



Curso de **Manipulación del DOM**

@jonalvarezz



Bienvenida al curso

¿Qué es el DOM?

Critical Rendering Path



Critical Rendering Path

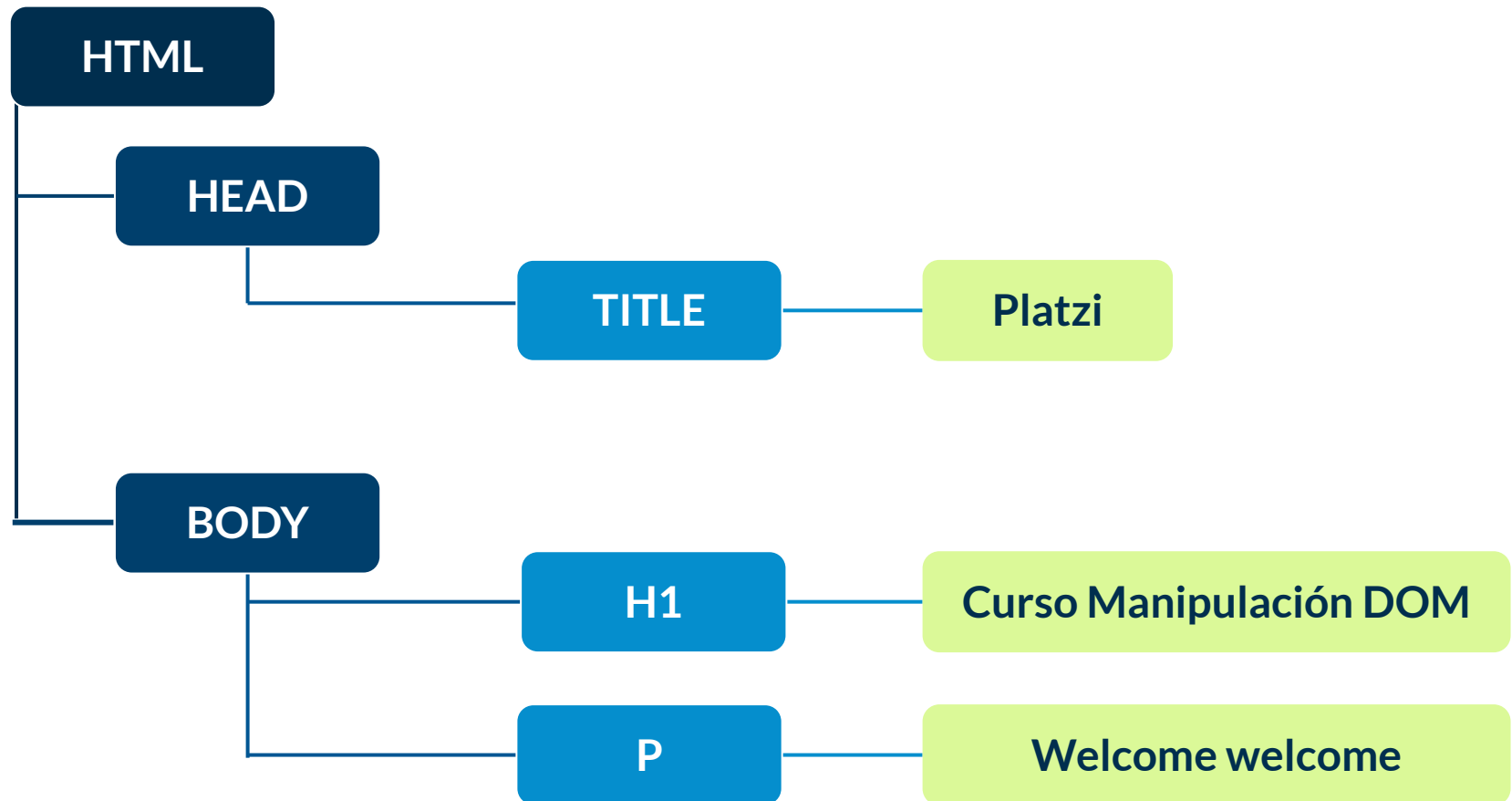
DOM

Document Object Model

CSSOM

CSS Object Model

DOM





DOM

- Es una representación del HTML.
- Estructura en forma de árbol de nodos.
- Es un modelo que puede ser modificado.



Web APIs modernas

DOM + JavaScript =
Web API



“

**Más de 70 Web APIs,
DOM es solo una de ellas.**

”

developer.mozilla.org/en-US/docs/Web/API

Biblias del desarrollo web

¿Cómo lo uso?



developer.mozilla.org

¿Puedo usarlo?



caniuse.com





Leer nodos



Leer nodos

- *document.getElementById*
- *parent.getElementsByTagName*
- *parent.getElementsByClassName*



Leer nodos

→ *parentElement*.querySelector

→ *parentElement*.querySelectorAll

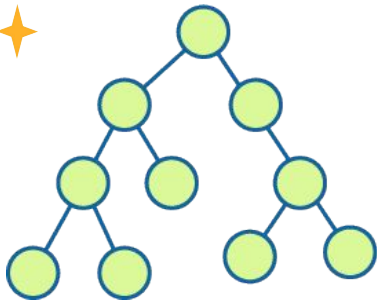


NodeLists vs. Array



Crear y agregar

Crear nodos



→ `document.createElement`

→ `document.createTextNode`

document.createElement

```
document.createElement(  
    "ELEMENTO"  
)
```



document.createElement

```
document.createElement(  
    "DIV"  
)
```



document.createElement

```
document.createElement(  
    "H1"  
)
```



document.createTextNode

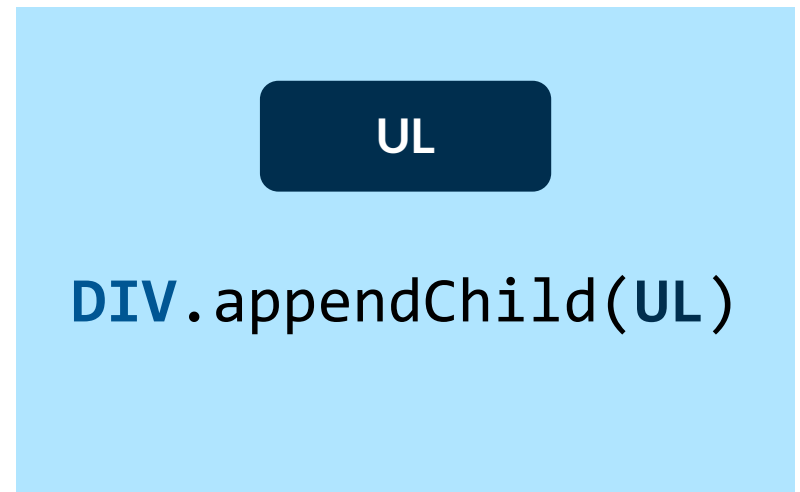
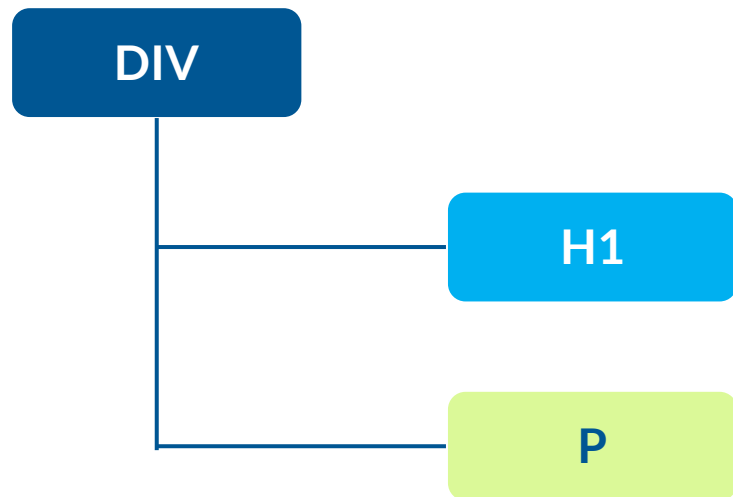
```
document.createTextNode(  
    "Texto"  
)
```



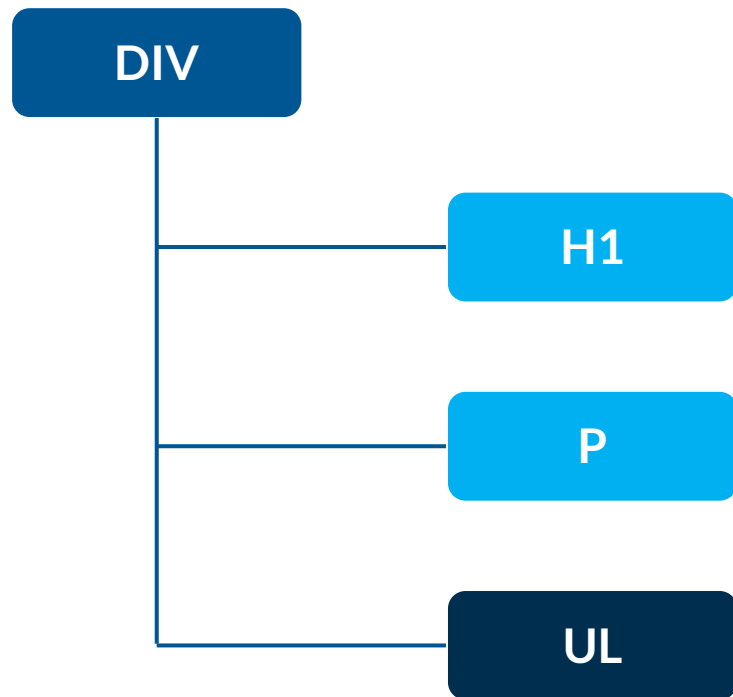
Agregar nodos

- `parentElement.appendChild`
- `parentElement.append`
- `parentElement.insertBefore`
- `parentElement.insertAdjacentElement`

node.appendChild



node.appendChild



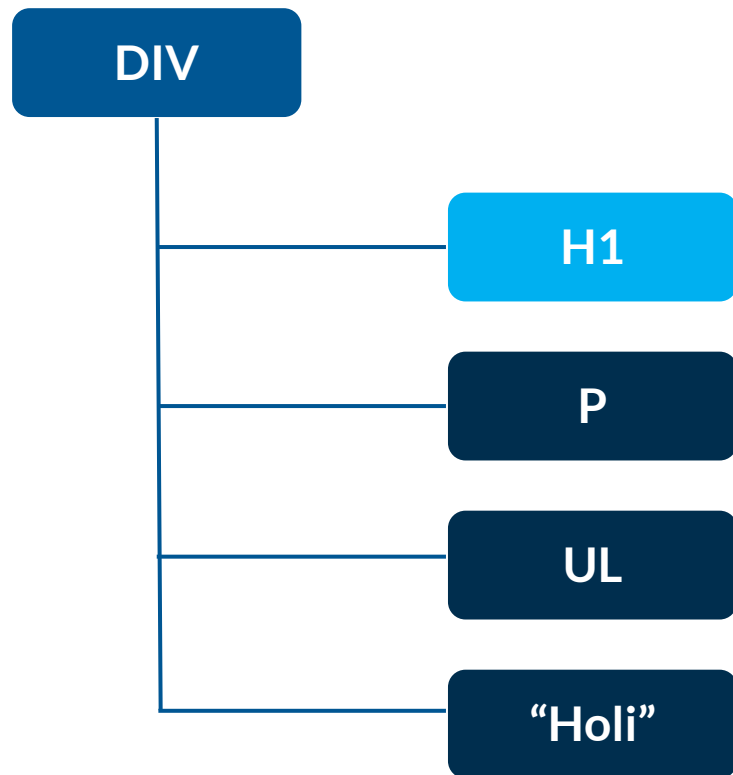
`DIV.appendChild(UL)`

node.append

Es la evolución de appendChild.

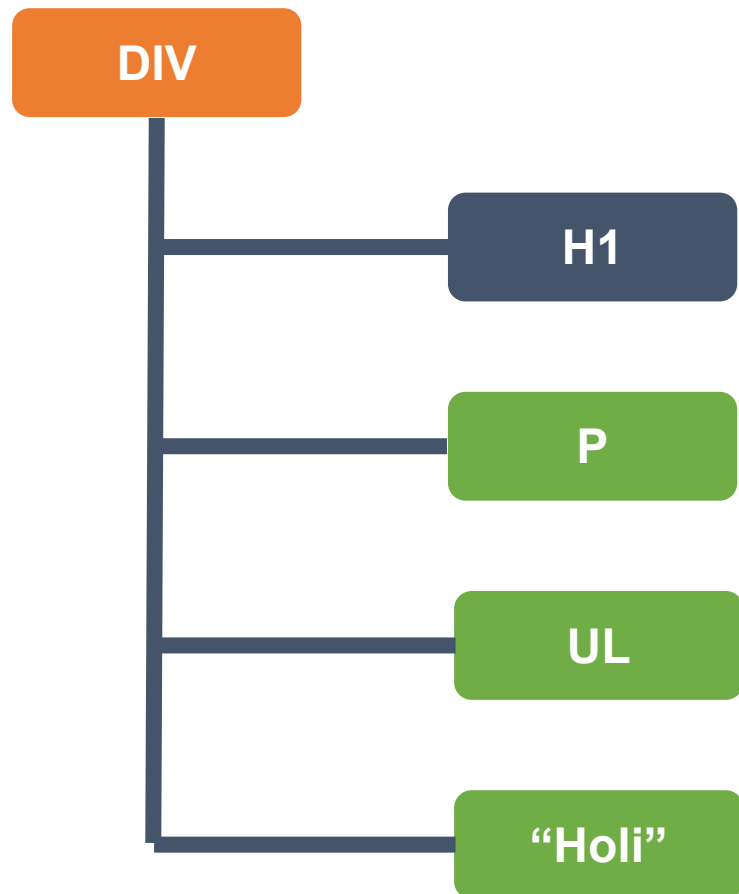
- Puedes agregar más de un nodo.
- Puedes agregar texto.
- IE 11: No soportado.

node.append



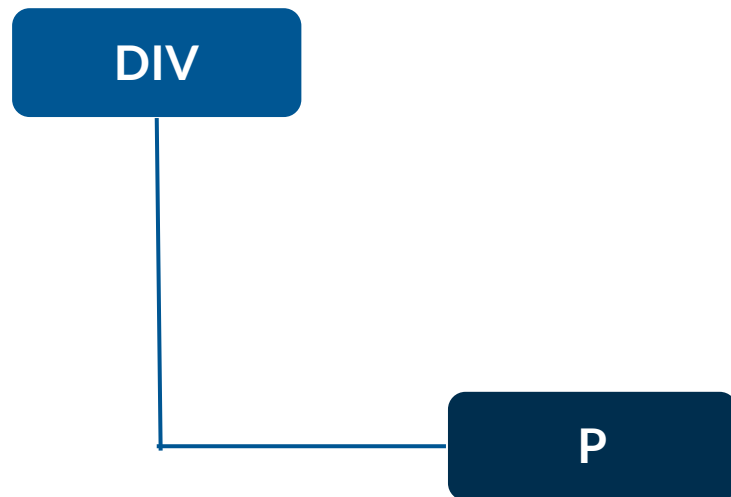
```
DIV.append(  
    P,  
    UL,  
    "Holi",  
)
```

node.append



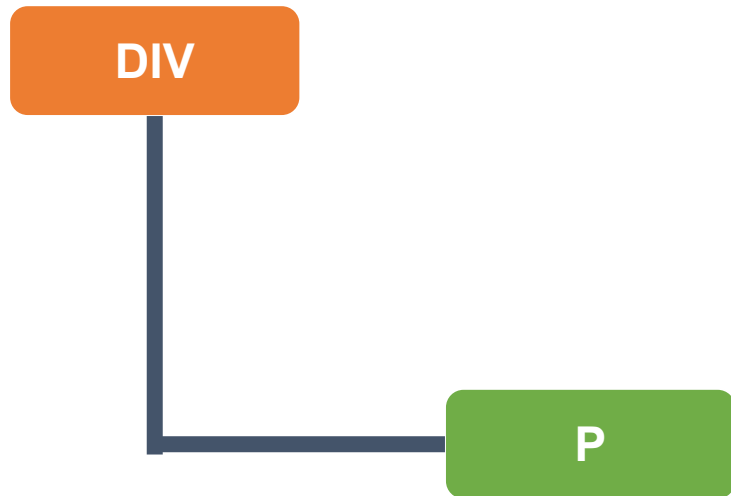
```
DIV.append(  
    P ,  
    UL ,  
    "Holi" ,  
)
```

node.insertBefore



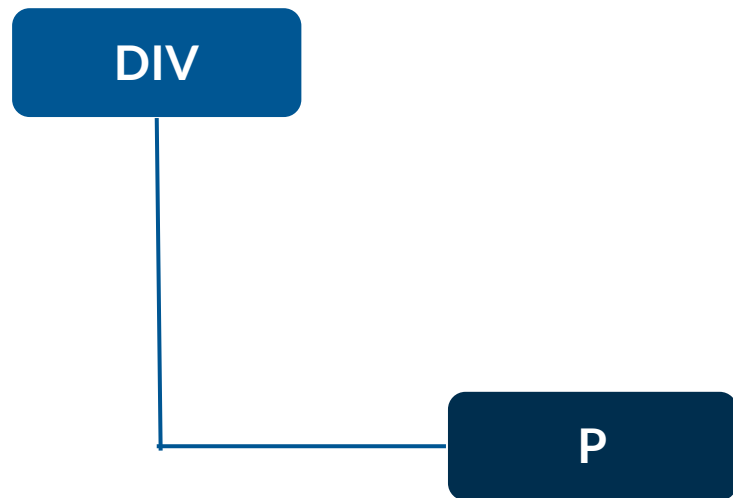
```
DIV.insertBefore(  
  NUEVO_NODO ,  
  REFERENCIA ,  
)
```

node.insertBefore



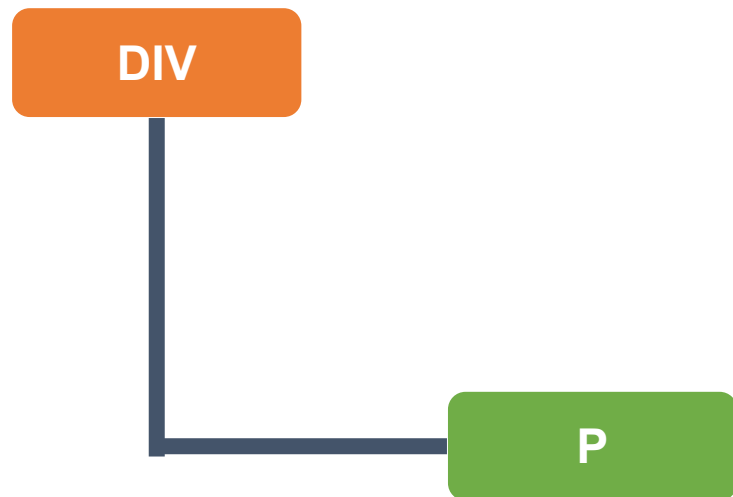
```
DIV.insertBefore(  
  NUEVO_NODO,  
  REFERENCIA,  
)
```

node.insertBefore



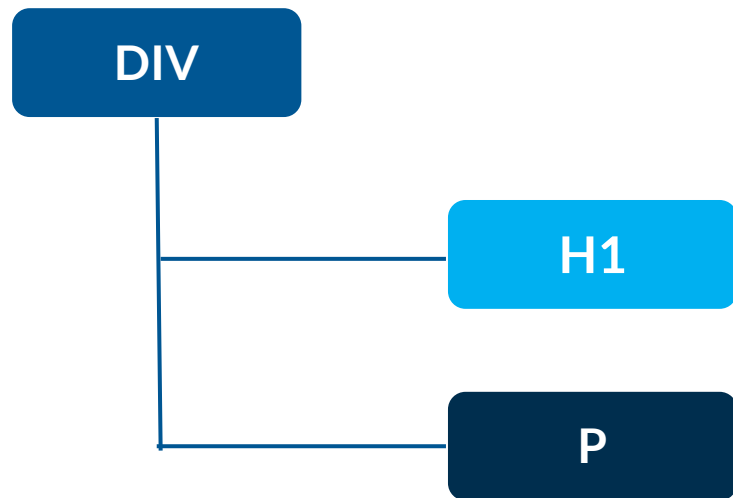
```
DIV.insertBefore(  
  H1 ,  
  P ,  
)
```

node.insertBefore



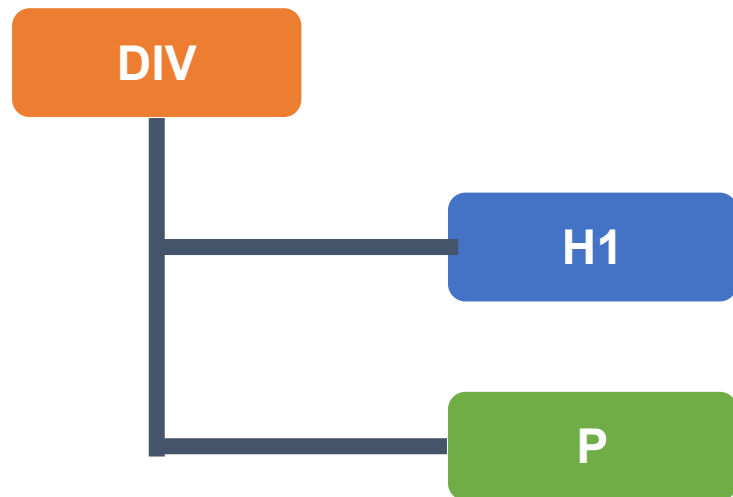
```
DIV.insertBefore(  
  H1 ,  
  P ,  
)
```

node.insertBefore



```
DIV.insertBefore(  
    H1 ,  
    P ,  
)
```


node.insertBefore



```
DIV.insertBefore(  
  H1 ,  
  P ,  
)
```

node.insertBefore

⚠ El nodo de referencia tiene que ser hijo directo del nodo base.

ref.insertAdjacentElement

`referencia.insertBefore(`

POSICIÓN:

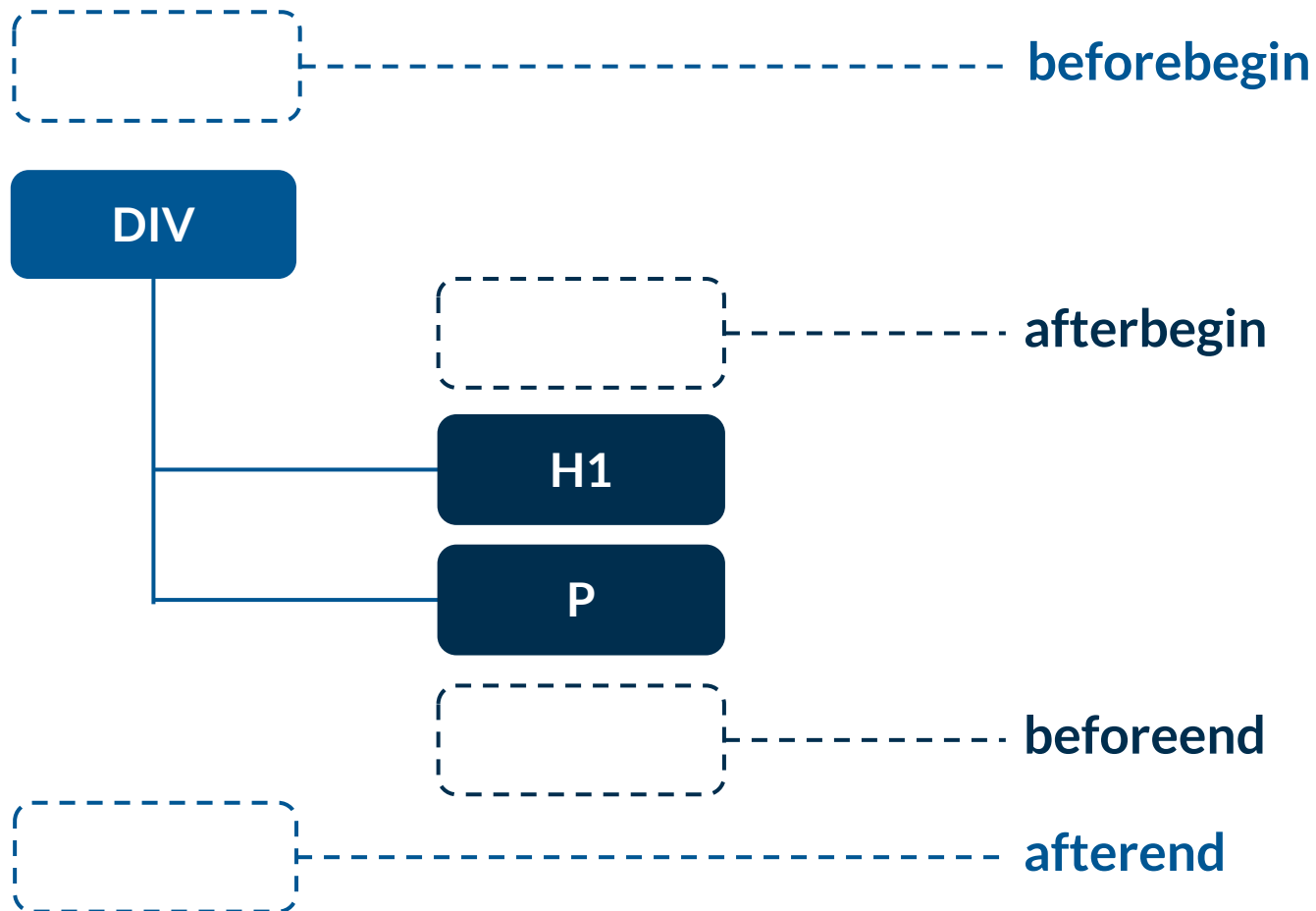
`'beforebegin' / 'afterbegin' / 'beforeend' / 'afterend',`

`NUEVO_NODO,`

`)`



ref.insertAdjacentElement





Otras formas de agregar





Otras formas

- `node.outerHTML` (leer)
- `node.innerHTML` (escribir)

Atributos y propiedades

¿Cuál es la diferencia?





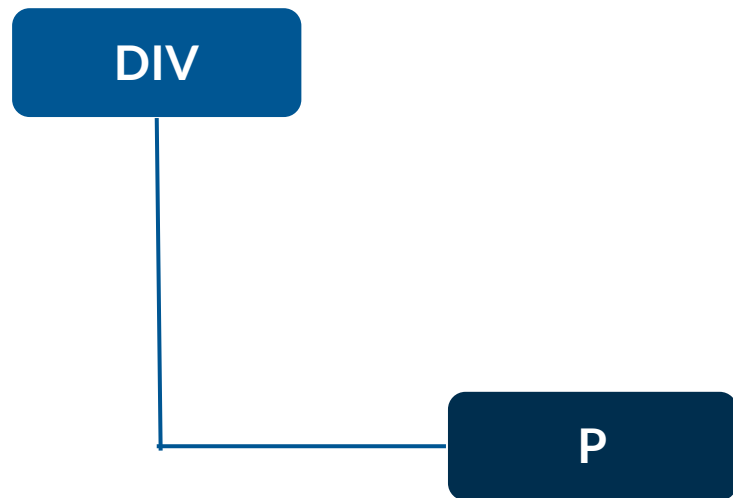
Eliminar nodos



Eliminar

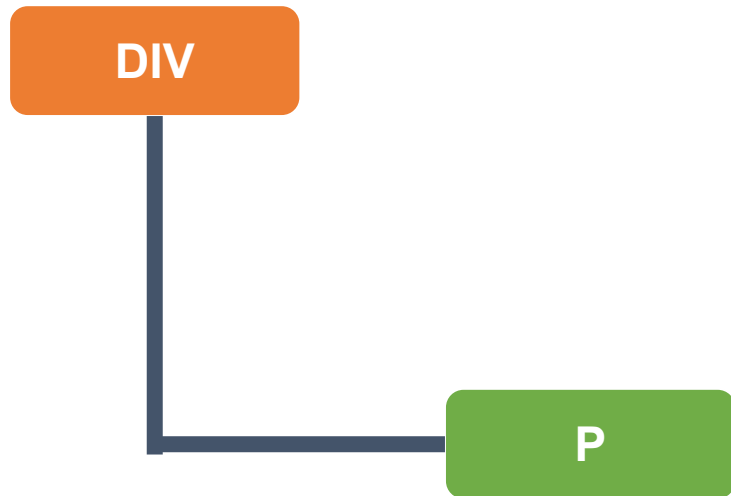
- `parentElement.removeChild`
- `document.remove`
- `document.replaceChild`

parent.removeChild



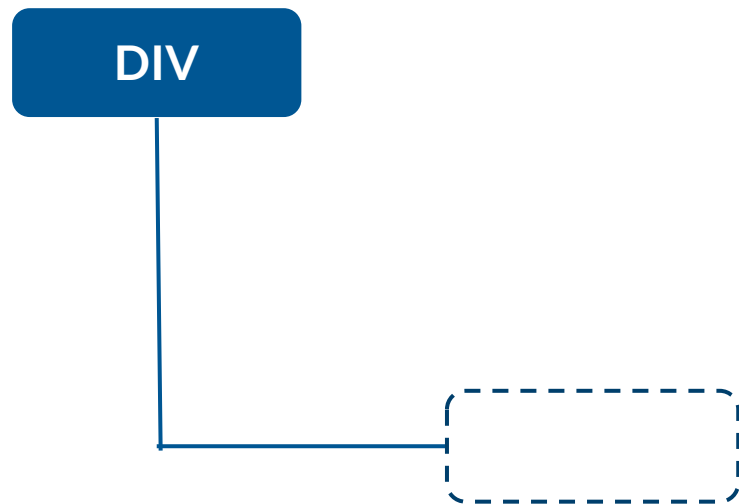
```
DIV.removeChild(  
    P ,  
)
```

parent.removeChild



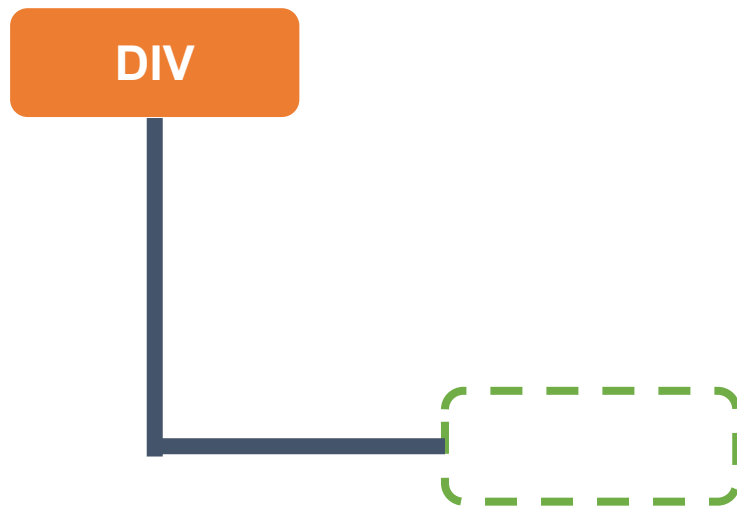
```
DIV.removeChild(  
    P,  
)
```

parent.removeChild



```
DIV.removeChild(  
    P  
)
```

parent.removeChild



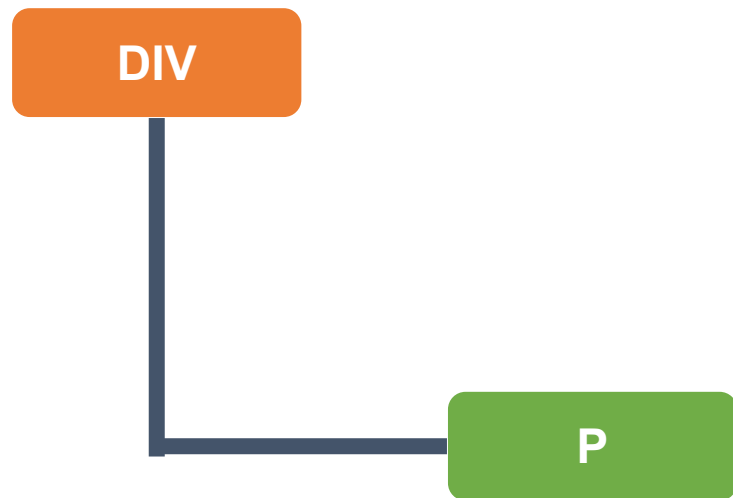
```
DIV.removeChild(  
    P  
,  
)
```



node.remove

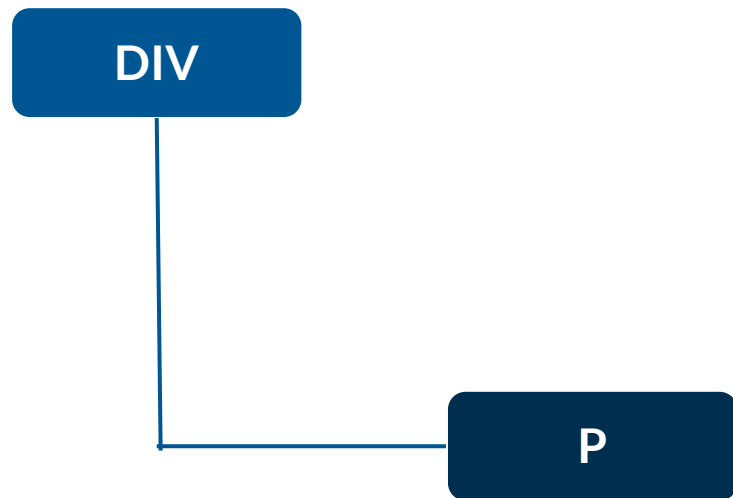
- Es la evolución de removeChild.
- No soportado IE.

parent.removeChild



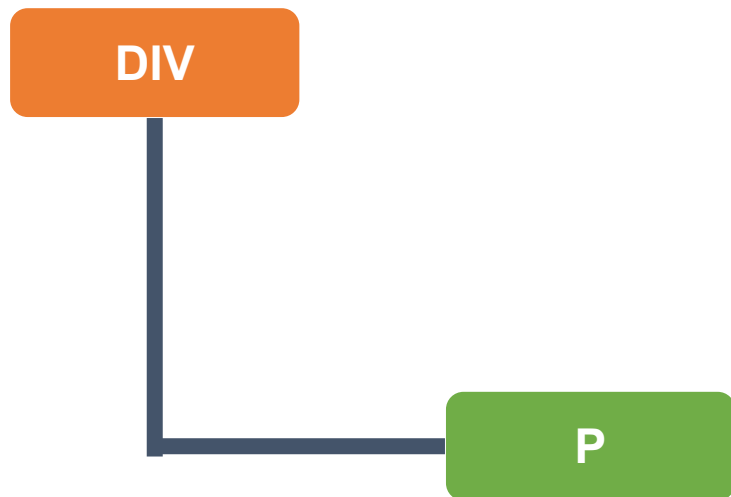
```
DIV.removeChild(  
    P,  
)
```


node.remove



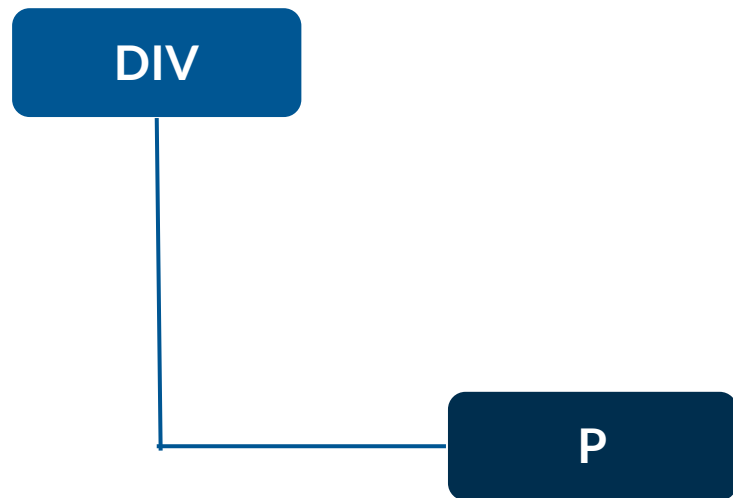
`P.remove()`

node.remove



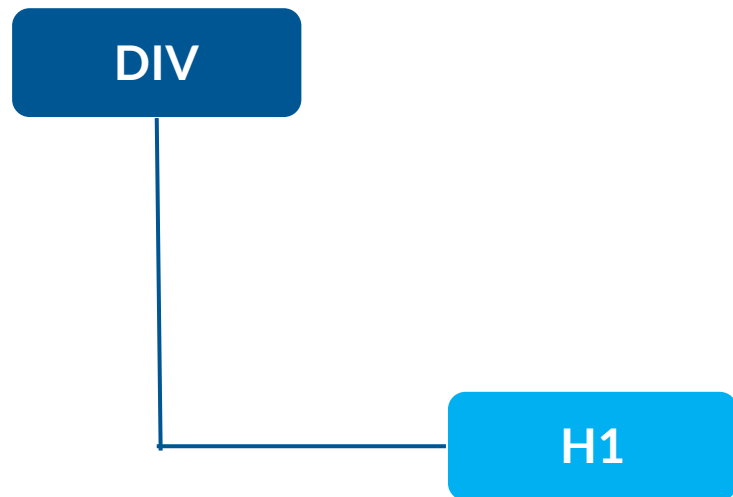
```
P.remove()
```

node.replaceChild



```
DIV.replaceChild(  
  H1 ,  
  P  
)
```

node.replaceChild



```
DIV.replaceChild(  
  H1 ,  
  P  
)
```

Operaciones en lote



Workshop #1

Fetch

PLATZI MANIPULACIÓN DOM

Avocados, la tiendita

Welcome to Avocados, La Tiendita, pásele and buy delicios
avocados



Maluma Hass Avocado

\$1.15



Fuerte Avocado

\$1.21



Gwen Hass Avocado

\$1.25



Bacon Avocado

\$1.51

Reaccionar a lo que sucede en el DOM

Eventos



Eventos

→ `node.addEventListener`

→ `node.removeEventListener`



node.addListener

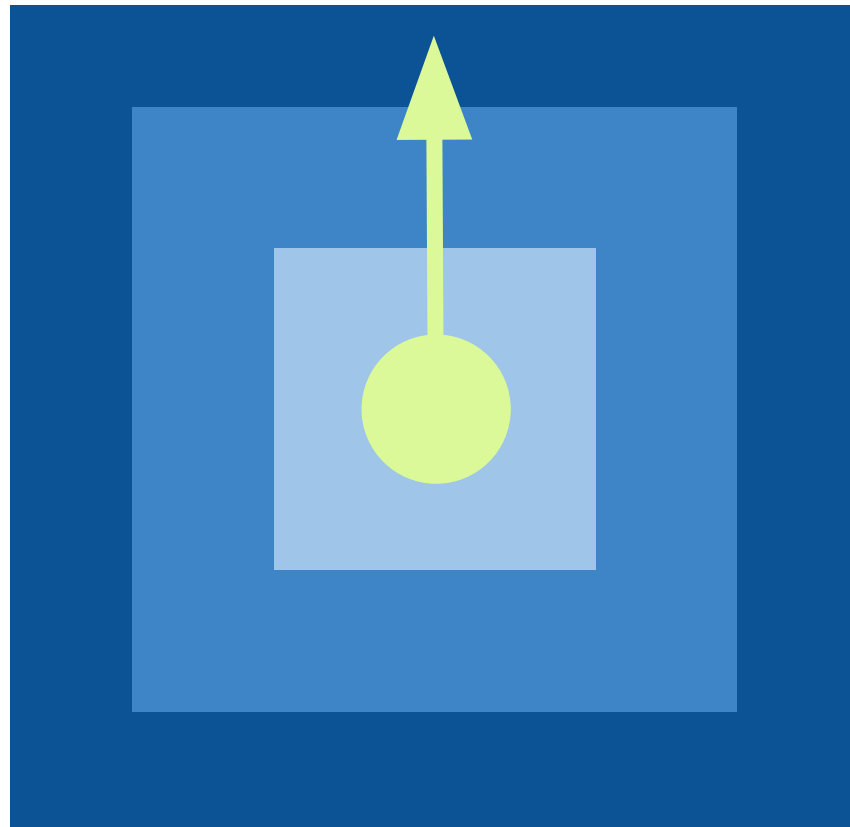
```
NODE.addListener(  
    tipo_de_evento,  
    acción (función),  
)
```

node.removeListener

```
NODE.removeEventListener(  
    tipo_de_evento,  
    acción (función),  
)
```

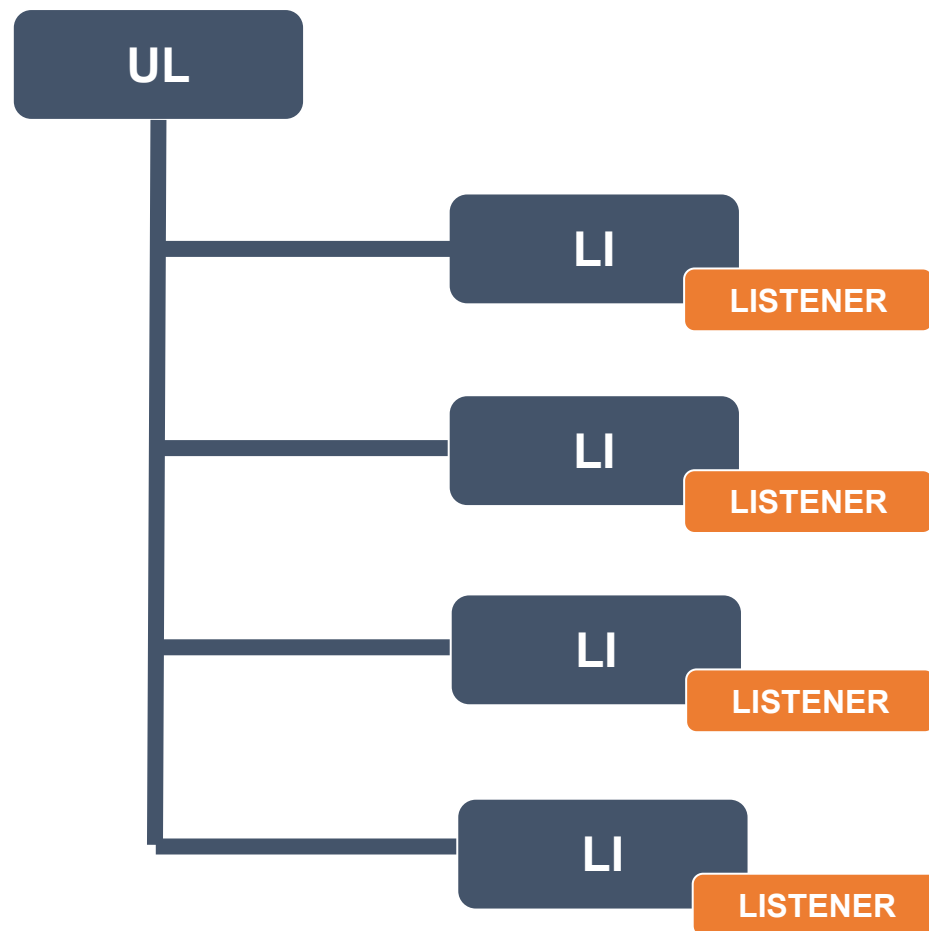
Propagación de eventos

Propagación

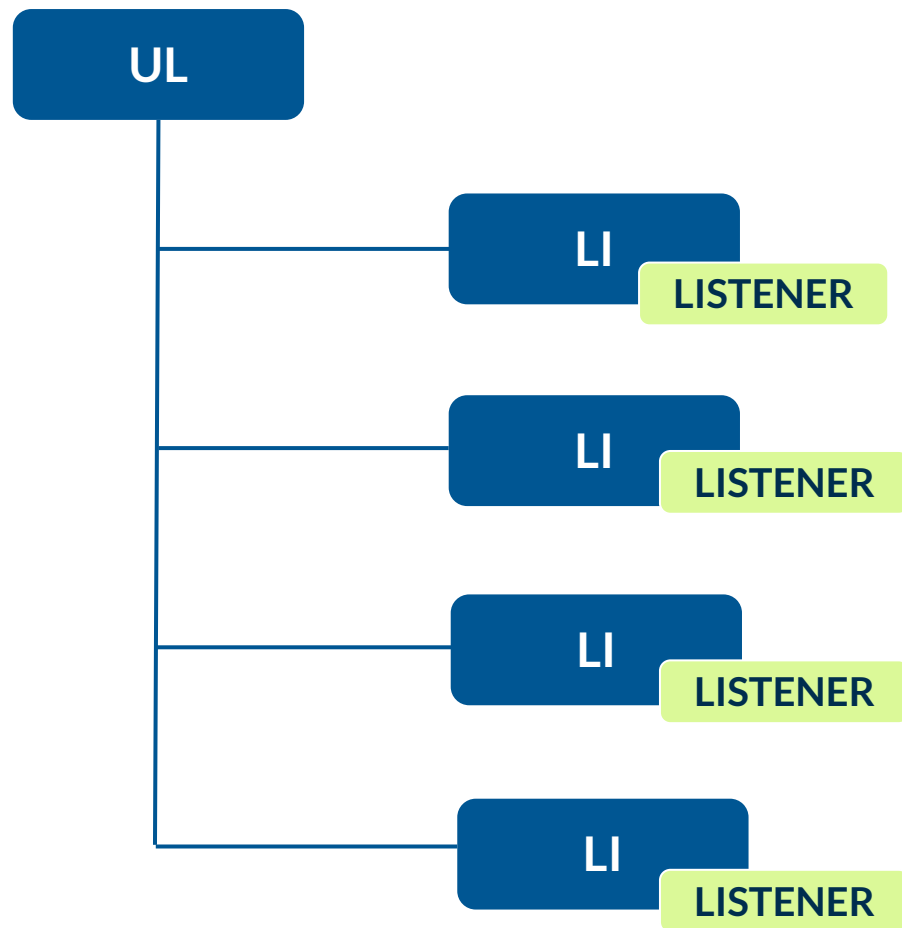


Delegación de eventos

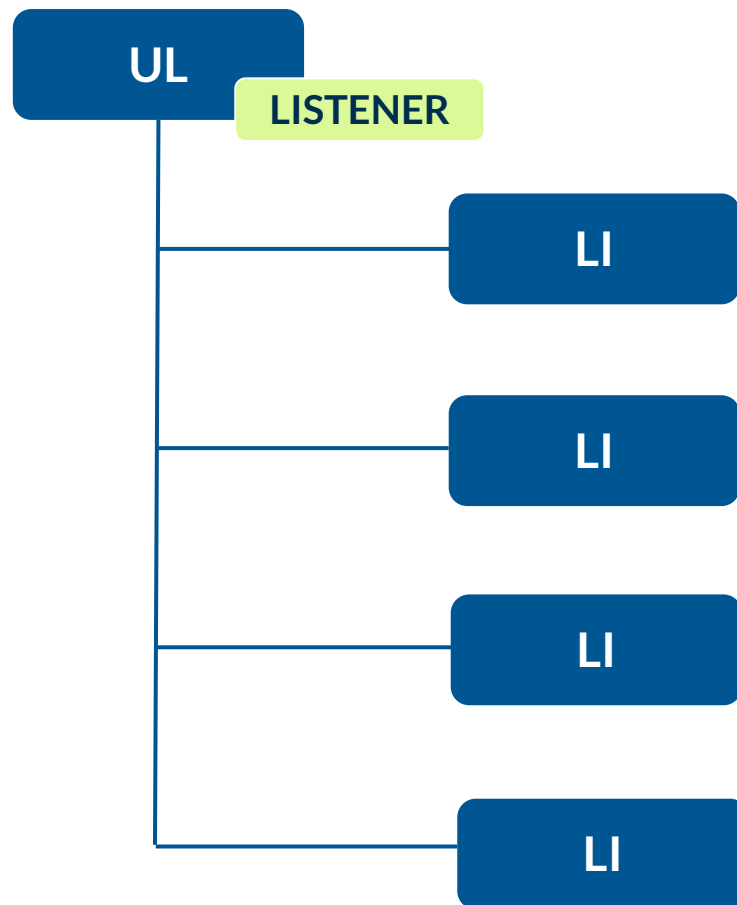
Sin delegación



Sin delegación



Con delegación





Workshop 2

Lazy loading

PLATZI MANIPULACIÓN DOM

Lazy loading

Plugin en vanilla JavaScript para cargar imágenes con lazy load.



Las imágenes agregadas no se descargarán hasta que sean visibles en la pantalla



Agrega una nueva imagen al final de la lista

Agregar Imagen

Limpiar





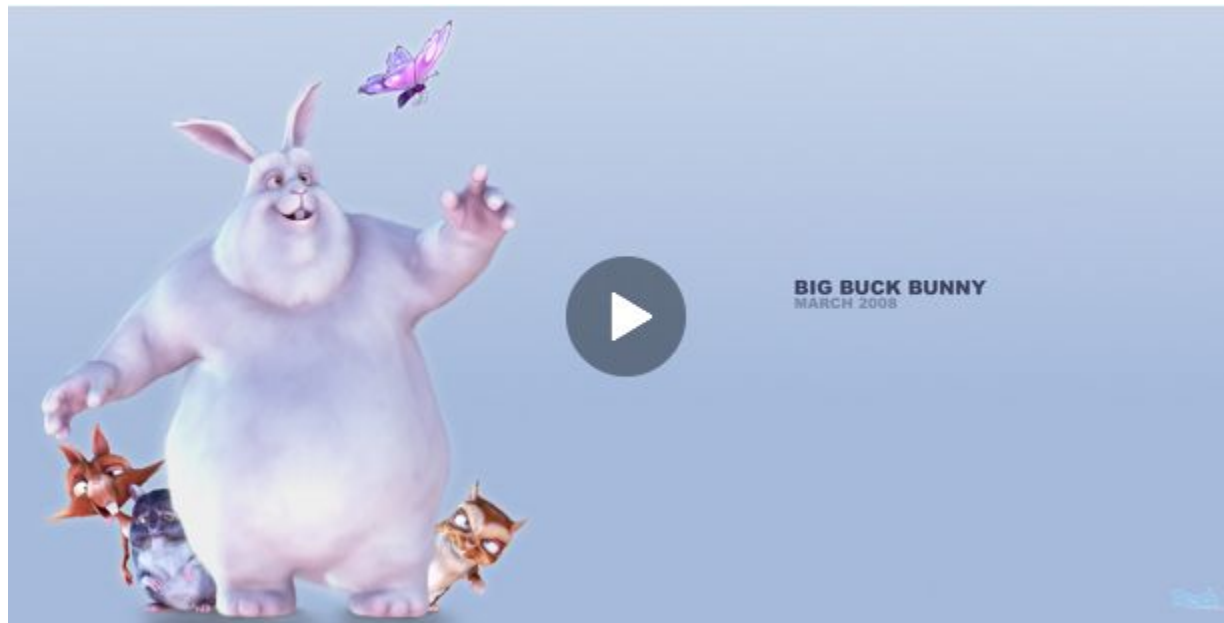
Workshop 3



Workshop 3

- Reproductor de video
- Aplicación del Clima

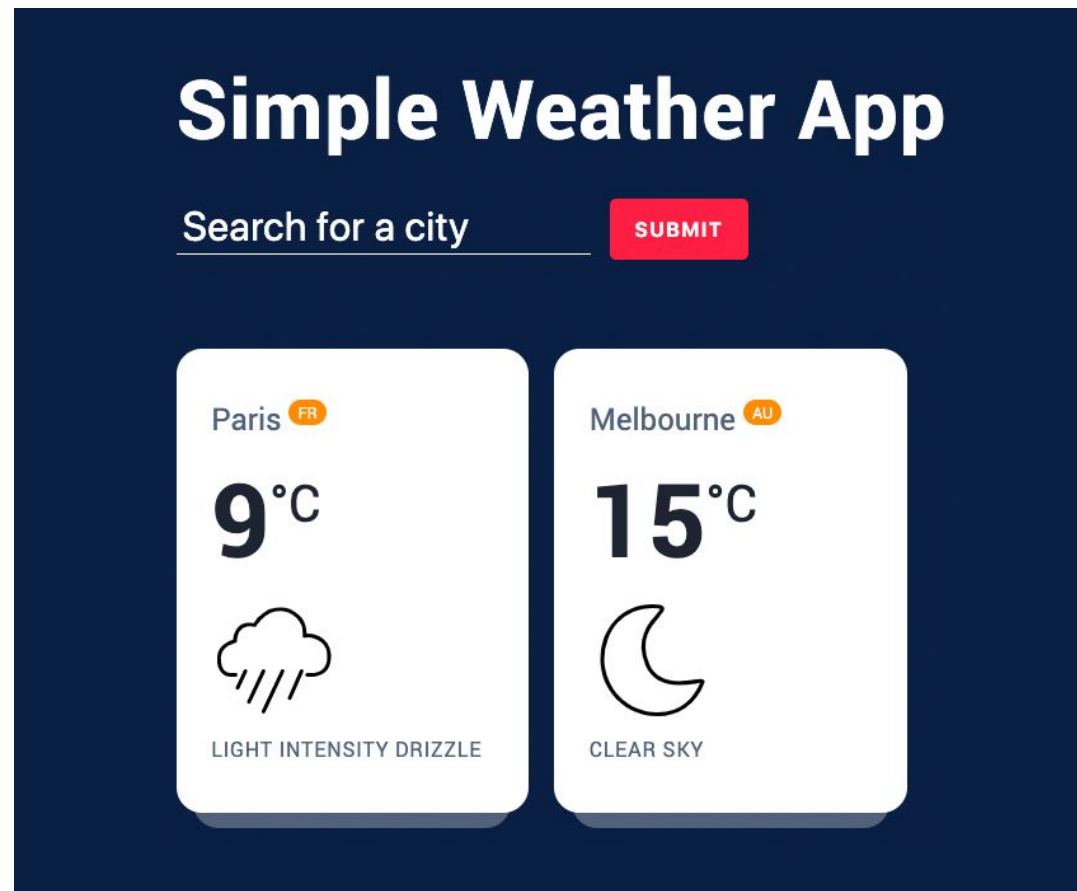
Reproductor de video



Reproductor de video

- Elementos: `<video>` y `<source>`
- Propiedades: `play()`, `pause()`, `mute()`,...
- Eventos: `play`, `complete`,...

Aplicación del Clima





Aplicación del Clima

- API: openweathermap.org
- Eventos: submit, input, click, ...

¿Y jQuery?




¿Y JSX?



Conclusiones

**Descargando
información y
creando nodos**



Enriqueciendo información

Usando la API de internacionalización

Comparte el resultado

**Nuestro propio plug in
de lazy loading**

Creando imágenes con JS

Interseccion observer



Aplicando lazy loading

Comparte el resultado