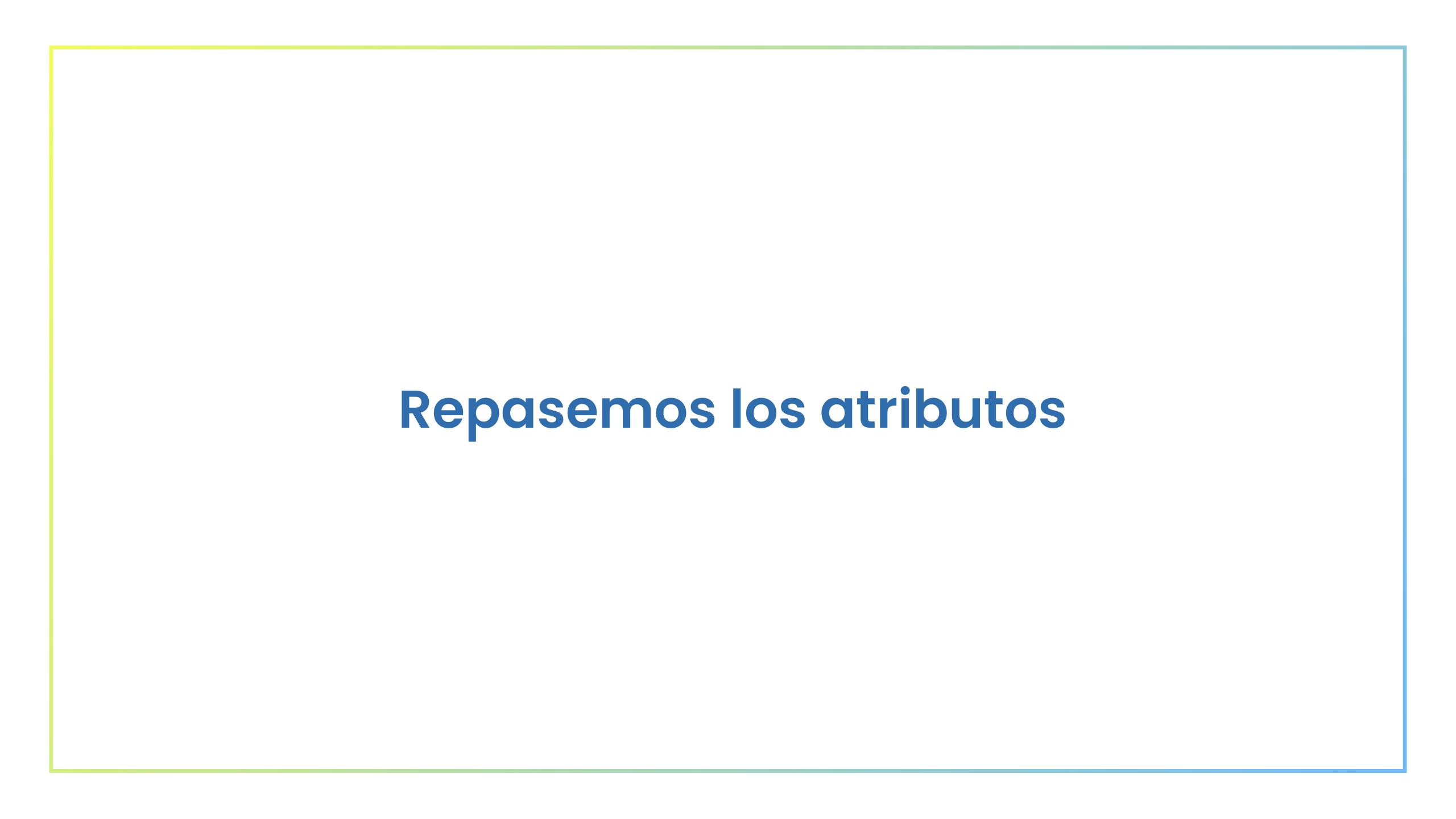


(Indice)

API de classList y dataset

Gestión de clases

Gestión de data attributes



Repasemos los atributos

Atributo - Valor - Booleano

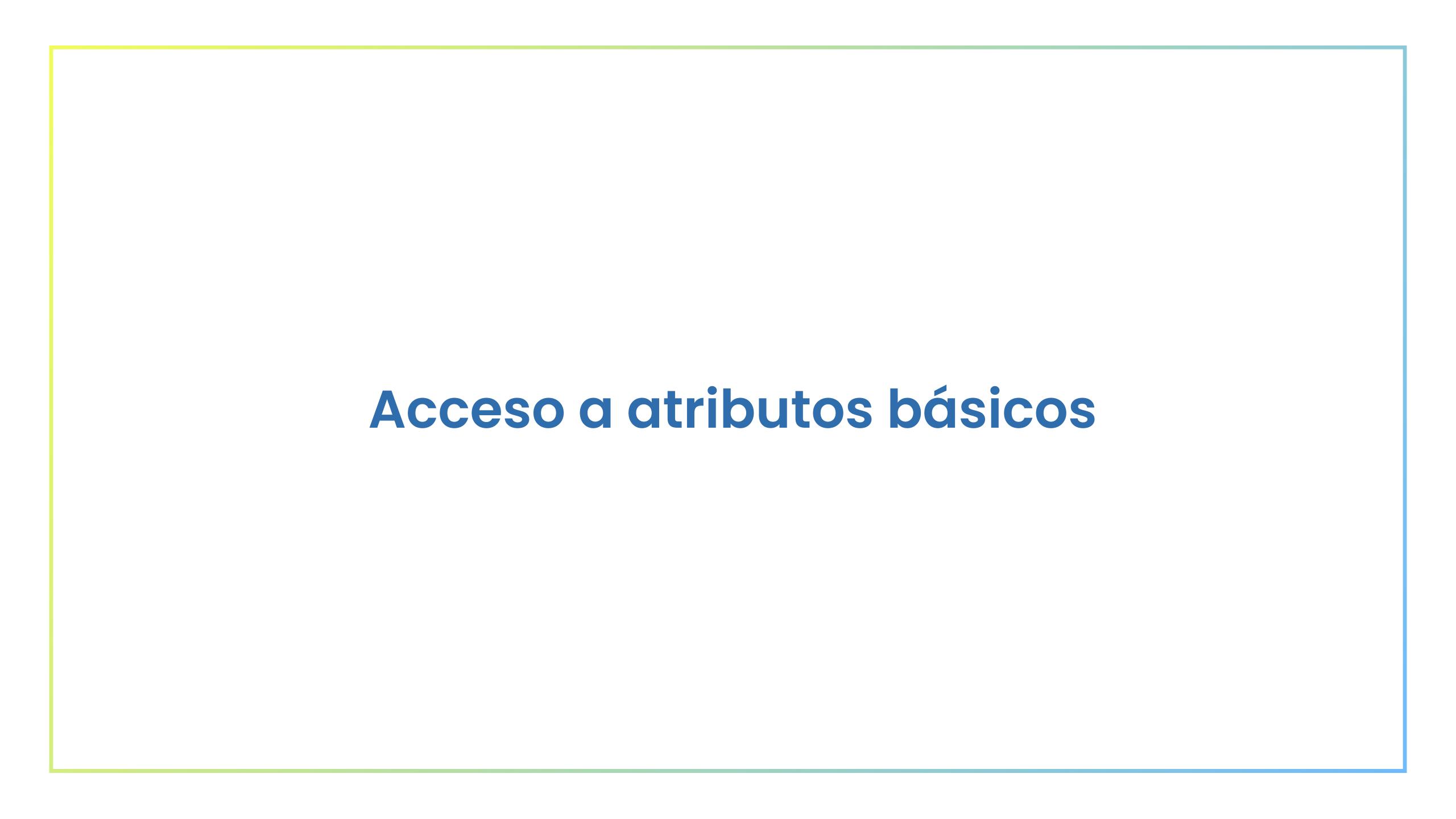
Repasemos los atributos

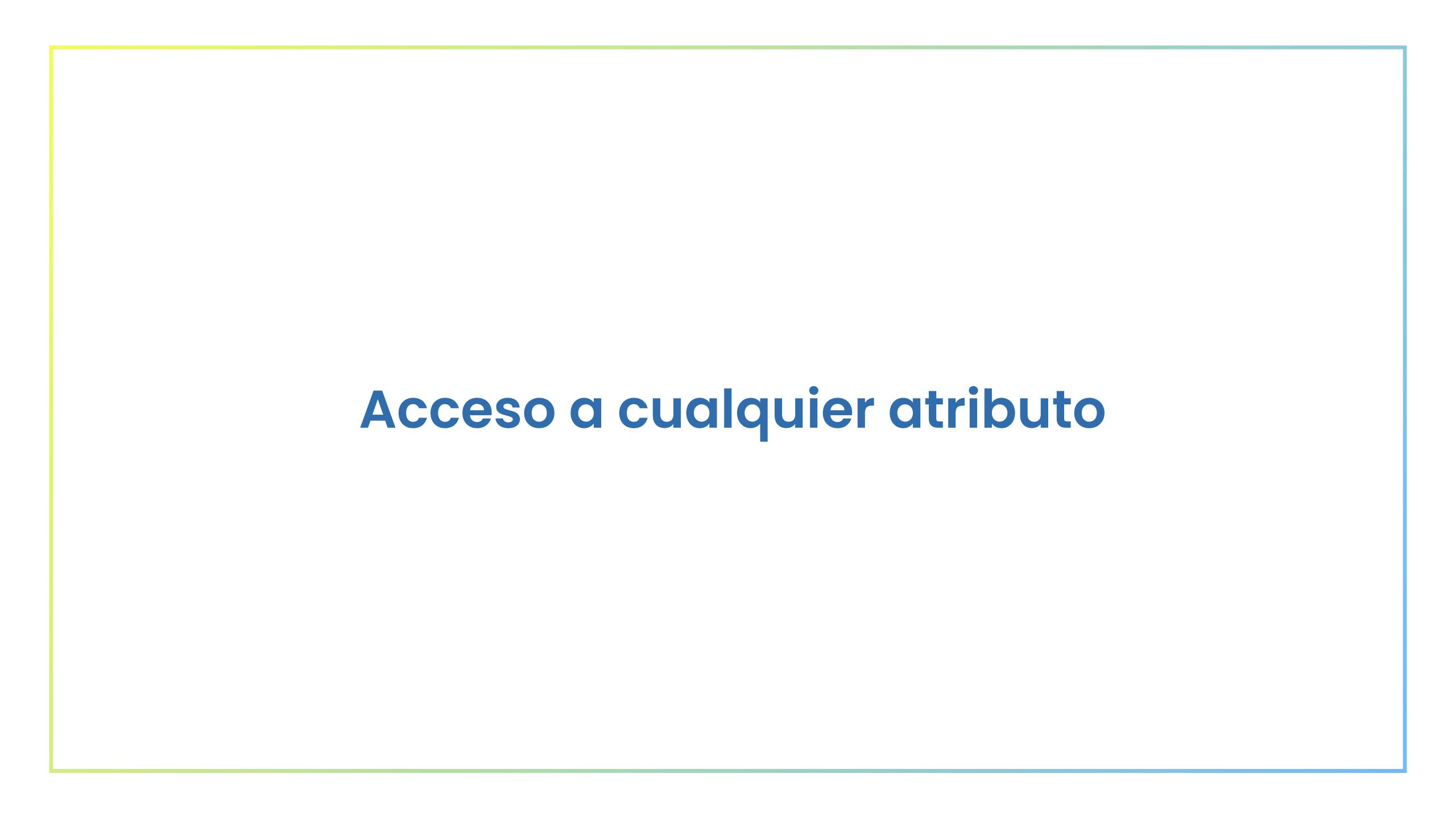
Atributos básicos

Atributos específicos

Atributos para todas las etiquetas

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





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



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).

```
const div = document.querySelector(".element");
// Obtener clases CSS
              // "element shine dark-theme"
div.className;
div.getAttribute("class"); // "element shine dark-theme"
// Modificar clases CSS
div.className = "element shine light-theme";
div.setAttribute("class", "element shine light-theme");
```

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 ⊓ 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.	



Algo parecido pasa con los data attributes elemento.dataset

Obtener atributos

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

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

Setear atributos

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

Eliminar atributos

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

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.

C Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez