

JavaScript Programming for Web Applications

Clases o Métodos	Descripción	Ejemplo
<code>appendChild()</code>	Un método HTML DOM que, después de crear un elemento, puede utilizar esta función para colocar el elemento en la ubicación adecuada dentro del documento. El elemento que se va a anexar es el único parámetro.	<pre>//Creates the element <p> and text "Hello World". Appends Hello World <p> to the HTML document. <head> <script> function addPara() { var newPara = document.createElement("p"); var newText = document.createTextNode("Hello World!"); newPara.appendChild(newText); document.body.appendChild(newPara); } </script> </head> <body onload="addPara()"> </body></pre>
Arrays	Se crea declarando los elementos de la matriz en []. Un array se puede asignar a una variable, normalmente usando la palabra clave <code>const</code> o <code>var</code> . Las matrices utilizan la indexación basada en cero para acceder a sus elementos.	<pre>const Beatles = ["Ringo", "Paul", "George", "John"]; //Here Beatles[0] is "Ringo".</pre>
<code>Date()</code>	El constructor es <code>new Date([parámetros opcionales])</code> . Si el constructor se declara sin parámetros, devuelve la fecha y hora locales actuales. Se pueden crear nuevas fechas pasando parámetros a la nueva función <code>Date</code> .	<pre>//create a new date from a string var newDate = new Date("2021-1-17 13:15:30"); //create a new date instance representing 17 Jan 2021 00:00:00 //note that the month number is zero-based var newDate = new Date(2021, 0, 17);</pre>
<code>document.createElement()</code>	Toma un parámetro de nombre de etiqueta y crea un elemento con ese nombre. Puede colocar el elemento en otro lugar de la página usando funciones como <code>insertBefore()</code> , <code>appendChild()</code> , <code>replaceChild()</code> .	<pre>//Creates the element <p> and text "Hello World". Appends Hello World <p> to the HTML document. <head> <script> function addPara() { var newPara = document.createElement("p"); var newText = document.createTextNode("Hello World!"); newPara.appendChild(newText); document.body.appendChild(newPara); } </script> </head></pre>

		<pre><body onload="addPara()"> </body></pre>
document.createTextNode()	Toma una cadena como texto de entrada y devuelve un nodo de texto con el texto de entrada.	<pre>//Creates the element <p> and text "Hello World". Appends Hello World <p> to the HTML document. <head> <script> function addPara() { var newPara = document.createElement("p"); var newText = document.createTextNode("Hello World!"); newPara.appendChild(newText); document.body.appendChild(newPara); } </script> </head> <body onload="addPara()"> </body></pre>
document.getElementById()	Un método del DOM que toma un parámetro de valor de ID y devuelve un elemento que coincide con el id.	<pre>//Changes the content of the div to "Hello World!" <div id="div1"> <p>Hello</p> <p>Hello</p> </div> <script> document.getElementById("div1").innerHTML = "<p>Hello World!</p>"; </script></pre>
document.getElementsByTagName()	Método del DOM que toma un parámetro de nombre de etiqueta y devuelve una matriz denominada "NodeList" que contiene elementos con el nombre de etiqueta especificado.	<pre>//Gets an array of all elements in a document with the <p> tag. var tagNameArray = document.getElementsByTagName("p");</pre>
document.write()	Escribe HTML o JavaScript en un documento. Tenga en cuenta que sobrescribe cualquier otro texto en el documento, por lo que se usa principalmente solo con fines de prueba.	<pre>//Writes "Hello World" to the output stream. document.write("Hello World");</pre>
element.removeAttribute()	Propiedad de la clase Element que quita todos los estilos CSS en línea establecidos previamente para un elemento determinado. Toma un parámetro: el nombre del	<pre>//Removes the CSS style color blue <div id="div1" style="color: blue"></div> <script> var div1 = document.getElementById("div1").getAttribute("sty le"); </script></pre>

	atributo que se va a quitar	
<code>element.getAttribute()</code>	Devuelve el valor del atributo especificado. Toma un parámetro: el nombre del atributo cuyo valor se va a devolver.	<pre>//Removes the CSS style color blue <div id="div1" style="color: blue"></div> <script> var div1 = document.getElementById("div1").getAttribute("sty le"); </script></pre>
<code>element.setAttribute()</code>	Propiedad de la clase Element que sobrescribe todos los estilos CSS en línea establecidos previamente para un elemento determinado. Toma dos parámetros: el nombre del atributo que se está estableciendo y el valor del atributo en el que se establece el atributo.	<pre>//In all elements named "thelimage" sets the name of all src attributes to "another.gif" document.getElementById("thelimage").setAttribut e("src", "another.gif");</pre>
<code>element.style()</code>	Propiedad de la clase Element que devuelve o modifica CSS en línea. La sintaxis es <code>element.style.propertyName = value</code>	<pre>//Changes the CSS style color from blue to red <div id="div1" style="color: blue"></div> <script> var div1 = document.getElementById("div1"); div1.style.color = "red"; </script></pre>
Error Objects	La instancia crea dos propiedades sobre el error: un mensaje que contiene una descripción del error y la propiedad <code>name</code> que identifica el tipo de error. Error genérico más otros 6 errores principales: <code>TypeError</code> , <code>RangeError</code> , <code>URIError</code> , <code>EvalError</code> , <code>ReferenceError</code> , <code>SyntaxError</code> . Error se puede ampliar para crear mensajes de error personalizados mediante la palabra clave <code>throw</code> .	<pre>//Catch statement defines a block of code to be executed if an error occurs in the try block. catch (err) { document.getElementById("myfile").innerHTML = err.name; } //Creates custom error message throw new Error("Only values 1-10 are permitted");</pre>
<code>insertBefore()</code>	Método HTML DOM que, después de crear un elemento, coloca un elemento secundario en la ubicación adecuada antes de un elemento secundario existente. El	<pre>//Creates a new element and places it in the elementList before the first child of let newLI = document.createElement("li"); newLI.innerText = "new Element"; let elementList = document.getElementById("thisList");</pre>

	método toma dos parámetros, el objeto de nodo que se va a insertar y el nodo existente que se va a insertar antes.	<code>elementList.insertBefore(newLI, elementList.childNodes[0]);</code>
Location Objects	El objeto location forma parte del objeto window y contiene información sobre la dirección URL actual.	<code>//Returns the hostname property let myhost = location.hostname; newLI.innerText = "new Element";</code>
Navigator Objects	El objeto navigator es parte de la clase de objeto de ventana en el DOM que representa el navegador de Internet del cliente, también llamado agente de usuario. No hay un estándar para este objeto, por lo que lo que devuelve difiere de un navegador a otro.	<code>//Retrieves the name of the browser var browsername = navigator.appName;</code>
<code>replaceChild()</code>	Después de crear un elemento, esta función reemplaza un nodo secundario por un nuevo nodo.	<code>//Creates a new node and replaces the second element in "thisList" with the word "blue" let secondBullet = document.createTextNode("blue"); var myList = document.getElementById("thisList").childNodes[1] ; myList.replaceChild(secondBullet, myList.childNodes[1]);</code>
Screen Objects	El objeto screen es parte de la clase de objeto window en el DOM que se puede usar para devolver propiedades sobre la pantalla del usuario.	<code>//Returns the height and width of the user's screen var height=screen.height; var width=screen.width;</code>
<code>onload()</code>	Un evento DOM que inicia un método cuando se carga una página.	<code>//Executes myFunction after MyHTMLPage has been loaded document.getElementById("MyHTMLPage").onload = function () {myFunction};</code>
Window Objects	El objeto de ventana DOM se encuentra en la parte superior de la jerarquía DOM y actúa como objeto global. Todo en el DOM tiene lugar en una ventana. El objeto window controla el entorno que contiene el documento.	<code>//Opens a new browser window with the specified URL window.open("http://www.w3schools.com");</code>

Wrapper Objects	Los tipos primitivos se pueden convertir en objetos mediante objetos contenedores. Son el mismo nombre que la primitiva, excepto que comienzan con una letra mayúscula. La palabra clave devuelve una cadena que indica el tipo de datos del operando.typeof	//Enables the use of properties and methods of the String class such as the property n.length let n = new String ("abc"); //Returns string typeof "abc"; //Returns object typeof new String("abc")
window.scrollTo()	Se desplaza a un lugar determinado de una ventana. Los parámetros incluyen la coordenada x, que es el píxel situado más a la izquierda, y la coordenada y, que es el píxel superior.	//Scrolls the window to the pixel located at the coordinate (20, 200) window.scrollTo(20, 200);
window.open()	Se abre en una ventana nueva. El primer parámetro es una ruta de acceso, una dirección URL o una cadena vacía, y los parámetros opcionales incluyen el nombre de la ventana, características como la ubicación de la ventana o las dimensiones, y un valor de reemplazo booleano. El parámetro feature es una cadena separada por comas de pares nombre-valor y el parámetro replace es un valor booleano opcional. Este parámetro ha quedado en desuso, por lo que es posible que los navegadores modernos no lo admitan. Este método devuelve una referencia al objeto de ventana nueva.	//Opens a new window that opens the IBM home page and has a width of 600 and a height of 800 let thisWindow = window.open("http://www.ibm.com", "myWindow", "width"=600, "height"=800);
History Objects	El objeto history es parte del objeto window y contiene las URL visitadas por el usuario dentro de una ventana del navegador. Expone	//Go back two pages if the history exists in the history list. history.go(-2);

	métodos y propiedades útiles que le permiten navegar hacia adelante y hacia atrás por el historial del usuario y manipular el contenido de la pila de historial.	
--	---	--