

{JS}

Clase 23



◀ Índice ▶

API de classList y dataset

Gestión de clases

Gestión de data attributes

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

classList

classList

className y classList

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

classList

```
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");
```

classList

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. **NULL** 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`.

data attributes

classList

Algo parecido pasa con los data attributes
elemento.dataset

classList

Obtener atributos

```
// Usando getAttribute  
var valor = document.getElementById('miElemento').getAttribute('data-mi-dato');  
  
// Usando dataset  
var valor = document.getElementById('miElemento').dataset.miDato;
```


classList

Setear atributos

```
// Usando setAttribute
document.getElementById('miElemento').setAttribute('data-mi-dato', 'nuevoValor');

// Usando dataset
document.getElementById('miElemento').dataset.miDato = 'nuevoValor';
```

classList

Eliminar atributos

```
document.getElementById('miElemento').removeAttribute('data-mi-dato');
```

classList

Notas importantes:

- Los nombres de los data atributos se convierten a camelCase cuando se acceden a través de dataset. Por ejemplo, data-mi-dato se convierte en miDato en el dataset.
- Los data atributos son útiles para almacenar información adicional que no se utiliza para otros fines, como estilos o scripts. No deben utilizarse para almacenar información que debería estar visible o accesible a los usuarios o para otra funcionalidad que HTML ya proporciona por otros medios.

◀ Despedida ▶

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez