

# MÉTODOS INTERESANTES API DOM

Métodos de búsqueda	Descripción
<b>ELEMENT</b> <code>.getElementById(id)</code>	Busca el elemento HTML con el id <code>id</code> . Si no, devuelve <code>NULL</code> .
<b>ARRAY</b> <code>.getElementsByName(class)</code>	Busca elementos con la clase <code>class</code> . Si no, devuelve <code>[]</code> .
<b>ARRAY</b> <code>.getElementsByName(name)</code>	Busca elementos con atributo name <code>name</code> . Si no, devuelve <code>[]</code> .
<b>ARRAY</b> <code>.getElementsByTagName(tag)</code>	Busca elementos <code>tag</code> . Si no encuentra ninguno, devuelve <code>[]</code> .
<b>ELEMENT</b> <code>.querySelector(sel)</code>	Busca el primer elemento que coincide con el selector CSS <code>sel</code> . Si no, <code>NULL</code> .
<b>ARRAY</b> <code>.querySelectorAll(sel)</code>	Busca todos los elementos que coinciden con el selector CSS <code>sel</code> . Si no, <code>[]</code> .

## Selectores CSS:

- Universal
- De tipo
- Clases
- Identificador
- Por atributo
- Descendiente
- Pseudo-clases

```

/* Selector Universal */
* {
    background-color: grey;
}
/* Selector de tipo */
p {
    background-color: red;
}
/* Selector de clase */
.clases {
    background-color: green;
}
/* Selector por identificador */
#cuadrotexto {
    background-color: blueviolet;
}
/* Selector por atributo */
[atributo="estiloatributo"] {
    background-color: chartreuse;
}

```

```

/* Selector por dependiente */
h4 b {
    background-color: darkgoldenrod;
}
h4 p b {
    background-color: rgb(72, 174, 187);
}
.clases strong {
    background-color: ghostwhite;
}
/* Selector por pseudo-clases */
a:hover {
    background-color: gold;
}

```

## Creación de elementos HTML

Métodos	Descripción
<b>ELEMENT</b> <code>.createElement(tag, options)</code>	Crea y devuelve el elemento HTML definido por el <b>STRING</b> <code>tag</code> .
<b>NODE</b> <code>.createComment(text)</code>	Crea y devuelve un nodo de comentarios HTML <code>&lt;!-- text --&gt;</code> .
<b>NODE</b> <code>.createTextNode(text)</code>	Crea y devuelve un nodo HTML con el texto <code>text</code> .
<b>NODE</b> <code>.cloneNode(deep)</code>	Clona el nodo HTML y devuelve una copia. <code>deep</code> es <code>false</code> por defecto.
<b>BOOLEAN</b> <code>.isConnected</code>	Indica si el nodo HTML está insertado en el documento HTML.

## Manejo de atributos

Métodos	Descripción
<b>BOOLEAN</b> <code>hasAttributes()</code>	Indica si el elemento tiene atributos HTML.
<b>BOOLEAN</b> <code>hasAttribute(attr)</code>	Indica si el elemento tiene el atributo HTML <code>attr</code> .
<b>ARRAY</b> <code>getAttributeNames()</code>	Devuelve un <b>ARRAY</b> con los atributos del elemento.
<b>STRING</b> <code>getAttribute(attr)</code>	Devuelve el valor del atributo <code>attr</code> del elemento o <b>NULL</b> si no existe.
<b>UNDEFINED</b> <code>removeAttribute(attr)</code>	Elimina el atributo <code>attr</code> del elemento.
<b>UNDEFINED</b> <code>setAttribute(attr, value)</code>	Añade o cambia el atributo <code>attr</code> al valor <code>value</code> .
<b>NODE</b> <code>getAttributeNode(attr)</code>	Idem a <code>getAttribute()</code> pero devuelve el atributo como <b>nodo</b> .
<b>NODE</b> <code>removeAttributeNode(attr)</code>	Idem a <code>removeAttribute()</code> pero devuelve el atributo como <b>nodo</b> .
<b>NODE</b> <code>setAttributeNode(attr, value)</code>	Idem a <code>setAttribute()</code> pero devuelve el atributo como <b>nodo</b> .

## Uso de fragmentos

Métodos	Descripción
<b>OBJECT</b> <code>document.createDocumentFragment()</code>	Crea un fragmento aislado ( <i>sin padre</i> ).

## Manejo de Contenido del DOM

Propiedades	Descripción
<b>STRING</b> <code>.nodeName</code>	Devuelve el nombre del nodo (etiqueta si es un elemento HTML). Sólo lectura.
<b>STRING</b> <code>.textContent</code>	Devuelve el contenido de texto del elemento. Se puede asignar para modificar.
<b>STRING</b> <code>.innerHTML</code>	Devuelve el contenido HTML del elemento. Se puede usar asignar para modificar.
<b>STRING</b> <code>.outerHTML</code>	Idem a <code>.innerHTML</code> pero incluyendo el HTML del propio elemento HTML.
<b>STRING</b> <code>.innerText</code>	Versión no estándar de <code>.textContent</code> de Internet Explorer con diferencias. <b>Evitar</b> .
<b>STRING</b> <code>.outerText</code>	Versión no estándar de <code>.textContent/.innerHTML</code> de Internet Explorer. <b>Evitar</b> .
<b>NODE</b> <code>.appendChild(node)</code>	Añade como hijo el nodo <code>node</code> . Devuelve el nodo insertado.
<b>ELEMENT</b> <code>.insertAdjacentElement(pos, elem)</code>	Inserta el elemento <code>elem</code> en la posición <code>pos</code> . Si falla, <b>NULL</b> .
<b>UNDEFINED</b> <code>.insertAdjacentHTML(pos, str)</code>	Inserta el código HTML <code>str</code> en la posición <code>pos</code> .
<b>UNDEFINED</b> <code>.insertAdjacentText(pos, text)</code>	Inserta el texto <code>text</code> en la posición <code>pos</code> .
<b>NODE</b> <code>.insertBefore(new, node)</code>	Inserta el nodo <code>new</code> antes de <code>node</code> y como hijo del nodo actual.
<b>UNDEFINED</b> <code>.remove()</code>	Elimina el propio nodo de su elemento padre.
<b>NODE</b> <code>.removeChild(node)</code>	Elimina y devuelve el nodo hijo <code>node</code> .
<b>NODE</b> <code>.replaceChild(new, old)</code>	Reemplaza el nodo hijo <code>old</code> por <code>new</code> . Devuelve <code>old</code> .

## Métodos de clases

Propiedad	Descripción
<b>STRING</b> <code>.className</code>	Acceso directo al valor del atributo HTML <b>class</b> . También se puede asignar.
<b>OBJECT</b> <code>.classList</code>	Objeto especial para manejar clases CSS. Contiene métodos y propiedades de ayuda.
<b>ARRAY</b> <code>.classList</code>	Devuelve la lista de clases del elemento HTML.
<b>NUMBER</b> <code>.classList.length</code>	Devuelve el número de clases del elemento HTML.
<b>STRING</b> <code>.classList.item(n)</code>	Devuelve la clase número <b>n</b> del elemento HTML. <b>NULL</b> si no existe.
<b>UNDEFINED</b> <code>.classList.add(c1, c2, ...)</code>	Añade las clases <b>c1</b> , <b>c2</b> ... al elemento HTML.
<b>UNDEFINED</b> <code>.classList.remove(c1, c2, ...)</code>	Elimina las clases <b>c1</b> , <b>c2</b> ... del elemento HTML.
<b>BOOLEAN</b> <code>.classList.contains(clase)</code>	Indica si la <b>clase</b> existe en el elemento HTML.
<b>BOOLEAN</b> <code>.classList.toggle(clase)</code>	Si la <b>clase</b> no existe, la añade. Si no, la elimina.
<b>BOOLEAN</b> <code>.classList.toggle(clase, expr)</code>	Si <b>expr</b> es <b>true</b> , añade <b>clase</b> . Si no, la elimina.
<b>BOOLEAN</b> <code>.classList.replace(old, new)</code>	Reemplaza la clase <b>old</b> por la clase <b>new</b> .

Documentación obtenida de : <https://lenguajejs.com/javascript/>

# LISTA DE ATRIBUTOS GLOBALES

Son aquellos que pueden aplicarse a todos los elementos del DOM

ACCESSKEY	AUTOCAPITALIZE	AUTOCOMPLETE
CLASS	CONTENTEDITABLE	CONTEXTMENU
DIR	DRAGGABLE	ENTERKEYHINT
HIDDEN	INPUTMODE	ID
IS	ITEMID	ITEMPROP
ITEMREF	ITEMSCOPE	ITEMTYPE
LANG	NONCE	SLOT
SPELLCHECK	STYLE	TABINDEX
TITLE	TRANSLATE	DROPZONE

Ver página <https://www.htmlquick.com/es/reference/attributes.html>

# Lista de atributos input

Pueden ser modificados directamente desde el objeto javascript, sin necesidad de utilizar el método setAttribute.

- className
- value
- type
- accept
- form
- placeholder
- required

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="ejemplo_attribute_input.css">
  <title>Document</title>
</head>
<body>
  <input type="text" class="input-normal" value="Escriba aquí ...">
  <p></p>
  <input type="file" accept="" id="fichero" value="Introduzca fichero">
  <p></p>

  <form id="formulario">
    <input type="text" name="">
    <input type="submit" name="">
  </form>
  <!--Este submit de fuera del formulario no actualizaría el cuadro de texto del
formulario-->
  <!--a menos que establezcamos la propiedad form = "id del formulario en el que está el
input"-->
  <input type="submit" name="">
  <script src="ejemplo_attribute_input.js">    </script>
</body>
</html>
```

## Contenido del fichero ejemplo\_attribute\_input.js

```
const input=document.querySelector(".input-normal");
const inputFile=document.querySelector("#fichero");
console.log(input.className);
console.log(input.value);
document.write (input.type="number");<br></br>
//document.write (input.type="range");<br></br>
//document.write (input.type="color");<br></br>
inputFile.accept="img/png";
input.minLength="9";
```

## Contenido del archivo ejemplo\_attribute\_input.css

```
.input-normal {
background-color: rgb(21, 255, 0);
}
```