

# {JS}

**Clase 22**



# ◀ Índice ▶

## Gestión de atributos del DOM

Repasemos los atributos

---

Acceso a atributos básicos

---

Acceso a cualquier atributo

---

Modificar atributos

---

Casos especiales

---

`toggleAttribute`

**Repasemos los atributos**

# Repasemos los atributos

```
<div class="container" data-attr="value">  
  <button disabled>Avisar</button>  
</div>
```

Atributo – Valor – Booleano

# Repasemos los atributos

Atributos básicos

Atributos específicos

Atributos para todas las etiquetas

Cuidado con el class: tiene sus propios métodos que ya veremos

# **Acceso a atributos básicos**

```
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>
```

**Acceso a cualquier atributo**



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.

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

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

element.hasAttributes();           // true (tiene 3 atributos)
element.hasAttribute("data-number"); // true (data-number existe)
element.hasAttribute("disabled");   // false (disabled no existe)

element.getAttributeNames();        // ["id", "data-number", "class"]
element.getAttribute("id");         // "page"
```

**Modificar atributos**

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

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

```
const element = document.querySelector("#page");  
  
element.setAttribute("data-number", "10"); // Cambiar data-number a 10  
element.removeAttribute("id");             // Elimina el atributo id  
element.setAttribute("id", "page");         // Vuelve a añadirlo
```

## **Casos especiales: booleanos**

Hay que hablar de un caso especial, que es el que comentamos en el que podemos establecer atributos HTML que son booleanos, es decir, que no tienen indicado ningún valor.

## **Casos especiales: booleanos**

Si esto lo hacemos con el método `setAttribute()` y le indicamos un booleano, no tendremos exactamente lo que buscamos. Recuerda que los atributos HTML son siempre de tipo string.

# Casos especiales: booleanos

```
const button = document.querySelector("button");  
  
button.setAttribute("disabled", true); // ✗ <button disabled="true">Clickme!</button>  
button.disabled = true;                // ✓ <button disabled>Clickme!</button>  
button.setAttribute("disabled", "");   // ✓ <button disabled>Clickme!</button>
```



## **Casos especiales: booleanos**

Recuerda que atributo HTML no es lo mismo que propiedad Javascript, aunque muchos frameworks o librerías Javascript simplifican para que parezcan que son la misma cosa.

**toggleAttribute**

Para atributos booleanos es más fácil usar:  
**toggleAttribute**

```
button.toggleAttribute("disabled");    // Como ya existe "disabled", lo elimina  
button.toggleAttribute("hidden");      // Como no existe "hidden", lo añade
```

**◀ Despedida ▶**

Email

**bienvenidosaez@gmail.com**

Instagram

**@bienvenidosaez**

Youtube

**youtube.com/bienvenidosaez**