

Programa académico **CAMPUS**



Trainer
Ing. Carlos H. Rueda C.



Manipulación Avanzada del DOM mediante la Jerarquía de Elementos

Inserción de elementos en el DOM

Manipular elementos dentro del DOM implica crear, mover, modificar o eliminar elementos. Este proceso se realiza usando métodos específicos para trabajar con nodos y elementos.

Administración de Nodos

`.appendChild()`

Permite insertar un nodo como hijo de otro.

```
// Antes:  
// <div id="contenedor"></div>  
  
const nuevoElemento = document.createElement('p');  
nuevoElemento.textContent = 'Este es un nuevo párrafo';  
  
document.getElementById('contenedor').appendChild(nuevoElemento);  
  
// Después:  
// <div id="contenedor">  
//   <p>Este es un nuevo párrafo</p>  
// </div>
```

.removeChild()

Elimina un nodo hijo de un elemento padre.

```
// Antes:  
// <div id="contenedor">  
//   <div id="hijo">Elemento hijo</div>  
// </div>  
  
const contenedor = document.getElementById('contenedor');  
const hijo = document.getElementById('hijo');  
contenedor.removeChild(hijo);  
  
// Después:  
// <div id="contenedor"></div>
```

`.replaceChild()`

Reemplaza un nodo hijo por otro nodo.

```
// Antes:  
// <div id="contenedor">  
//   <div id="hijo">Elemento viejo</div>  
// </div>  
  
const nuevoNode = document.createElement('span');  
nuevoNode.textContent = 'Elemento nuevo';  
const contenedor = document.getElementById('contenedor');  
const viejoNode = document.getElementById('hijo');  
contenedor.replaceChild(nuevoNode, viejoNode);  
  
// Después:  
// <div id="contenedor">  
//   <span>Elemento nuevo</span>  
// </div>
```

`.insertBefore(new, node)`

Inserta un nodo antes de un nodo específico.

```
// Antes:  
// <div id="contenedor">  
//   <div id="referencia">Elemento referencia</div>  
// </div>  
  
const nuevoNodo = document.createElement('p');  
nuevoNodo.textContent = 'Insertado antes';  
const referencia = document.getElementById('referencia');  
const contenedor = referencia.parentNode;  
contenedor.insertBefore(nuevoNodo, referencia);  
  
// Después:  
// <div id="contenedor">  
//   <p>Insertado antes</p>  
//   <div id="referencia">Elemento referencia</div>  
// </div>
```

`.insertBefore(new, null)`

Inserta un nodo al final si se pasa `null` como referencia.

```
// Antes:  
// <div id="contenedor">  
//   <div>Elemento existente</div>  
// </div>  
  
const nuevoNodo = document.createElement('p');  
nuevoNodo.textContent = 'Insertado al final';  
const contenedor = document.getElementById('contenedor');  
contenedor.insertBefore(nuevoNodo, null);  
  
// Después:  
// <div id="contenedor">  
//   <div>Elemento existente</div>  
//   <p>Insertado al final</p>  
// </div>
```


Propiedades para modificación de elementos

`.before()`

Inserta contenido justo antes del elemento actual.

```
// Antes:  
// <div id="referencia">Elemento referencia</div>  
  
const referencia = document.getElementById('referencia');  
referencia.before('Texto antes de este elemento');  
  
// Después:  
// Texto antes de este elemento  
// <div id="referencia">Elemento referencia</div>
```

.after()

Inserta contenido justo después del elemento actual.

```
// Antes:  
// <div id="referencia">Elemento referencia</div>  
  
const referencia = document.getElementById('referencia');  
referencia.after('Texto después de este elemento');  
  
// Después:  
// <div id="referencia">Elemento referencia</div>  
// Texto después de este elemento
```

.prepend()

Inserta contenido al principio de un elemento.

```
// Antes:  
// <div id="contenedor">  
//   <p>Elemento existente</p>  
// </div>  
  
const contenedor = document.getElementById('contenedor');  
contenedor.prepend('Texto al inicio');  
  
// Después:  
// <div id="contenedor">  
//   Texto al inicio  
//   <p>Elemento existente</p>  
// </div>
```

.append()

Inserta contenido al final de un elemento.

```
// Antes:  
// <div id="contenedor">  
//   <p>Elemento existente</p>  
// </div>  
  
const contenedor = document.getElementById('contenedor');  
contenedor.append('Texto al final');  
  
// Después:  
// <div id="contenedor">  
//   <p>Elemento existente</p>  
//   Texto al final  
// </div>
```

`.remove()`

Elimina el elemento actual del DOM.

```
// Antes:  
// <div id="referencia">Elemento a eliminar</div>  
  
const referencia = document.getElementById('referencia');  
referencia.remove();  
  
// Después:  
// <!-- Contenido vacío -->
```

`.replaceChildren()`

Reemplaza todos los hijos de un elemento por nuevos nodos.

```
// Antes:  
// <div id="contenedor">  
//   <p>Hijo 1</p>  
//   <p>Hijo 2</p>  
// </div>  
  
const contenedor = document.getElementById('contenedor');  
const nuevoHijo = document.createElement('div');  
nuevoHijo.textContent = 'Hijo reemplazado';  
contenedor.replaceChildren(nuevoHijo);  
  
// Después:  
// <div id="contenedor">  
//   <div>Hijo reemplazado</div>  
// </div>
```

.replaceWith()

Reemplaza el elemento actual con nuevos elementos o nodos.

```
// Antes:  
// <div id="referencia">Elemento original</div>  
  
const referencia = document.getElementById('referencia');  
const nuevoElemento = document.createElement('span');  
nuevoElemento.textContent = 'Elemento reemplazado';  
referencia.replaceWith(nuevoElemento);  
  
// Después:  
// <span>Elemento reemplazado</span>
```

Propiedades de inserción adyacente

`.insertAdjacentElement()`

Inserta un elemento en una posición adyacente específica.

```
// Antes:  
// <div id="referencia">Elemento referencia</div>  
  
const referencia = document.getElementById('referencia');  
const nuevoElemento = document.createElement('span');  
nuevoElemento.textContent = 'Elemento adyacente';  
referencia.insertAdjacentElement('beforebegin', nuevoElemento);  
  
// Después:  
// <span>Elemento adyacente</span>  
// <div id="referencia">Elemento referencia</div>
```


.insertAdjacentHTML()

Inserta contenido HTML en una posición adyacente específica.

```
// Antes:  
// <div id="referencia">Elemento referencia</div>  
  
const referencia = document.getElementById('referencia');  
referencia.insertAdjacentHTML('afterend', '<p>HTML adyacente</p>');  
  
// Después:  
// <div id="referencia">Elemento referencia</div>  
// <p>HTML adyacente</p>
```

`.insertAdjacentText()`

Inserta texto en una posición adyacente específica.

```
// Antes:  
// <div id="referencia">Elemento referencia</div>  
  
const referencia = document.getElementById('referencia');  
referencia.insertAdjacentText('beforebegin', 'Texto adyacente');  
  
// Después:  
// Texto adyacente  
// <div id="referencia">Elemento referencia</div>
```

Navegación de elementos en el DOM

Navegar por elementos

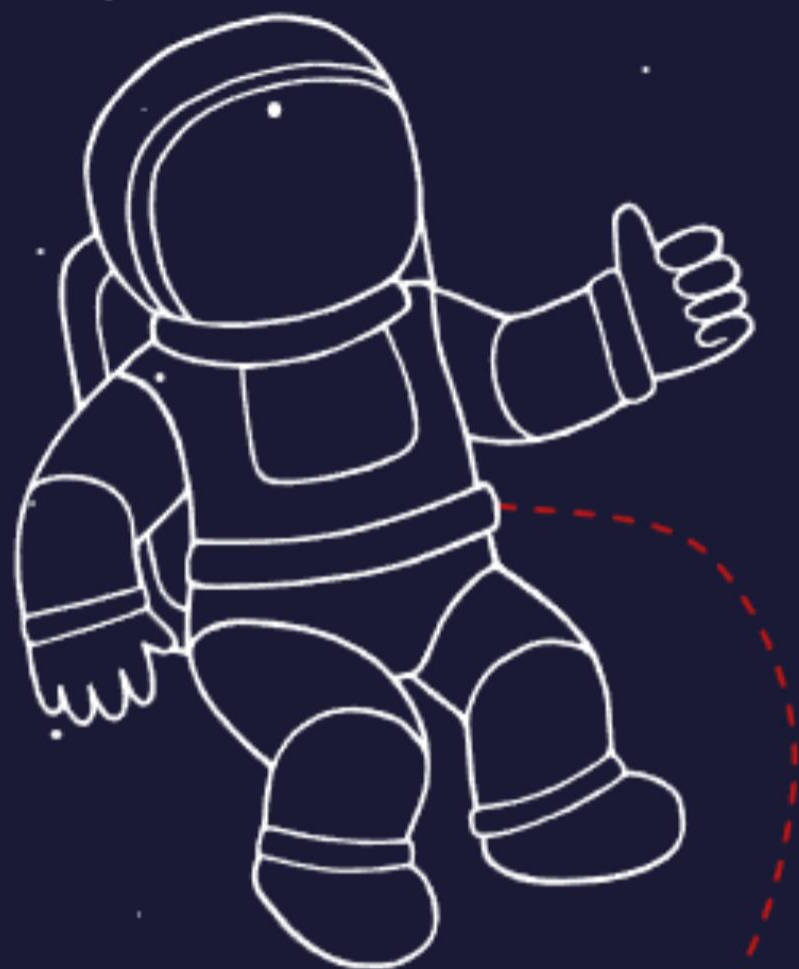
Usa propiedades como `parentElement`, `children`, `firstElementChild`, `lastElementChild` para navegar por elementos específicos.

```
const contenedor = document.getElementById('contenedor');  
console.log(contenedor.children); // Lista de hijos
```

Navegar por nodos

Incluye nodos no elementales como comentarios y texto con propiedades como `parentNode`, `childNodes`, etc.

```
const contenedor = document.getElementById('contenedor');  
console.log(contenedor.childNodes); // Incluye texto y comentarios
```

Programa acad mico CAMPUS

Trainer
Ing. Carlos H. Rueda C.

