subcadena en una cadena.
   Ejemplo: "Hola, mundo".lastIndexOf("o")

7. String.slice()
   - Extrae una parte de una cadena y la devuelve como una nueva cadena.
   Ejemplo: "JavaScript".slice(0, 4)

8. String.substring()
   - Similar a slice, pero no admite índices negativos.
   Ejemplo: "JavaScript".substring(4, 10)

9. String.substr()
   - Extrae una cantidad específica de caracteres a partir de una posición.
   Ejemplo: "JavaScript".substr(4, 6)

10. String.replace()
  - Reemplaza una subcadena con otra en una cadena.
  Ejemplo: "Hola, mundo".replace("mundo", "amigo")

11. String.toLowerCase()
  - Convierte una cadena a minúsculas.
  Ejemplo: "Hola, Mundo".toLowerCase()

12. String.toUpperCase()
  - Convierte una cadena a mayúsculas.
  Ejemplo: "Hola, Mundo".toUpperCase()

13. String.trim()
  - Elimina espacios en blanco al principio y al final de una cadena.
  Ejemplo: " Hola, mundo ".trim()

14. String.split()
  - Divide una cadena en un array de subcadenas basado en un delimitador.
  Ejemplo: "Manzana,Plátano,Uva".split(",")

15. String.startsWith()
  - Verifica si una cadena comienza con una subcadena específica.
  Ejemplo: "Hola, mundo".startsWith("Hola")

16. String.endsWith()
  - Verifica si una cadena termina con una subcadena específica.
  Ejemplo: "Hola, mundo".endsWith("mundo")

17. String.includes()
  - Verifica si una cadena contiene una subcadena específica.
  Ejemplo: "Hola, mundo".includes("mundo")

Objeto Array - Métodos

1. Array.length
  - Propiedad que devuelve la cantidad de elementos en un array.
  Ejemplo: const miArray = [1, 2, 3]; console.log(miArray.length); // 3

2. Array.push()
  - Añade uno o más elementos al final de un array y devuelve la nueva longitud.
  Ejemplo: const miArray = [1, 2, 3]; miArray.push(4); // miArray ahora es [1, 2, 3, 4]

3. Array.pop()
  - Elimina el último elemento de un array y lo devuelve.

Ejemplo: const miArray = [1, 2, 3]; const elementoEliminado = miArray.pop(); // elementoEliminado es 3

4. Array.unshift()
- Añade uno o más elementos al principio de un array y devuelve la nueva longitud.
Ejemplo: const miArray = [2, 3]; miArray.unshift(1); // miArray ahora es [1, 2, 3]

5. Array.shift()
- Elimina el primer elemento de un array y lo devuelve.
Ejemplo: const miArray = [1, 2, 3]; const elementoEliminado = miArray.shift(); // elementoEliminado es 1

6. Array.concat()
- Combina dos o más arrays y devuelve un nuevo array resultante.
Ejemplo: const array1 = [1, 2]; const array2 = [3, 4]; const newArray = array1.concat(array2); // newArray es [1, 2, 3, 4]

7. Array.join()
- Combina todos los elementos de un array en una cadena, separados por un separador.
Ejemplo: const miArray = [1, 2, 3]; const cadena = miArray.join(', '); // cadena es "1, 2, 3"

8. Array.slice()
- Extrae una parte de un array y la devuelve como un nuevo array.
Ejemplo: const miArray = [1, 2, 3, 4]; const subArray = miArray.slice(1, 3); // subArray es [2, 3]

9. Array.splice()
- Cambia el contenido de un array eliminando, reemplazando o agregando elementos.
Ejemplo: const miArray = [1, 2, 3, 4]; miArray.splice(1, 2, 5, 6); // miArray ahora es [1, 5, 6, 4]

10. Array.indexOf()
- Devuelve el primer índice en el que se encuentra un elemento en el array, o -1 si no se encuentra.
Ejemplo: const miArray = [1, 2, 3]; const indice = miArray.indexOf(2); // indice es 1

11. Array.lastIndexOf()
- Devuelve el último índice en el que se encuentra un elemento en el array, o -1 si no se encuentra.
Ejemplo: const miArray = [1, 2, 3, 2]; const indice = miArray.lastIndexOf(2); // indice es 3

12. Array.includes()
- Comprueba si un array contiene un elemento determinado y devuelve `true` o `false`.
Ejemplo: const miArray = [1, 2, 3]; const incluyeDos = miArray.includes(2); // incluyeDos es true

Objeto Navigator - Métodos

Atributos:

navigator.userAgent
Devuelve una cadena que contiene información sobre el agente de usuario del navegador.
Ejemplo: navigator.userAgent

navigator.platform
Devuelve una cadena que indica la plataforma del sistema en la que se ejecuta el navegador.
Ejemplo: navigator.platform

Métodos:

navigator.geolocation
Proporciona acceso a la ubicación geográfica del usuario.
Ejemplo (para obtener la ubicación actual del usuario):
javascript
Copy code

```
navigator.geolocation.getCurrentPosition(function(position) {
  const latitud = position.coords.latitude;
  const longitud = position.coords.longitude;
  console.log(`Latitud: ${latitud}, Longitud: ${longitud}`);
});
```

navigator.language
Devuelve una cadena que representa el idioma preferido del usuario.
Ejemplo: navigator.language

navigator.cookieEnabled
Indica si las cookies están habilitadas en el navegador.
Ejemplo: navigator.cookieEnabled

navigator.onLine
Indica si el navegador está en línea (conexión a Internet activa) o fuera de línea.
Ejemplo: navigator.onLine

navigator.userAgentData
Proporciona información detallada sobre la configuración del agente de usuario.
Ejemplo: navigator.userAgentData

Objeto Screen - Métodos

1. screen.width
   - Devuelve el ancho de la pantalla en píxeles.
   Ejemplo: const anchoPantalla = screen.width;

2. screen.height
   - Devuelve la altura de la pantalla en píxeles.
   Ejemplo: const alturaPantalla = screen.height;

3. screen.availWidth
   - Devuelve el ancho de la pantalla disponible para las ventanas del navegador en píxeles.
   Ejemplo: const anchoDisponible = screen.availWidth;

4. screen.availHeight
   - Devuelve la altura de la pantalla disponible para las ventanas del navegador en píxeles.
   Ejemplo: const alturaDisponible = screen.availHeight;

5. screen.orientation.lock()
   - Bloquea la orientación de la pantalla en una orientación específica.
   Ejemplo: screen.orientation.lock("portrait-primary");

6. screen.orientation.unlock()
   - Desbloquea la orientación de la pantalla.
   Ejemplo: screen.orientation.unlock();

7. screen.orientation.type
   - Devuelve el tipo de orientación de la pantalla.
   Ejemplo: const tipoOrientacion = screen.orientation.type;

Objeto Window - Métodos

// Métodos:
1. window.open()
   - Abre una nueva ventana del navegador.
   Ejemplo: window.open("https://www.ejemplo.com");

2. window.close()
   - Cierra la ventana actual.
   Ejemplo: window.close();

3. window.alert()
   - Muestra un cuadro de diálogo de alerta con un mensaje.
   Ejemplo: window.alert("¡Hola, mundo!");

4. window.confirm()
   - Muestra un cuadro de diálogo de confirmación con opciones "Aceptar" y "Cancelar".
   Ejemplo: const confirmado = window.confirm("¿Estás seguro?");

5. window.prompt()
   - Muestra un cuadro de diálogo de entrada de texto.
   Ejemplo: const nombre = window.prompt("Ingresa tu nombre:");

6. window.print()
   - Abre el cuadro de diálogo de impresión del navegador.
   Ejemplo: window.print();

7. window.scrollTo()
   - Desplaza la ventana a una posición específica en la página.
   Ejemplo: window.scrollTo(0, 500);

8. window.scrollBy()
   - Desplaza la ventana en relación a su posición actual.
   Ejemplo: window.scrollBy(0, 100);

9. window.resizeTo()
   - Cambia el tamaño de la ventana a dimensiones específicas.
   Ejemplo: window.resizeTo(800, 600);

10. window.resizeBy()
    - Cambia el tamaño de la ventana en relación a su tamaño actual.
    Ejemplo: window.resizeBy(100, 0);

11. window.focus()
    - Da el foco a la ventana actual.
    Ejemplo: window.focus();

12. window.blur()
    - Quita el foco de la ventana actual.
    Ejemplo: window.blur();

13. window.open()
    - Abre una nueva ventana del navegador.
    Ejemplo: window.open("https://www.ejemplo.com");

14. window.close()
    - Cierra la ventana actual.
    Ejemplo: window.close();

15. window.alert()
   - Muestra un cuadro de diálogo de alerta con un mensaje.
   Ejemplo: window.alert("¡Hola, mundo!");

16. window.confirm()
   - Muestra un cuadro de diálogo de confirmación con opciones "Aceptar" y "Cancelar".
   Ejemplo: const confirmado = window.confirm("¿Estás seguro?");

17. window.prompt()
   - Muestra un cuadro de diálogo de entrada de texto.
   Ejemplo: const nombre = window.prompt("Ingresa tu nombre:");

18. window.scrollTo()
   - Desplaza la ventana a una posición específica en la página.
   Ejemplo: window.scrollTo(0, 500);

19. window.scrollBy()
   - Desplaza la ventana en relación a su posición actual.
   Ejemplo: window.scrollBy(0, 100);

20. window.resizeBy()
   - Cambia el tamaño de la ventana en relación a su tamaño actual.
   Ejemplo: window.resizeBy(100, 0);

Propiedades

// Propiedades:
1. window.innerHeight
  - Devuelve la altura del área de contenido de la ventana en píxeles.

2. window.innerWidth
  - Devuelve el ancho del área de contenido de la ventana en píxeles.

3. window.outerHeight
  - Devuelve la altura de la ventana exterior, incluyendo barras de herramientas y marcos.

4. window.outerWidth
  - Devuelve el ancho de la ventana exterior, incluyendo barras de herramientas y marcos.

5. window.screenX
  - Devuelve la posición horizontal de la ventana en relación a la pantalla.

6. window.screenY
  - Devuelve la posición vertical de la ventana en relación a la pantalla.

7. window.scrollX
  - Devuelve la posición horizontal de la ventana desplazada.

8. window.scrollY
   - Devuelve la posición vertical de la ventana desplazada.

9. window.pageXOffset
   - Devuelve la cantidad de desplazamiento horizontal de la ventana, equivalente a `window.scrollX`.

10. window.pageYOffset
    - Devuelve la cantidad de desplazamiento vertical de la ventana, equivalente a `window.scrollY`.

11. window.location
    - Proporciona información sobre la ubicación actual del documento y permite navegar a otras páginas.

12. window.parent
    - Devuelve la ventana principal que contiene la ventana actual.

13. window.top
    - Devuelve la ventana superior en la jerarquía de ventanas.

14. window.self
    - Hace referencia a la ventana actual.

15. window.document
    - Devuelve el objeto `Document` asociado a la ventana.

16. window.history
    - Proporciona acceso al historial de navegación del navegador.

17. window.frames
    - Devuelve una colección de los objetos `Window` en el contexto de la ventana actual.

18. window.opener
    - Devuelve la ventana que abrió la ventana actual (si existe).

19. window.navigator
    - Proporciona información sobre el navegador del usuario.

20. window.sessionStorage
    - Proporciona acceso al almacenamiento de sesión del navegador.

21. window.localStorage
    - Proporciona acceso al almacenamiento local del navegador.

22. window.console
    - Proporciona acceso a la consola del navegador para la salida de mensajes y errores.

23. window.status
    - Obtiene o establece el texto que se muestra en la barra de estado del navegador.

24. window.name
    - Obtiene o establece el nombre de la ventana actual.

Objeto Document - Métodos

// Métodos de Document:
1. document.getElementById()
   - Devuelve una referencia al primer elemento con el valor del atributo "id" especificado.
   Ejemplo: const elemento = document.getElementById("miId");

2. document.getElementsByTagName()
   - Devuelve una colección de elementos con el nombre de la etiqueta HTML especificada.
   Ejemplo: const elementos = document.getElementsByTagName("p");

3. document.getElementsByClassName()
   - Devuelve una colección de elementos con la clase CSS especificada.
   Ejemplo: const elementos = document.getElementsByClassName("miClase");

4. document.querySelector()
   - Devuelve el primer elemento que coincida con el selector CSS especificado.
   Ejemplo: const elemento = document.querySelector("#miId .miClase");

5. document.querySelectorAll()
   - Devuelve todos los elementos que coincidan con el selector CSS especificado en una colección.
   Ejemplo: const elementos = document.querySelectorAll("p.miclase");

6. document.createElement()
   - Crea un nuevo elemento HTML con el nombre de etiqueta especificado.
   Ejemplo: const nuevoElemento = document.createElement("div");

7. document.createTextNode()
   - Crea un nuevo nodo de texto con el contenido de texto especificado.
   Ejemplo: const texto = document.createTextNode("Hola, mundo");

8. document.appendChild()
   - Agrega un nodo al final de la lista de hijos de un elemento.
   Ejemplo: padre.appendChild(hijo);

9. document.removeChild()
   - Elimina un nodo hijo de un elemento.
   Ejemplo: padre.removeChild(hijo);

10. document.addEventListener()
    - Registra un controlador de eventos en el elemento para escuchar eventos específicos.
    Ejemplo: elemento.addEventListener("click", miFuncion);

Propiedades

// Propiedades de Document:
1. document.title
   - Devuelve o establece el título del documento.
   Ejemplo: const titulo = document.title;

2. document.URL
   - Devuelve la URL completa del documento.
   Ejemplo: const url = document.URL;

3. document.domain
   - Devuelve o establece el dominio del documento.
   Ejemplo: const dominio = document.domain;

4. document.referrer
   - Devuelve la URL del documento que llevó al usuario al documento actual.
   Ejemplo: const referencia = document.referrer;

5. document.lastModified
   - Devuelve la fecha de la última modificación del documento.
   Ejemplo: const modificacion = document.lastModified;

6. document.characterSet
   - Devuelve la codificación de caracteres utilizada en el documento.
   Ejemplo: const codificacion = document.characterSet;

7. document.documentElement
   - Devuelve el elemento raíz (normalmente `<html>`) del documento.
   Ejemplo: const raiz = document.documentElement;

8. document.body
   - Devuelve el elemento `<body>` del documento.
   Ejemplo: const cuerpo = document.body;

9. document.head
   - Devuelve el elemento `<head>` del documento.
   Ejemplo: const cabeza = document.head;

10. document.images
    - Devuelve una colección de todos los elementos `<img>` en el documento.
    Ejemplo: const imagenes = document.images;

11. document.links
   - Devuelve una colección de todos los elementos `<a>` en el documento.
   Ejemplo: const enlaces = document.links;

12. document.forms
   - Devuelve una colección de todos los elementos `<form>` en el documento.
   Ejemplo: const formularios = document.forms;

13. document.scripts
   - Devuelve una colección de todos los elementos `<script>` en el documento.
   Ejemplo: const scripts = document.scripts;

14. document.styleSheets
   - Devuelve una colección de hojas de estilo CSS en el documento.
   Ejemplo: const hojasDeEstilo = document.styleSheets;

15. document.activeElement
   - Devuelve el elemento actualmente enfocado en el documento.
   Ejemplo: const enfocado = document.activeElement;

16. document.designMode
   - Establece o devuelve si el documento es editable ("on" para editable, "off" para no editable).
   Ejemplo: const modoEdicion = document.designMode;

17. document.cookie
   - Establece o devuelve las cookies asociadas con el documento.
   Ejemplo: const cookies = document.cookie;

18. document.defaultView
   - Devuelve la vista por defecto asociada con el documento.
   Ejemplo: const vista = document.defaultView;

19. document.location
   - Devuelve un objeto `Location` que proporciona información sobre la URL del documento.
   Ejemplo: const ubicacion = document.location;

20. document.readyState
   - Devuelve el estado de carga del documento (complete cuando está completamente cargado).
   Ejemplo: const estadoCarga = document.readyState;

21. document.charset
   - Obtiene o establece la codificación de caracteres del documento.
   Ejemplo: const codificacion = document.charset;