Desarrollo de Aplicaciones Web

HTML DOM

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

HTML DOM

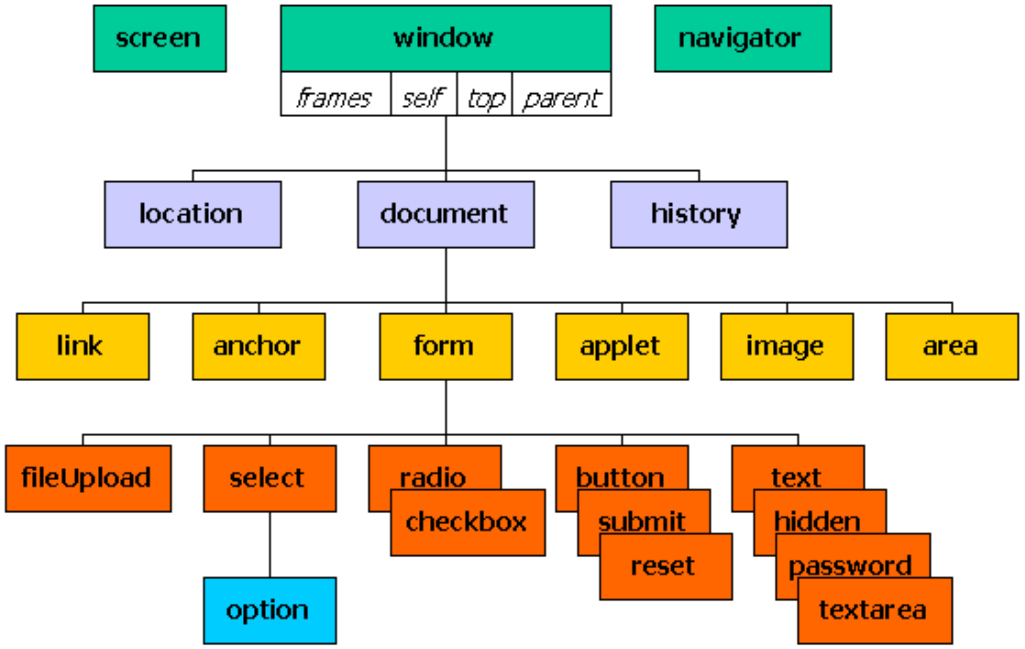
W3C, a través del "*Document Object Model*" (DOM), define un mecanismo estándar de acceso a documentos HTML y XML, permitiendo de esta manera que programas y scripts puedan modificar su contenido, estructura y/o estilo

W3C define tres partes separadas:

- **Core DOM** → modelo estándar para cualquier documento estructurado
- **XML DOM** → modelo estándar para cualquier documento XML
- **HTML DOM** → modelo estándar para cualquier documento HTML

HTML DOM

Mediante JavaScript, es posible tener acceso a los diferentes objetos relacionados con un documento HTML, permitiendo la realización de todo tipo de modificaciones.



HTML DOM

Objeto Screen →Permite obtener información sobre la pantalla en que se está ejecutando el código JavaScript.

Propiedades:

- availHeight
- availWidth
- colorDepth
- height
- pixelDepth
- width

Métodos:

HTML DOM

Objeto Navigator → Permite obtener información sobre el navegador en que se está ejecutando el código JavaScript.

Propiedades:

- appCodeName
- appName
- appVersion
- userAgent
- plugins
- mimeTypes
- …

Métodos:

- javaEnabled()
- …

HTML DOM

Objeto Window → Representa cualquier ventana abierta por el navegador.

Propiedades:

- name
- closed
- length
- self
- parent
- opener
- top
- status
- defaultStatus
- location
- …

Métodos:

- alert()
- setTimeout()
- setInterval()
- clearTimeout()
- prompt()
- confirm()
- blur()
- focus()
- close()
- scroll()
- open()
- …

HTML DOM

Objeto Location → Contiene información referente a la localización del documento que se muestra.

Propiedades:

- href
- hash
- port
- hostname
- host
- protocol
- pathname
- search

Métodos:

- assign()
- reload()
- replace()

HTML DOM

Objeto History → Permite navegar por el histórico de páginas visitadas.

Propiedades:
- length

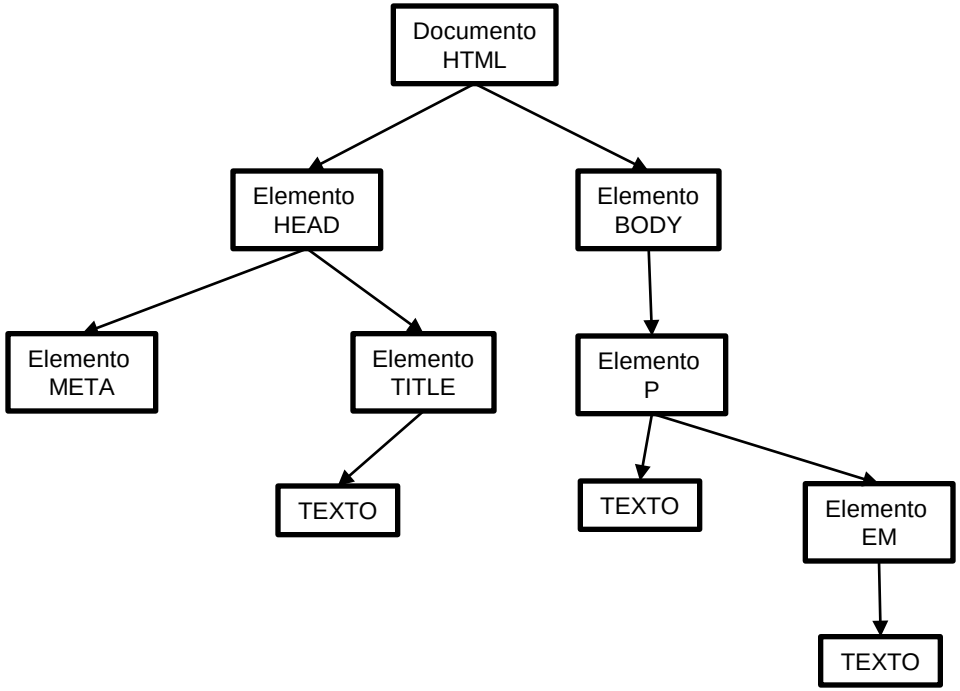
Métodos:
- back()
- forward()
- go()

HTML DOM

Objeto Document → Representa al propio documento que se está mostrando en la ventana del navegador.

Basados en HTML DOM, los navegadores transforman automáticamente las un documento HTML en una estructura arborescente.

```
<html>
  <head>
     <meta  ----------------- />
     <title> ------------------- </title>
  </head>
  <body>
      <p> ------ <em> ----- </em></p>
  </body>
</html>
```



Pablo García Tahoces

HTML DOM

La transformación automática de la página en un árbol sigue las reglas:

- Las etiquetas HTML se transforman en dos nodos:
  - La propia etiqueta.
  - Texto, que aparece como hijo de nodo que representa la etiqueta en cuestión.
- Si una etiqueta HTML se encuentra dentro de otra, se sigue el procedimiento anterior, siendo los nodos generados hijos del que representa a la etiqueta en cuestión.

HTML DOM

HTML DOM define métodos y propiedades para acceder/modificar cada uno de los nodos del árbol generado. Algunos de estos son:

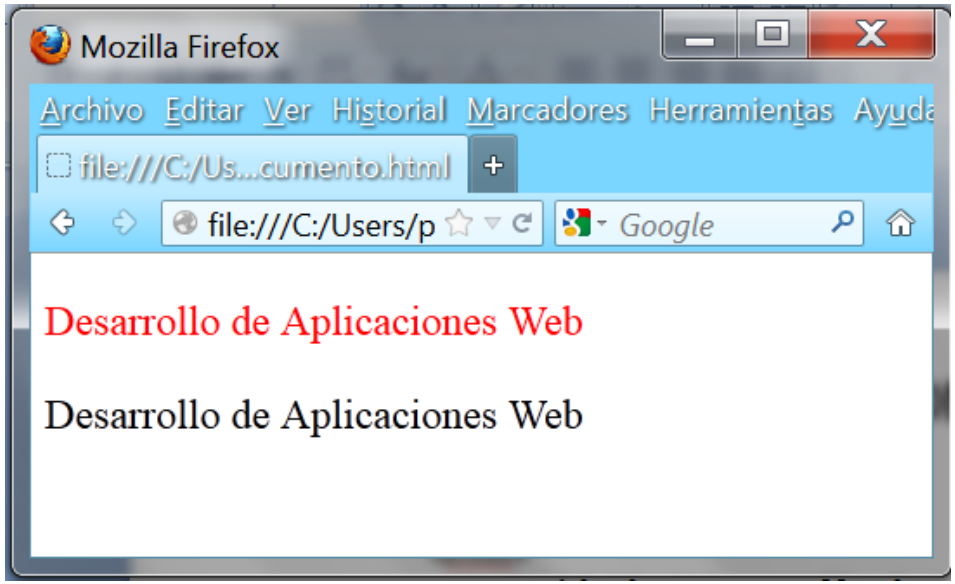
- *getElementByTagName(nombreEtiqueta)* → devuelve un array con todos los elementos de la página cuya etiqueta sea igual al parámetro pasado.
- *getElementByName(nombre)* → devuelve el elemento HTML cuyo atributo name coincide con el parámetro pasado.
- *getElementById(nombre)* → devuelve el elemento HTML cuyo atributo id coincide con el parámetro pasado.
- write(txt) → escribe el texto “txt” en la posición del documento en que ha sido invocado
- innerHTML → propiedad que almacena el contenido de un elemento HTML.
- style → permite modificar el estilo del elemento asociado.

HTML DOM

Ej.-

```
<html>
    <body>
        <p id=“daw">Desarrollo de Aplicaciones Web</p>
        <script>
            var txt=document.getElementById(“daw").innerHTML;
            document.write(txt);
            document.getElementById("daw").style.color="red";
        </script>

    </body>
</html>
```



HTML DOM

HTML5 incorpora nuevos métodos para realizar la selección de los nodos DOM, basada en selectores de las reglas CSS:

- *querySelector(CSS_Selector)* → devuelve el primer elemento de la página coincidente con el parámetro pasado.
- *querySelectorAll(CSS_Selector)* → devuelve todos los elementos de la página coincidententes con el parámetro pasado.

Ej.-

```
var x= document.querySelector(“.usoClass1”, “usoClass2”);

var y= document.querySelector(“a:visited”);
```