

# Desarrollo Web En Entorno Cliente

## TEMA 5

### Interacción con el usuario y gestión de eventos



# Índice

- ▶ Acceso al HTML desde JavaScript.
- ▶ Manejo de eventos en JavaScript.
- ▶ Introducción a JQuery.
- ▶ Validación básica de formularios.

# HTML desde JavaScript

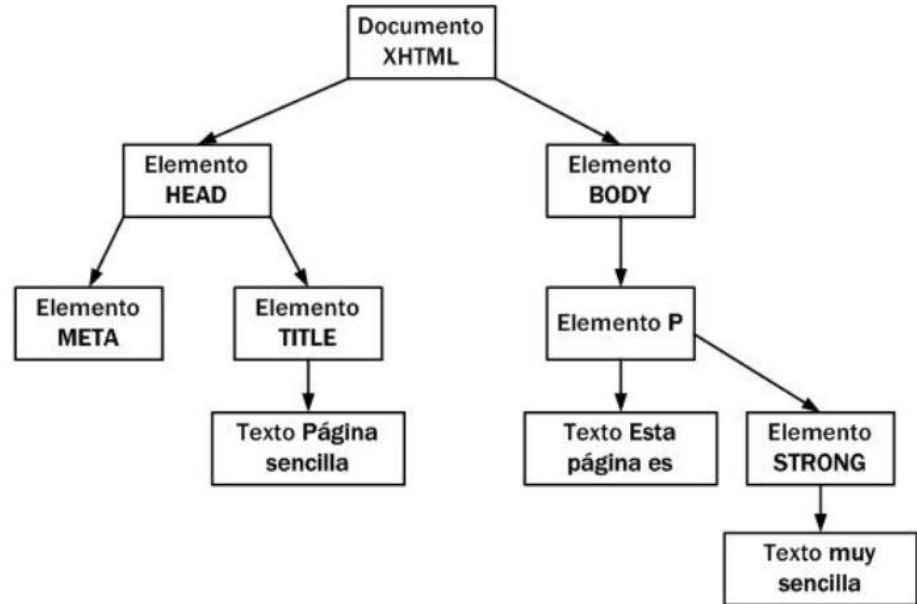


- ▶ El **DOM** es la herramienta que más ha influido en el desarrollo de páginas webs dinámicas y aplicaciones web complejas.
- ▶ Hasta ahora, tan solo podíamos añadir nuevos elementos al comienzo de la página.
- ▶ Eso ensuciaba y complicaba en exceso el código.
- ▶ El **DOM** permite acceder y manipular páginas HTML.
- ▶ De este modo podemos modificar o eliminar una parte concreta de la Web o añadir nuevos elementos.

- ▶ Una de las tareas que más se realizan con JS es la de manipular la apariencia de una página Web, es decir, recuperar y modificar el valor de los atributos CSS o HTML.
- ▶ Es decir, el DOM es un estándar para obtener, modificar, añadir, borrar elementos de un documento HTML.
- ▶ El DOM transforma un documento HTML en una serie de nodos interconectados que representan el contenido de la página y sus relaciones. Estos nodos se organizan como un árbol formando el llamado árbol de nodos.

## Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset=UTF-8" />
  <title>Página sencilla</title>
</head>
<body>
<p>Esta página es <strong>muy sencilla</strong></p>
</body>
</html>
```





- ▶ La especificación completa de DOM define 12 tipos de nodos, aunque nosotros vamos a trabajar solamente con unos pocos:
- ▶ **Document**: nodo raíz que derivan todos los demás nodos del árbol.
- ▶ **Element**: representa cada una de las etiquetas HTML. Este es el único tipo de nodo que contiene atributos y del que pueden derivar otros nodos.
- ▶ **Attr**: representa cada uno de los atributos de las etiquetas HTML, es decir, uno por cada atributo=valor.

- ▶ **Text:** nodo que contiene el texto de una etiqueta HTML.
- ▶ Las páginas HTML producen árboles con miles de nodos.
- ▶ El DOM proporciona una serie de **funciones** que permiten acceder a **cualquier nodo de la página de forma sencilla e inmediata**.
- ▶ En las ultimas revisiones de JavaScript se ha evolucionado de manera que solamente son necesarias dos.
- ▶ **querySelector y querySelectorAll.**



- ▶ Para poder acceder a los nodos, **la página debe haber sido cargada por completo**, para que el árbol de nodos se haya construido.
- ▶ Una de las opciones para asegurarse de que el navegador ha cargado, es poner el script después de la etiqueta body (no siempre funciona).
- ▶ Pero la opción más recomendable es utilizar **el evento onload**.

```
window.onload=Inicio;  
  
function Inicio()  
{  
    //Codigo para acceder a los elementos de la página  
}
```

- ▶ Las funciones para acceder directamente a un nodo o nodos son:
- ▶ **querySelector()** // Busca usando selectores CSS un solo elemento de la página. **Devuelve un objeto de tipo Element.**
- ▶ **querySelectorAll()** //Busca usando selectores CSS todos los elementos que coincidan con ese selector. **Devuelve un Array de tipo Element**, aunque solo coincida un solo elemento.
- ▶ Son funciones del objeto **document**.

- ▶ El código siguiente guarda en la variable enlaces todos los nodos de la página que sean enlaces.

```
var enlaces = document.querySelectorAll("a");  
var primer= enlaces[0]; // guarda el primer enlace de la página  
for(i=0; i<enlaces.length; i++)  
{  
    alert(enlaces[i].href+"<br>"); //muestra el href de cada enlace  
}
```

- ▶ La función **querySelector** también funcionaria, pero solo devolvería solo el primer enlace.



- ▶ **EJERCICIO 1:** Realizar un script para la página pagina.html que muestre el número de párrafos de la página y el número de enlaces de la página. Después mostrará a donde apunta el primer y el último enlace.
- ▶ El funcionamiento del script será el siguiente: Una vez se haya cargado la página por completo se mostrarán mediante alert los siguientes 3 mensajes:

**Resumen: X párrafos y X enlaces**

**El primer enlace apunta a: enlace**

**El último enlace apunta a: enlace**

- ▶ Podemos seleccionar **por id**.
- ▶ Esto lleva implícito que solo debe haber un elemento con un id.
- ▶ Para ello usamos **querySelector()** que es el indicado cuando solo queremos seleccionar un elemento.

```
var cabecera = document.querySelector("#cabecera");  
alert(cabecera);
```

- ▶ **EJERCICIO 2:** Realizar un script para pagina.html muestre las etiquetas con los siguientes id:
- ▶ **buscador**
- ▶ **noticias**
- ▶ **menu\_principal**
- ▶ **secundario**
- ▶ **pie**
- ▶ ¿Cuál de ellos esta aplicada una etiqueta distinta?



- ▶ Si queremos seleccionar elementos de una clase debemos usar **querySelectorAll()**.
- ▶ **EJERCICIO 3:** Realizar un script para la página pagina.html que muestre cuántos elementos tienen aplicada cada una de las siguientes clases y de qué tipo de etiqueta son:
  - ▶ **clear**
  - ▶ **fecha**
  - ▶ **artículo**

- ▶ Uno de los atributos del objeto **Element** es **innerHTML**.
- ▶ Este **atributo es muy potente** y hace referencia a todo el contenido de un elemento incluido el código que pueda contener.
- ▶ Podemos **consultar, modificar, editar o borrar el contenido** de cualquier etiqueta HTML de la página.

```
var parrafo=document.querySelector("titular"); //selecciona el elemento
alert(parrafo.innerHTML); //muestra Noticia <strong> Destacada</strong>

parrafo.innerHTML=parrafo.innerHTML + "del día";
alert(parrafo.innerHTML);// muestra Noticia <strong> Destacada</strong> del día

parrafo.innerHTML= "Nueva Noticia" //borra el contenido y guarda Nueva Noticia
alert(parrafo.innerHTML);
```

## ► Otros selectores posibles:

```
var enlaces=document.querySelectorAll("#principal a");  
var lista=document.querySelectorAll("header .cursiva");  
var tabla=document.querySelectorAll("table .subrayado");  
var articulos=document.querySelectorAll("div.articulo");  
var h2_derecha=document.querySelectorAll("h2[align='right']");  
var etiquetasvarias=document.querySelectorAll("p,h1,span");  
var grandeImportante=document.querySelectorAll(".grande.importante");
```



- ▶ **EJERCICIO 4:** a partir del fichero pagina.html mostrar en un solo alert la siguiente información:
- ▶ - **Nº de enlaces de la página**
- ▶ - **Nº Enlaces dentro del id noticias**
- ▶ - **Nº Elementos con atributo alt en la página**
- ▶ - **Nº Imágenes de la página**
- ▶ - **Nº Capas con la clase artículo**
- ▶ - **Nº Enlaces dentro de clases artículo**

- **EJERCICIO 5:** Usando el objeto Date, setInterval y con DOM crear una web (dentro de un div) que muestre un reloj digital y que se actualice cada segundo.

