



Somos un **ecosistema** de desarrolladores de software

# Java Script DOM (Document Object Model)




```
<!-- _____ BEGIN NAVIGATION  
>  
">Home</a></li>  
.html">Home Events</a></li>  
nu.html">Multiple Column Men  
<a href="#" class="current"  
utton-header.html">Tall But  
logo.html">Image Logo</a></  
href="tall-logo.html">Ta  
f="#">Carousels</a>  
th-slider.html">Variat  
lider.html">Testimoni
```



# DOM

Gestionar atributos del DOM –  
¿Qué es un atributo HTML?

En HTML (Hypertext Markup Language), un "atributo" es una característica adicional que se agrega a una etiqueta para proporcionar información adicional sobre cómo debe ser tratado o presentado un elemento. Los atributos suelen constar de un nombre y un valor, separados por un signo igual (=) y rodeados por comillas (simples o dobles). Aquí hay un ejemplo simple:



```
<etiqueta atributo="valor">Contenido</etiqueta>
```

- <etiqueta> es el nombre de la etiqueta HTML.
- "atributo" es el nombre del atributo.
- "valor" es el valor asignado al atributo.

# DOM

Acceder a atributos HTML

una vez tenemos un elemento sobre el que vamos a crear algunos atributos, lo más sencillo es asignarle valores como propiedades de objetos

```
const element = document.querySelector("div"); // <div class="container"></div>

element.id = "page"; // <div id="page" class="container"></div>
element.style = "color: red"; // <div id="page" class="container" style="color: red"></div>
element.className = "data"; // <div id="page" class="data" style="color: red"></div>
```

# DOM

## Obtener atributos HTML

</Riwi>

Métodos	Descripción
BOOLEAN hasAttributes()	Indica si el elemento tiene atributos HTML.
BOOBLEAN hasAttribute(attr)	Indica si el elemento tiene el atributo HTML attr.
ARRAY getAttributeNames()	Devuelve un ARRAY con los atributos del elemento.
STRING getAttribute(attr)	Devuelve el valor del atributo attr del elemento o NULL si no existe.

# DOM

## Obtener atributos HTML

```
<button id="miBoton" data-number="42" disabled>Click me</button>

<script>
  // Obtener el elemento del botón
  const boton = document.getElementById("miBoton");

  // Verificar si el elemento tiene algún atributo
  const tieneAtributos = boton.hasAttributes();
  console.log("¿Tiene atributos?", tieneAtributos); // Imprimirá: true

  // Verificar si el elemento tiene el atributo "data-number"
  const tieneDataNumber = boton.hasAttribute("data-number");
  console.log("¿Tiene el atributo 'data-number'?", tieneDataNumber); // Imprimirá: true

  // Verificar si el elemento tiene el atributo "disabled"
  const tieneDisabled = boton.hasAttribute("disabled");
  console.log("¿Tiene el atributo 'disabled'?", tieneDisabled); // Imprimirá: true

  // Obtener una lista de todos los nombres de atributos
  const nombresAtributos = boton.getAttributeNames();
  console.log("Nombres de atributos:", nombresAtributos); // Imprimirá: ["id", "data-
number", "disabled"]

  // Obtener el valor del atributo "id"
  const valorId = boton.getAttribute("id");
  console.log("Valor del atributo 'id':", valorId); // Imprimirá: "miBoton"
</script>
```

# DOM

Modificar o eliminar atributos HTML

Métodos	Descripción
setAttribute(attr, value)	Añade o cambia el atributo attr al valor value del elemento HTML.
toggleAttribute(attr, force)	Añade atributo attr si no existe, si existe lo elimina.
removeAttribute(attr)	Elimina el atributo attr del elemento HTML.



# DOM

## Modificar o eliminar atributos HTML

```

<button id="miBoton" data-number="42" disabled>Click me</button>

<script>
  // Obtener el elemento del botón
  const boton = document.getElementById("miBoton"); // .querySelector("#")

  // Cambiar el valor del atributo "data-number" a 10
  boton.setAttribute("data-number", "10");

  // Eliminar el atributo "id"
  boton.removeAttribute("id");

  // Añadir nuevamente el atributo "id" con el valor "page"
  boton.setAttribute("id", "page");
</script>

```



# DOM

La API classList de Javascript – Manipular clases CSS de una etiqueta HTML

La propiedad .className

La propiedad .className viene a ser la modalidad directa y rápida de utilizar el getter .getAttribute("class") y el setter .setAttribute("class", value).

Propiedad	Descripción
STRING.className	Acceso directo al valor del atributo HTML class. También se puede asignar.
OBJECT .classList	Objeto especial para manejar clases CSS. Contiene métodos y propiedades de ayuda.

```
<div class="element shine dark-theme"></div>

const div = document.querySelector(".element");

// Obtener clases CSS
div.className;           // "element shine dark-theme"
div.getAttribute("class"); // "element shine dark-theme"

// Modificar clases CSS
div.className = "element shine light-theme";
div.setAttribute("class", "element shine light-theme");
```

# DOM

## .classList

Método	Descripción
Obtener información	
ARRAY.classList	Devuelve la lista de clases del elemento HTML.
NUMBER.classList.length	Devuelve el número de clases del elemento HTML.
STRING .classList.item(n)	Devuelve la clase número n del elemento HTML. si no existe.
BOOLEAN.classList.contains(clase)	Indica si la clase existe en el elemento HTML.
Acciones sobre clases	
.classList.add(c1, c2, ...)	Añade las clases c1, c2... al elemento HTML.
.classList.remove(c1, c2, ...)	Elimina las clases c1, c2... del elemento HTML.
BOOLEAN .classList.toggle(clase)	Si la clase no existe, la añade. Si no, la elimina.
BOOLEAN.classList.toggle(clase, expr)	Si expr es true, añade la clase. Si es false, la elimina.
BOOLEAN.classList.replace(old, new)	Reemplaza la clase old por la clase new.

# DOM

.classList

</Riwi>

```
<div id="page" class="info data dark" data-number="5"></div>
```

```
const element = document.querySelector("#page");
```

```
// ¿Qué clases tiene?
```

```
element.classList; // ["info", "data", "dark"]
```

```
(DOMTokenList)
```

```
element.classList.value; // "info data dark" (String)
```

```
element.classList.length; // 3
```

```
// Convertirlas a array
```

```
Array.from(element.classList) // ["info", "data", "dark"]
```

```
(Array)
```

```
[...element.classList]; // ["info", "data", "dark"]
```

```
(Array)
```

```
// Consultarlas
```

```
element.classList.item(0); // "info"
```

```
element.classList.item(1); // "data"
```

```
element.classList.item(3); // null
```

# DOM

.classList Añadir y eliminar clases CSS

```
<div id="page" class="info data dark" data-number="5"></div>

const element = document.querySelector("#page");

element.classList.add("uno", "dos");
element.classList; // ["info", "data", "dark", "uno", "dos"]

element.classList.remove("uno", "dos");
element.classList; // ["info", "data", "dark"]
```

# DOM

.classList comprobar si existen clases  
CSS

```
<div id="page" class="info data dark" data-number="5"></div>
```

```
const element = document.querySelector("#page");
```

```
element.classList; // ["info", "data", "dark"]  
element.classList.contains("info"); // Devuelve `true` (existe esa clase)  
element.classList.contains("warning"); // Devuelve `false` (no existe esa  
clase)
```

# DOM

.classList Conmutar o alternar clases CSS

```
<div id="page" class="info data dark" data-number="5"></div>
```

```
const element = document.querySelector("#page");
```

```
element.classList; // ["info", "data", "dark"]
```

```
element.classList.toggle("info"); // Como "info" existe, lo elimina. Devuelve  
"false"
```

```
element.classList; // ["data", "dark"]
```

```
element.classList.toggle("info"); // Como "info" no existe, lo añade. Devuelve  
"true"
```

```
element.classList; // ["info", "data", "dark"]
```

# DOM

.classList Reemplar una clase CSS

```
<div id="page" class="info data dark" data-number="5"></div>
```

```
const element = document.querySelector("#page");
```

```
element.classList; // ["info", "data", "dark"]
```

```
element.classList.replace("dark", "light"); // Devuelve `true` (se hizo  
el cambio)
```

```
element.classList.replace("warning", "error"); // Devuelve `false` (no  
existe warning)
```



# DOM

## Contenido en el DOM - Propiedades

Propiedades	Descripción
STRING .nodeName	Devuelve el nombre del nodo (etiqueta si es un elemento HTML). Sólo lectura.
Contenido de texto	
STRING.textContent	Devuelve el contenido de texto del elemento. Se puede asignar para modificar.
STRING.innerHTML	Versión no estándar de .textContent de Internet Explorer con diferencias. <b>Evitar</b> .
STRING.outerText	Versión no estándar de .textContent/.outerHTML de Internet Explorer. <b>Evitar</b> .
Contenido HTML	
STRING.innerHTML	Devuelve el contenido HTML del elemento. Se puede usar asignar para modificar.
STRING.outerHTML	Idem a .innerHTML pero incluyendo el HTML del propio elemento HTML.
.setHTML(htmlCode, options)	Método que inserta HTML, pero sanitizando la entrada de datos.

# DOM

.textContent

</Riwi>

```
<div class="container">
  <div class="parent">
    <p>Hola a todos.</p>
    <p class="message">Mi nombre es <strong>Mailet</strong>.</p>
  </div>
</div>
```

```
const element = document.querySelector(".message");
```

```
element.textContent; // "Mi nombre es Mailet."
```

```
element.textContent = "Hola a todos";
```

```
element.textContent; // "Hola a todos"
```

# DOM

.innerText y .outerText

</Riwi>

```
<div class="container">
  <p>Hola a todos.</p>
  <p>Me llamo <strong>Mailet</strong>. <mark
style="display: none">New message</mark></p>
  <p hidden>Esto es un mensaje posterior oculto
semánticamente.</p>
  <details>
    <summary>Más información</summary>
    <div>Esto es un desplegable que está colapsado.
</div>
  </details>
</div>
```

# DOM

.innerText y .outerText

```
const element = document.querySelector(".container");  
element.innerText;  
  
// "Hola a todos."  
//  
// Me llamo Mailet.  
//  
// Más información"
```

# DOM

.innerHTML

</Riwi>

```
const element = document.querySelector(".message");

element.innerHTML;    // "Mi nombre es
<strong>Mailet</strong>."
element.textContent;  // "Mi nombre es Mailet."
```

```
element.innerHTML = "<strong>Importante</strong>";
// Se lee "Importante" (en negrita)
element.textContent = "<strong>Importante</strong>";
// Se lee "<strong>Importante</strong>"
```

# DOM

.outerHTML

</Riwi>

```
const data = document.querySelector(".data");
data.innerHTML = "<h1>Tema 1</h1>";

data.textContent;    // "Tema 1"
data.innerHTML;      // "<h1>Tema 1</h1>"
data.outerHTML;      // "<div class='data'><h1>Tema
1</h1></div>"
```

</Be a  
coder>