

# Objetos – DOM

## Introducción:

Un objeto es un grupo de variables agrupadas, estas variables se denominan propiedades. Los objetos también disponen de una serie de métodos o funciones que realizan operaciones con las variables propias de estos.

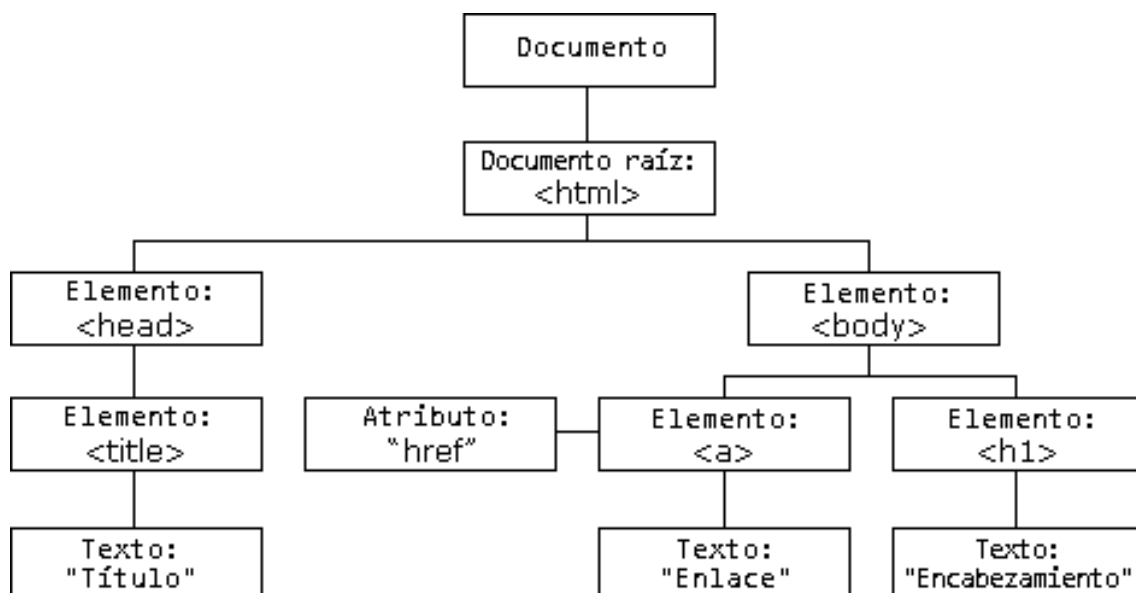
## Objetos incorporados:

JavaScript utiliza todo un conjunto de objetos que representan el navegador y sus elementos, el documento HTML, las imágenes, los enlaces, las tablas etc.

Cada uno de estos objetos esta definido por sus propiedades, los datos que almacena en su interior y por sus métodos, es decir, las acciones que podemos hacer sobre ellos.

Para comprender a fondo la programación en JavaScript, es imprescindible conocer la jerarquía de objetos que lo forman. Cada uno de estos objetos puede contener varios tipos de elementos, otros objetos JavaScript, propiedades y métodos.

La jerarquía de los objetos JavaScript es la siguiente:



El modelo **DOM** (*Document Object Model*) es una interfaz de programación (*API*) para documentos HTML y XML válidos, que permite acceder de manera estándar a todos sus elementos como si fueran objetos, con una estructura lógica.

El modelo DOM es un estándar creado por la W3C.

Con el modelo DOM HTML se puede modificar, añadir o eliminar cualquier elemento del documento HTML.

Este modelo es accesible desde JavaScript.

Ejemplo de cambio de texto en un elemento:

```
var texto = document.getElementById("titulo");  
texto.innerHTML = "Nuevo título";
```

## Acceso:

En el modelo DOM HTML el objeto **document** es la página web, es el *padre* de todos los elementos del documento, y el acceso a cualquier elemento del documento se realiza a través de él.

Métodos para encontrar elementos:

Método	Descripción
<code>document.getElementById()</code>	Busca un elemento por su atributo id
<code>document.getElementsByTagName()</code>	Busca elementos por su etiqueta HTML
<code>document.getElementsByClassName()</code>	Busca elementos por su nombre de clase
<code>document.forms[], document.images[], document.links[], document.anchors[]</code>	Busca elementos por los elementos de colección del objeto HTML

Ejemplos de búsqueda de elementos:

```
// Buscar un elemento por la id="contenedor"

var x = document.getElementById("contenedor");

// Buscar los párrafos que hay en el elemento con id="contenedor"

var parrafos = x.getElementsByTagName("p");

parrafos[0].innerHTML = "Le cambiamos el texto al primer párrafo";

// Buscar elementos span en el documento

var s = document.getElementsByTagName("span");

// Buscar los elementos por el nombre de la clase "encabezamientos"

var e = document.getElementsByClassName("encabezamientos");
```

Desde el modelo DOM HTML se puede acceder a colecciones de objetos que dependen de otros:

- Anclas: **document.anchors.**
- Formularios: **document.forms.**
- Imágenes: **document.images.**
- Enlaces: **document.links**

Ejemplos:

```
// Imprime el valor de todos los elementos input del primer formulario

for (var i = 0; i < document.forms[0].length; i++) {

    document.writeln(document.forms[0].elements[i].value + "<br>");

}

// Cambiar el atributo src de la primera imagen del documento

document.images[0].src = "nueva/imagen.png";
```

## Modificación contenido:

Para modificar el contenido, atributos o estilos de text CSS se hace uso de las siguientes propiedades y metodos:

Método/propiedad	Descripción
<code>elemento.innerHTML="contenido"</code>	Cambia el contenido de un elemento HTML
<code>elemento.attribute="valor"</code>	Cambia el valor del <i>atributo</i> de un elemento HTML
<code>elemento.setAttribute(atributo,valor)</code>	Añade un <i>atributo</i> a un elemento HTML y le asigna un <i>valor</i>
<code>elemento.style.propiedad="estilo"</code>	Cambiar el estilo CSS <i>propiedad</i> de un elemento HTML

// Cambiar el texto del elemento identificado como 'id\_elemento'

```
var x = document.getElementById("id_elemento");
```

```
x.innerHTML = "Nuevo texto al elemento";
```

```
x.style.color = "rgb(0, 0, 255)"; // color de texto azul
```

```
x.className = "claseCSS"; // Cambiarle el valor del atributo class
```

// Asignar una clase CSS a un elemento de formulario

```
var i = document.getElementById("id_input");
```

```
i.size = "60"; // Le cambia el valor del tamaño del elemento input
```

```
i.setAttribute("class", "claseCSS"); // Le añade el atributo class
```

```
i.className = i.className + " otraClaseCSS"; Añadirle otra clase
```

## Añadir, eliminar o modificar elementos HTML:

Se pueden añadir, eliminar o modificar elementos HTML con los siguientes métodos:

Método	Descripción
<code>document.createElement(<i>eLemento</i>)</code>	Crea un nuevo elemento HTML <i>elemento</i> .
<code>document.removeChild(<i>nodo</i>)</code>	Borra un nodo hijo (un elemento HTML).
<code>document.appendChild(<i>nodo</i>)</code>	Añade un nodo (un elemento HTML)
<code>document.replaceChild(<i>nodo</i>)</code>	Reemplaza un nodo (un elemento HTML)
<code><u>document.createTextNode(<i>texto</i>)</u></code>	Crear nodo de texto para un elemento HTML

```
// Crear una nueva lista

var u = document.createElement("UL");

// Crear un nuevo elemento de lista

var l = document.createElement("LI");

// Crear un nodo de texto

var t = document.createTextNode("Elemento lista");

// Añadir el nodo de texto al elemento de lista

l.appendChild(t);

// Añadir el elemento de lista a la nueva lista

u.appendChild(l);

// Añadir la nueva lista al documento

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