

Que es DOM?

DOM

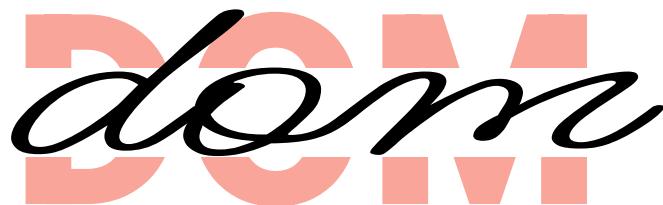
La traducción del acrónimo inglés DOM (Document Object Model) significa Modelo de Documento de Objeto; se trata de modelar todo el HTML.

Que podemos hacer gracias a DOM?

Cuando se carga una página web, el navegador crea un modelo de objetos de documento de la página.

Con el modelo de objetos, JavaScript está totalmente habilitado para crear HTML dinámico:

- JavaScript puede añadir, cambiar y eliminar todos los elementos y atributos HTML de la página.
- JavaScript puede cambiar todos los estilos CSS en la página.
- JavaScript puede reaccionar a todos los eventos existentes en la página.
- JavaScript puede crear nuevos eventos en la página.



Como accedo al DOM?

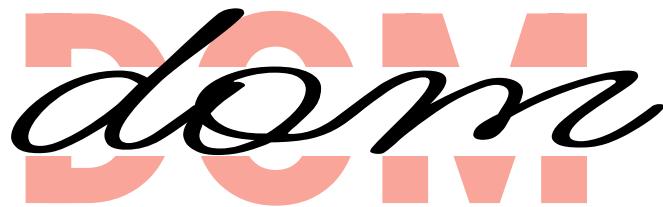
En Javascript, la forma de acceder al DOM es a través de un objeto llamado document, que representa el árbol DOM de la página o pestaña del navegador donde nos encontramos. En su interior pueden existir varios tipos de elementos, pero principalmente serán o :

- ELEMENT no es más que la representación genérica de una etiqueta: HTMLElement.
- NODE es una unidad más básica, la cuál puede ser ELEMENT o un nodo de texto.

Todos los elementos HTML, dependiendo del elemento que sean, tendrán un tipo de dato específico.

Tipo de dato	Tipo específico	Etiqueta	Descripción
ELEMENT	HTMLElement	HTMLDivElement	<div> Capa divisoria invisible (en bloque).
ELEMENT	HTMLElement	HTMLSpanElement	 Capa divisoria invisible (en línea).
ELEMENT	HTMLElement	HTMLImageElement	 Imagen.
ELEMENT	HTMLElement	HTMLAudioElement	<audio> Contenedor de audio.

Obviamente, existen muchos tipos de datos específicos, uno por cada etiqueta HTML.

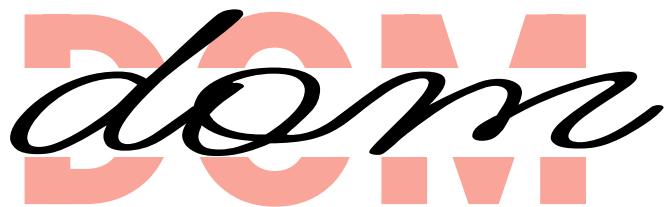


Donde se inserta DOM?

Y la pregunta que surge es: ¿pero el DOM es parte de HTML o de JavaScript? De hecho, de ninguno — él es **generado por el browser**. Al cargar la página, el navegador crea el documento, la interfaz, y Javascript usa el DOM para conectarse al HTML.

Para comunicarse entre ellos, es necesario insertar la etiqueta `script` en el archivo HTML, y como buena práctica, debe ser **antes del cierre de la etiqueta body** para que los scripts se carguen después del código base.

```
<script src="script.js"></script>
```



Todo muy lindo pero... Como navego el DOM?

Hay varias formas de navegar dentro del DOM, en JavaScript usamos el objeto `document` y a través del punto accedemos a las propiedades y métodos, siendo posible seleccionar, cambiar, borrar y crear elementos a los componentes del sitio web. Para realizar estas acciones disponemos de algunos métodos, tales como:

- `document.getElementById();`
- `document.getElementsByClassName();`
- `document.getElementsByTagName();`
- `document.querySelector();`
- `document.querySelectorAll();`
- `document.createElement();`
- `element.addEventListener();`

Y como toda la teoría no tiene sentido si no la ponemos en práctica, aquí te dejo algunos ejemplos muy simples para comprender como funciona esta hermosa parte de JavaScript. Porque, por si aún no lo notaste, es mi parte favorita



Primero que nada, es super importante tener en definido en qué parte de mi HTML voy a interactuar a través de javascript. En este caso creamos un contenedor vacío con un ID único

```
<main>
  <section>
    <div id="miTexto"></div>
  </section>
```

Luego de eso, en mi hoja de JavaScript voy a pedirle que lo traiga: “hey Javascript, necesito obtener este contenedor. Lo llamaremos “texto” para poder darle funcionalidad dinámica y así poder modificarlo

```
let texto = document.getElementById("miTexto")
```



Ahora bien, qué es lo que puedo hacer? A continuación te voy a mostrar un ejemplo muy simple, no como los de internet que no hacen mas que confundirnos.
Supongamos que tengo un array así:

```
const productos = [  
  {id: 1, nombre: "zapatillas deportivas", precio: 500, stock: 5, img: "../assets/deportivas.jpeg", descripcion: "zapatillas unisex deportivas"},  
  {id: 2, nombre: "zapatillas urbanas", precio: 700, stock: 10, img: "../assets/urbanas.jpeg", descripcion: "zapatillas unisex urbanas"},  
  {id: 3, nombre: "zapatillas nauticas", precio: 400, stock: 3, img: "../assets/nauticas.jpeg", descripcion: "zapatillas nauticas unisex"},  
  {id: 4, nombre: "sandalias", precio: 350, stock: 20, img: "../assets/sandalias.jpeg", descripcion: "sandalias unisex"},  
  {id: 5, nombre: "ojotas", precio: 200, stock: 15, img: "../assets/ojotas.jpeg", descripcion: "ojotas de verano unisex"},  
]
```

Notarás que los objetos de mi array tienen varias propiedades en común: Nombre, precio, id, etc.

Todas esas propiedades las voy a utilizar para poder crear un ecommerce que sea dinámico y me permita simular procesos de compra.

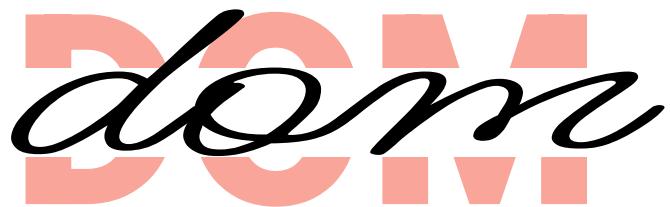


Voy a comenzar recorriendo mi array de productos con un forEach. De esta manera:

```
productos.forEach( producto =>{  
  let div = document.createElement('div')  
  div.className = "card"  
})
```

Que está haciendo mi código?

Basicamente le estoy pidiendo a JavaScript que recorra mi array, le estoy diciendo que cada objeto de mi array “productos” se llamará “producto” y por ultimo estoy creando un div, el cual va a contener toda la informacion de mis productos. Además a ese div ya estoy asignandole una classname que me permitirá darle estilos



A continuación, haré una prueba rápida para corroborar que mi ciclo `forEach` se esté ejecutando correctamente. De qué manera?

```
productos.forEach( producto =>{  
    let div = document.createElement('div');  
    div.className = "card"  
    div.innerHTML= `  
        <p>${producto.nombre}</p>  
    `  
  
    texto.appendChild(div)  
})
```

Si todo está correcto, al levantar mi web en el navegador voy a encontrarme con algo así:

A screenshot of a Mac OS X desktop environment showing a browser window. The window title is "Document". The address bar shows the URL "127.0.0.1:5500/index.html". The content area of the browser displays five lines of text, each representing an item from the "productos" array: "zapatillas deportivas", "zapatillas urbanas", "zapatillas nauticas", "sandalias", and "ojotas".

zapatillas deportivas
zapatillas urbanas
zapatillas nauticas
sandalias
ojotas



Entonces con este código, podemos entender qué:

```
productos.forEach( producto =>{  
    let div = document.createElement('div');  
    div.className = "card"  
    div.innerHTML= `  
        <p>${producto.nombre}</p>  
    `  
    texto.appendChild(div)  
})
```

a mi nuevo contenedor “div” le estoy pidiendo que dentro contenga un fragmento HTML. Ese fragmento de HTML deberá mostrar el nombre de mi objeto producto. El cual se llama producto* ya que es el nombre que le asignamos como explicamos al principio.

Una vez definido todo esto le estoy diciendo: “Hey, recuerdas la variable “texto” que declaré al principio? Okey, mi nueva variable div con contenido HTML será un HIJO de ese contenedor! ”

*si no lo recuerdas ve a ver la declaración del forEach cuando empezamos a declarar la función



Por supuesto que todo esto no termina aquí! A mi innerHTML puedo colocarle todas las propiedades que contenga mi objeto, y darle los estilos que mas se amolden a mi gusto :D Cómo lo hago?

```
productos.forEach( producto =>{  
    let div = document.createElement('div');  
    div.className = "card"  
    div.innerHTML= `  
        <p>${producto.nombre}</p>  
          
        <p>Precio: $$ ${producto.precio}</p>  
        <span>${producto.descripcion}</span>  
        <span>Stock disponible: ${producto.stock}  
        </span>  
        `;  
    texto.appendChild(div)  
})
```



Y para evitar que se vea así de feito, le vamos a dar unos buenos estilos con css, bootstrap o de la forma que prefieras! :D

Precio: \$400

zapatillas nauticas unisex Stock disponible: 3

sandalias





Nos puede quedar asi como algo potable, pero si dan rienda suelta a la creatividad, seguro quedará mucho mejor!!

2 (2) WhatsApp apuntes-programacion - Figma Document

127.0.0.1:5500/index.html

zapatillas deportivas



Precio: \$500

zapatillas unisex deportivas
Stock disponible: 5

zapatillas urbanas



Precio: \$700

zapatillas unisex urbanas
Stock disponible: 10

zapatillas nauticas



Precio: \$400

zapatillas nauticas unisex
Stock disponible: 3

Buscar

15:21 10/5/2023



Y por supuesto, aquí te dejo los estilos que implementé:

```
#miTexto {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    column-gap: 25px;  
    row-gap: 25px;  
    justify-items: center;  
}
```

```
.card {  
    display: flex;  
    flex-direction: column;  
    gap: 15px;  
    padding: 20px;  
    align-items: center;  
}
```

```
.card img {  
    width: 200px;  
}  
.card p {  
    font-size: 20px;  
    font-family: "Segoe UI", Tahoma, Geneva,  
    Verdana, sans-serif;  
}
```

```
.card span {  
    font-size: 15px;  
    font-family: "Courier New", Courier,  
    monospace;  
}
```

```
.card {  
    border: solid 1px red;  
    border-radius: 10px;  
    width: fit-content;  
}
```