



Argentina  
programa  
4.0

**.UBA**fiuba   
FACULTAD DE INGENIERÍA

# **JavaScript: Clase 4 – DOM**

Bianchi – Zulaica

Febrero 2023

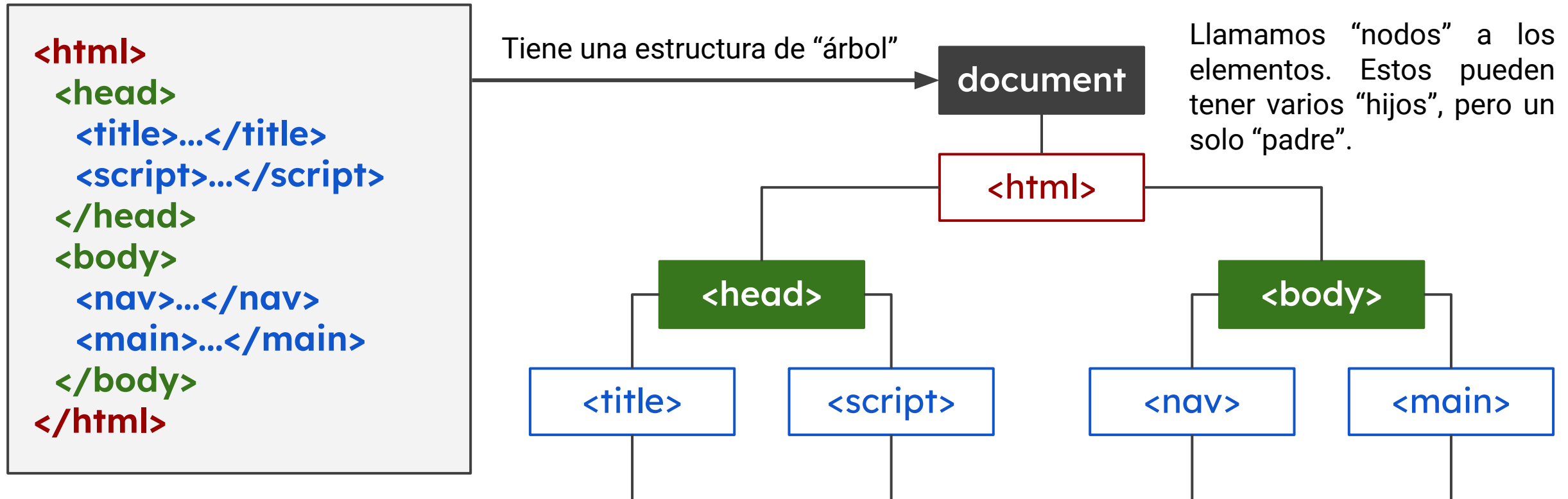
# Contenido

---

- Qué es el DOM
- Acceso a elementos
- Modificación de contenido
- Modificación de estilo
- Creacion y eliminacion de elementos

# Qué es el DOM

Domain Object Model: Es el modelo de **objetos** del documento, presenta las **etiquetas** html como objetos a los que podemos **acceder** o **modificar**



# Acceso a elementos

```
let elemento = document.querySelector( selector )
```

- **document** es el nodo “raíz” del DOM y podemos accederlo directamente.
- **querySelector** es un método de los elementos que: recibe un selector de css y devuelve el primer elemento que lo cumpla.

También existen métodos como **getElementById**, al cual le pasamos directamente un id; y **querySelectorAll**, que devuelve una lista de todos los elementos válidos.

# Ejemplo con JS Interno

```
<html>
  <head>
    <title>Mi Pagina</title>
  </head>
  <body>
    <h1>Hola Mundo!</h1>
    <p>Esta es mi primera pagina web</p>
    <script>
      let parrafo = document.querySelector("p");
      parrafo.innerText = "Texto cambiado desde JS";
    </script>
  </body>
</html>
```

El **script** debe estar **después** del **párrafo**. Ya que el párrafo debe “existir” previo a obtenerlo.

El elemento "p"

Obtenemos el elemento

Accedemos al texto interno

# Ejemplo con JS Externo

```
<html>
  <head>
    <title>Mi Pagina</title>
    <script src="script.js" defer></script>
  </head>
  <body>
    <h1>Hola Mundo!</h1>
    <p>Esta es mi primera pagina web</p>
  </body>
</html>
```

index.html

Usamos **defer** para posponer la ejecución. Ya que el párrafo debe “existir” previo a obtenerlo.

Espera hasta procesar el HTML para ejecutar el JS

```
let parrafo = document.querySelector("p");
parrafo.innerText = "Texto cambiado desde JS";
```

script.js

Igual que antes

# Modificación de contenido

`elemento.innerText`

→ Como vimos en el ejemplo, podemos acceder y reasignar el texto

`elemento.innerHTML`

→ También podemos acceder y reasignar el contenido como HTML

Proba cambiar el código del ejemplo anterior con este

```
let parrafo = document.querySelector("p");
parrafo.innerText = "Texto cambiado desde JS";
parrafo.innerHTML += " <strong>Texto en negrita</strong>";
```

**script.js**



# Modificación de estilo

`elemento.style`

→ Podemos editar el atributo style, da estilo en línea (no recomendable)

`elemento.classList`

→ Podemos editar la **lista** de **clases** del elemento

Proba agregar este código al ejemplo.  
Definí un estilo para la clase “titulo”.

```
let titulo = document.querySelector("h1");  
titulo.style = "color: red; font-size: 20px";  
titulo.classList.add("titulo");
```

**script.js**

# Creación y Eliminación de elementos

- Podemos **crear** elementos editando el **innerHTML** de un elemento existente y agregando una etiqueta, como ya vimos.
- También podríamos utilizar los métodos **createChild** y **appendChild**, que son más *eficientes*.
- Para **eliminar** elementos, utilizamos el método **removeChild**. El método debe llamarse sobre el nodo padre con el hijo como parámetro.

```
let parrafo = document.querySelector("p");
document.body.removeChild(parrafo);
let subTitulo = document.createElement("h2");
subTitulo.innerText = "Este es el subtítulo";
document.body.appendChild(subTitulo);
```

script.js

# Ejemplo

## agregarTarea.html

```
<html>
  <head>
    <title>Tareas</title>
    <script src="agregarTarea.js"></script>
  </head>
  <body>
    <h2>Tareas</h2>
    <input type="text" />
    <button type="button"
      onclick="agregarTarea()"
    >Agregar</button>
    <hr />
    <ul></ul>
  </body>
</html>
```

## agregarTarea.js

```
function agregarTarea() {
  let input = document.querySelector("input");
  // Accedemos al atributo value (valor)
  let tarea = input.value;
  let ul = document.querySelector("ul");
  ul.innerHTML += "<li>" + tarea + "</li>";
}
```

# Resumen

---

- ✓ El DOM presenta las **etiquetas** html como objetos a los que podemos **acceder** o **modificar**
- ✓ Utilizamos **document.querySelector** para acceder a los elementos utilizando **selectores** de **CSS**
- ✓ Podemos modificar el **contenido** mediante los atributos **innerText** o **innerHTML**
- ✓ Podemos modificar el **estilo** mediante los atributos **style** o **classList**
- ✓ Además de editar el innerHTML, podemos utilizar los métodos **createChild**, **appendChild** y **removeChild** se pueden utilizar para crear o eliminar elementos

# Ejercicio

---

La siguiente aplicacion: <https://codepen.io/nicolasezequielzulaicarivera/pen/rNZZbgx> debería ser un TATETI, pero le faltan pedazos de código, necesitamos que los completes.

Las líneas con código faltante estarán marcadas con comentarios describiendo lo que debería hacer. Encontrarás las siguientes tareas:

1. Devolver el elemento con id "clase"+numero
2. Agregar el número "i" como texto de la celda
3. Verificar si la lista contiene la clase que marca al oponente
4. Agregar la marca pasada por parámetro como clase
5. Remover la clase que marca al oponente

**[www.ingenieria.uba.ar](http://www.ingenieria.uba.ar)**

**f**    /ingenieriauba

 /FIUBAoficial