

{JS}

Clase 24



◀ Índice ▶

Insertar y movernos por el DOM

Propiedades de un elemento

Insertar elementos en el DOM

Navegar por los elementos del DOM

Propiedades de un elemento

Propiedades de un elemento

`.textContent`

Sirve para obtener y asignar el contenido de texto de un elemento

Propiedades de un elemento

`.innerText`

Obtiene solo el texto visible después de renderizar el documento y aplicar los estilos CSS

Propiedades de un elemento

`.innerHTML`

Sirve para obtener o cambiar el contenido de una etiqueta en formato HTML. Esta propiedad si renderiza código HTML

Propiedades de un elemento

.innerHTML

Ten mucho cuidado a la hora de insertar contenido HTML utilizando .innerHTML puesto que si añades contenido que provenga del usuario sin revisarlo, podrían insertar HTML que realice acciones dañinas como inyección de código malicioso.

Propiedades de un elemento

.setHTML

“Sanitiza” previamente el contenido antes de introducirlo.

Experimental

Propiedades de un elemento

`.outerHTML`

Muy similar a `innerHTML` pero esta propiedad devuelve también la etiqueta sobre la que estamos ejecutando

Insertar elementos en el DOM

API de nodos

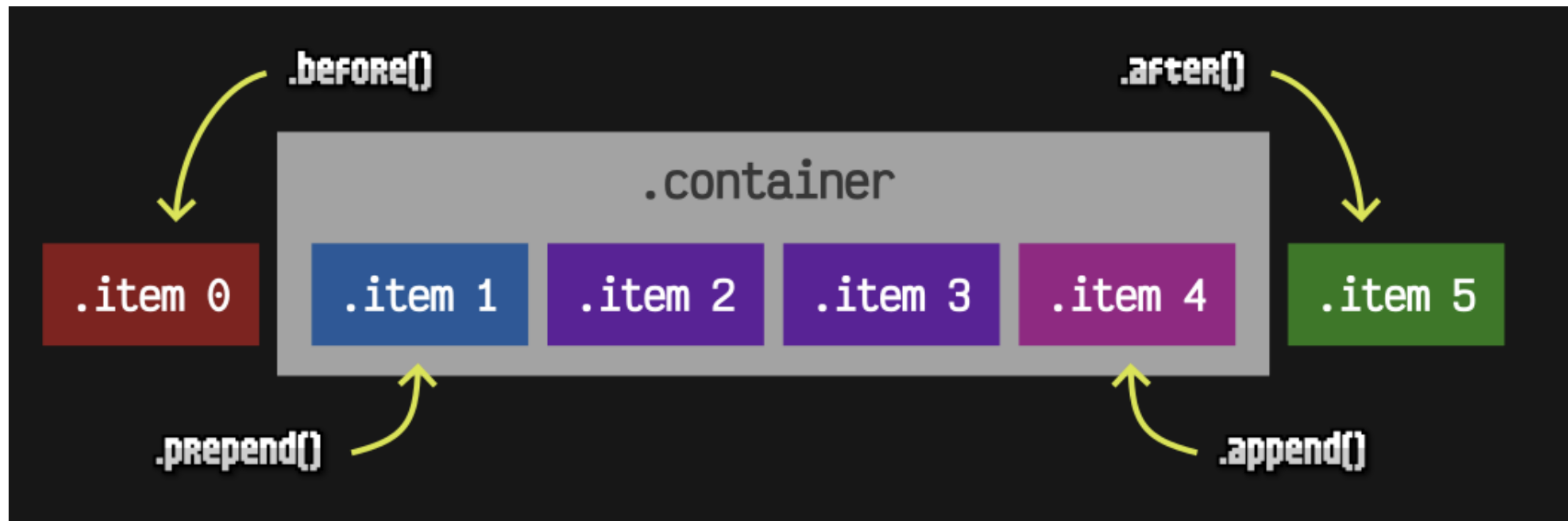
API de nodos

Métodos	Descripción
NODE .appendChild(node)	Añade como hijo el nodo <code>node</code> . Devuelve el nodo insertado.
NODE .removeChild(node)	Elimina y devuelve el nodo hijo <code>node</code> .
NODE .replaceChild(new, old)	Reemplaza el nodo hijo <code>old</code> por <code>new</code> . Devuelve <code>old</code> .
NODE .insertBefore(new, node)	Inserta el nodo <code>new</code> antes de <code>node</code> y como hijo del nodo actual.
NODE .insertBefore(new, NULL))	Inserta el nodo <code>new</code> después del último nodo hijo. Equivale a <code>.appendChild()</code> .

Insertar elementos en el DOM

API de elementos

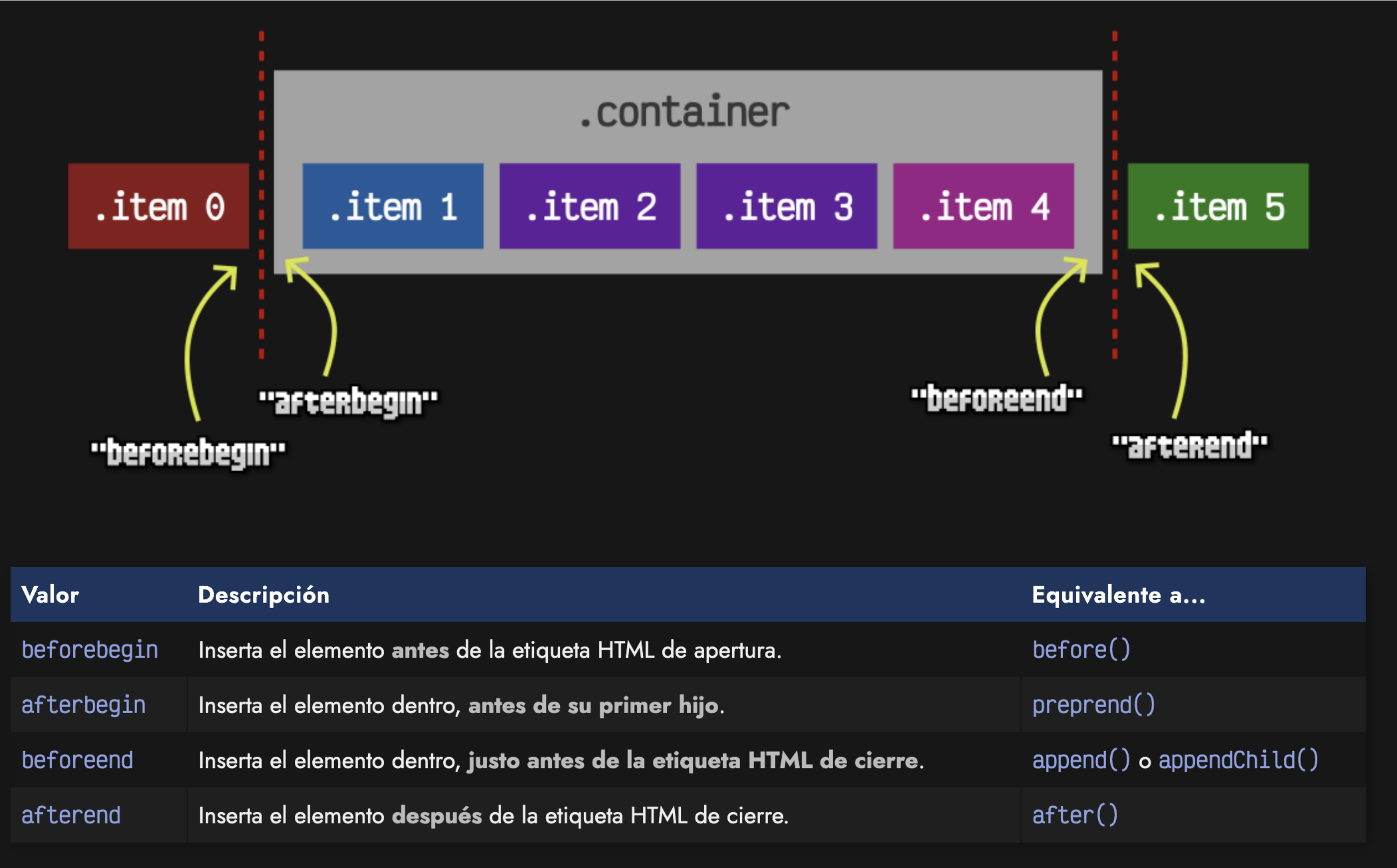
Métodos	Descripción
<code>.before()</code>	Añade el nuevo elemento justo antes.
<code>.after()</code>	Añade el nuevo elemento justo después.
<code>.prepend()</code>	Se añade el nuevo elemento antes del primer hijo.
<code>.append()</code>	Se añade el nuevo elemento después del último hijo.
<code>.replaceChildren()</code>	Elimina todos los hijos y los sustituye por el nuevo elemento.
<code>.replaceWith()</code>	Se sustituye por el nuevo elemento.
<code>.remove()</code>	Elimina el propio elemento.



Insertar elementos en el DOM

API de inserción adyacente

Métodos	Descripción
ELEMENT .insertAdjacentElement(position, element)	Inserta el <code>element</code> en la posición <code>position</code> . Si falla, <code>NULL</code> .
.insertAdjacentHTML(position, htmlCode)	Inserta el código HTML de <code>htmlCode</code> en la posición <code>position</code> .
.insertAdjacentText(position, text)	Inserta el texto <code>text</code> en la posición <code>position</code> .



```
const container = document.querySelector(".container");

// Creamos un nuevo <div>Ejemplo</div>
const div = document.createElement("div");
div.textContent = "Ejemplo";

container.insertAdjacentElement("beforebegin", div);
// A) <div>Ejemplo</div> <div class="container">container</div>

container.insertAdjacentElement("afterbegin", div);
// B) <div class="container"> <div>Ejemplo</div> container</div>

container.insertAdjacentElement("beforeend", div);
// C) <div class="container">container <div>Ejemplo</div> </div>

container.insertAdjacentElement("afterend", div);
// D) <div class="container">App</div> <div>Ejemplo</div>
```

Navegar por el DOM por elementos

Propiedades de elementos HTML	Descripción
ARRAY children	Devuelve una lista de elementos HTML hijos.
ELEMENT parentElement	Devuelve el padre del elemento o NULL si no tiene.
ELEMENT firstElementChild	Devuelve el primer elemento hijo.
ELEMENT lastElementChild	Devuelve el último elemento hijo.
ELEMENT previousElementSibling	Devuelve el elemento hermano anterior o NULL si no tiene.
ELEMENT nextElementSibling	Devuelve el elemento hermano siguiente o NULL si no tiene.

```
<html>
  <body>
    <div id="app">
      <div class="header">
        <h1>Titular</h1>
      </div>
      <p>Párrafo de descripción</p>
      <a href="/">Enlace</a>
    </div>
  </body>
</html>
```

```
document.body.children.length; // 1
document.body.children;        // <div id="app">
document.body.parentElement;   // <html>

const app = document.querySelector("#app");

app.children;                  // [div.header, p, a]
app.firstChild;                // <div class="header">
app.lastElementChild;          // <a href="/">

const a = app.querySelector("a");

a.previousElementSibling;      // <p>
a.nextElementSibling;          // null
```

Navegar por el DOM por nodos

Propiedades de nodos HTML	Descripción
ARRAY <code>childNodes</code>	Devuelve una lista de nodos hijos. Incluye nodos de texto y comentarios.
NODE <code>parentNode</code>	Devuelve el nodo padre del nodo o NULL si no tiene.
NODE <code>firstChild</code>	Devuelve el primer nodo hijo.
NODE <code>lastChild</code>	Devuelve el último nodo hijo.
NODE <code>previousSibling</code>	Devuelve el nodo hermano anterior o NULL si no tiene.
NODE <code>nextSibling</code>	Devuelve el nodo hermano siguiente o NULL si no tiene.


```
document.body.childNodes.length; // 3
document.body.childNodes;        // [text, div#app, text]
document.body.parentNode;        // <html>

const app = document.querySelector("#app");

app.childNodes;                  // [text, div.header, text, p, text, a, text]
app.firstChild.textContent;      // "
app.lastChild.textContent;       // "

const a = app.querySelector("a");

a.previousSibling;               // #text
a.nextSibling;                   // #text
```


Ejercicios

◀ Despedida ▶

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez