FrontEndII
Profesor: Martín Bonino

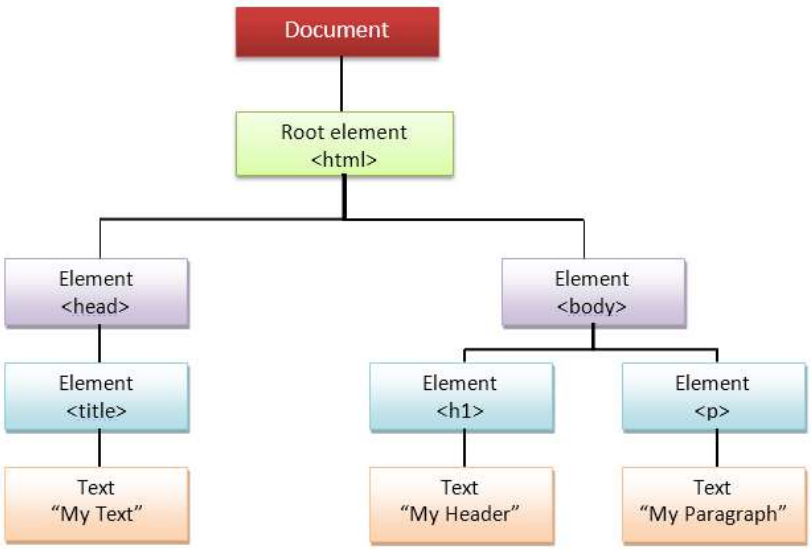
Módulo 2. Manipulación del DOM

Sesión 4. Introducción al DOM
Marzo 8 de 2022

Objeto Window y Document

El objeto window representa la ventana que contiene al documento y el objeto document representa al DOM (documento HTML) cargado en esa ventana. El objeto document se encuentra dentro del objeto window (ej: window.document.title).

DOM:
El DOM (document object model) representa al documento que se carga en el navegador como un árbol de nodos, en donde cada nodo representa una parte del documento.



Objeto Windows

En la consola:
window // Retorna la dirección que se encuentra en la barra de direcciones

window.innerHeight // Retorna la altura de la ventana en pixeles
window.innerWidtt // Retorna el ancho de la ventana en pixeles

window.location // Objeto que incluye el host, el hostname, el pah, etc.

window.location.href = 'http://digitalhouse.com' // Carga la pagina de DH

Objeto document:

document // Devuelve HTMLDocument http://localhost

document.bgColor = "red" // El color de fondo cambia a rojo

document.styleSheets // Muestra todas las hojas de estilo vinculadas

DOM Selectores

querySelector()

let titulo = document.querySelector(`.title'); // Retorna el primer elemento del HTML que contenga la clase "title"

querySelectorAll()

let nombres = document.querySelectorAll(); // Retorna un nodlist (listado de elementos)

let div=document.querySelectorAll('div');

getElementById()

let marca=document.getElementById('marca'); // Retorna el elemento con el id especificado

Otra opción:
let marca=document.querySelector('#marca');

Sincrónico:

innerHTML: Permite insertar HTML en un elemento
innerText: Permite insertar un texto a un elemento

```
let p1 = getElementById("p1");
p1.innerHTML = `<ul>
                <li>Soy un item</li>
                <li>Soy otro item</li>
                </ul>`;
```

(template literal)

Sesión 5. Modificando elementos con JavaScript
Marzo 10 de 2022

.innerHTML
Permite leer o modificar el contenido de una etiqueta HTML.

element.innerHTML += 'Contenido agregado';

.innerText
Permite leer o modificar el texto de una etiqueta HTML

element.innerText += 'Texto agregado';

Propiedad Style
Permite leer y sobreescribir las reglas CSS que se aplican sobre un elemento seleccionado

```
let titulo = document.querySelector('.title');
titulo.style.color = 'cyan';
titulo.style.textAlign = 'center'; // Usar camelCase en JavaScript
titulo.style.fontSize = '12px';
titulo.style.backgroundColor = '#dddddd';
```

Template literals

` Mi variable vale ${miVariable} `// Lo envuelto entre llaves se interpreta como código JavaScript

Ejemplo:

```
const nombre= 'Mauro';
const miTemplate = `Mi nombre es ${nombre}`;
console.log(miTemplate); // Imprime “Mi nombre es Mauro”
```

Ejemplo:

index.HTML

```
<html lang="es">
<head>
	<meta charset="UTF-8">
	<script src="scripts.js"></script>
	<title>Ejemplo</title>
</head>
<body id="body">
```

```
</body>
</html>
```

scripts.js

```
function escribirHTML(titulo, texto) {
	const body = document.getElementById('body');
	const miTemplate = `
		<h1>${titulo}</h1>
		<p>${texto}</p>
		`;
	body.innerHTML += miTemplate;
}

escribirHTML('Hola', 'Esto es un ejemplo de template string en html.');
escribirHTML('Es dinámico', 'Podemos insertar elementos HTML mediante
<b>JavaScript</b>');
```

Modificar clases

classList.add()
Agrega la clase al elemento.

classList.remove()
Elimina la clase del elemento.

classList.toogle()
Agrega la clase, si es que no la tiene. En caso de tenerla, la remueve.

classList.contains()
Pregunta si el elemento tiene la clase o no. Devuelve un valor booleano.

Sincrónico

```
let btntarea = document.getElementById(“btntarea”)

btntarea.addEventListener(“click”, () => {
	let tarea = document.getElementById(“tarea1”);
	tarea.classList.toogle(“completada”);
}
```

Sesión 6.
Marzo 11 de 2022

Sesión 7. Trabajando con nodos.
Marzo 15 de 2022

Nodos en HTML

Son elementos o etiquetas del HTML que en conjunto forman un "árbol de nodos" al que llamamos DOM (Document Object Model). Cada nodo del árbol es un objeto.

- document (Nodo principal)
- etiquetas HTML (nodos de elementos)
- Nodos de texto
- Nodos de comentarios

Creación de nodos en el DOM:

createElement()
Crea un nodo de tipo elemento según el nombre de etiqueta HTML especificada

```
document.createElement("inpu");
```

createTextNode()
Crea un nodo de texto explicitado entre comillas. Para visualizarlo, se requiere asignarlo a un elemento HTML existente.

```
document.createTextNode("Hola Mundo");
```

appendChild()
Adhiere dentro del DOM un elemento hijo a un elemento padre.

```
document.body.appendChild(titulo);
```

Ejemplo:
```
var botonVerMas = document.createElement("button");
var botonTexto = document.createTextNode("Ver más");
botonVerMas.appendchild(botonTexto);
document.body.appendChild(botonVerMas);
```

Atributos dinámicos

hasAttribute()
Sirve para consultar si el elemento posee o no un determinado atributo.

```
element.hasAttribute("src");
```

getAttribute()
Permite obtener el valor de un determinado atributo.

element.getAttibute(“src”);

removeAttribute()
Borra el atributo y sus valores del elemento. Si no lo encuentra, no hace nada.

element.removeAttibute(“src”);

SetAttribute()
Permite agregar un atributo con su respectivo valor al elemento seleccionado.

elemento.setAttribute("src", "imagen_portada.jpg");