

YUSEF HASSAN MONTERO
FRANCISCO JESÚS MARTÍN FERNÁNDEZ

Introducción a DOM

Todos los derechos reservados
www.nosolousabilidad.com

INTRODUCCIÓN

DOM, Document Object Model o Modelo de Objetos de Documento, es una interfaz estándar que posibilita a scripts y programas, y de forma dinámica, el acceso y manipulación de la estructura, estilo y contenido de los documentos.

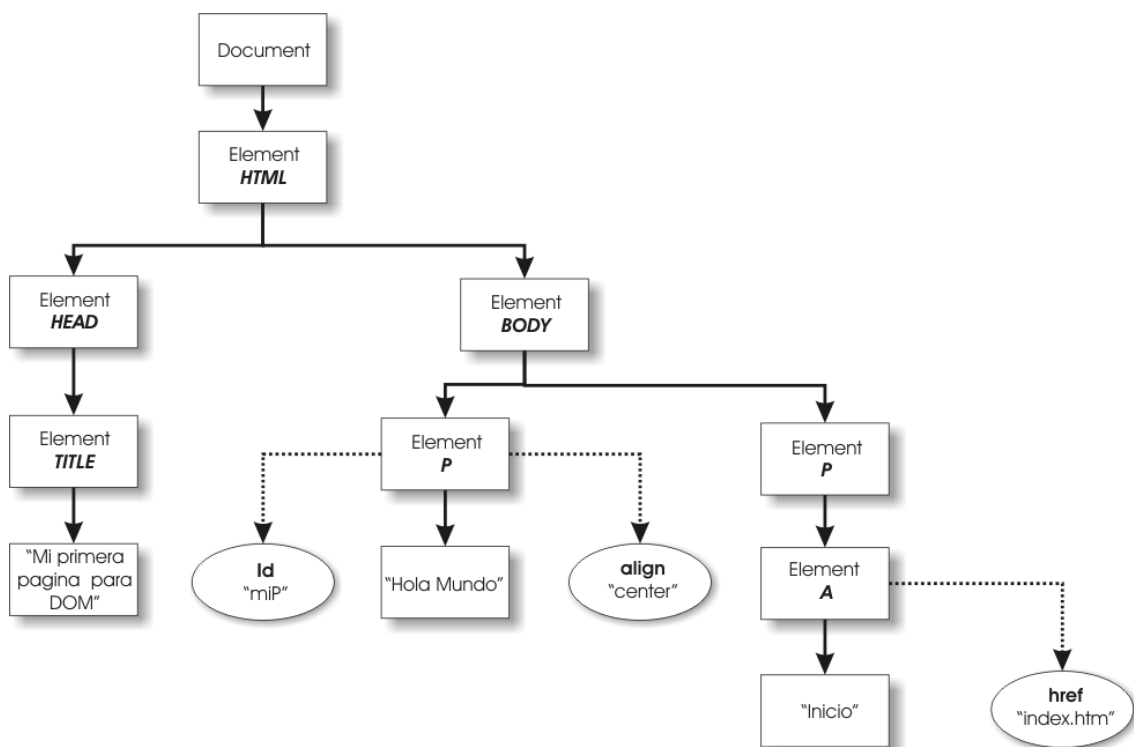
En este documento vamos a tratar el acceso y manipulación de documentos HTML con JavaScript y a través del estándar DOM2. Esto significa que muchos de los métodos y propiedades que vamos a utilizar no están soportados por algunos navegadores antiguos, aunque al tratarse DOM de un estándar, los actuales navegadores más extendidos, al igual que (presuponemos) futuros navegadores, no tendrán problemas en interpretar correctamente el código.

EMPEZANDO...

Cuando el navegador carga una página HTML, crea internamente una representación del documento en forma de árbol jerárquico. En el siguiente ejemplo de página HTML:

```
<HTML>
  <HEAD>
    <TITLE>Mi primera página para DOM</TITLE>
  </HEAD>
  <BODY>
    <P id="primera" align="center">Hola Mundo!</P>
    <P id="segunda">
      <A href="index.htm">Inicio</A>
    </P>
  </BODY>
</HTML>
```

El árbol generado internamente por el navegador sería como este:



Como podemos observar en el grafo, el documento está compuesto por una serie de nodos relacionados jerárquicamente. Entre los nodos podemos diferenciar cuatro tipos diferentes: Nodo de Documento (document), Nodos de Elementos (element), Nodos de Texto (“.”) y Atributos.

Los Elementos se corresponden con las etiquetas del documento HTML. Los Nodos de Texto con las cadenas de caracteres (encerradas entre etiquetas), y los Atributos con los atributos de las etiquetas. El nodo de documento representa el documento HTML en sí.

Además, esta jerarquía establece implícitamente relaciones entre los nodos:

Padre-Hijo: Por ejemplo, el elemento HTML es padre del elemento HEAD, que a su vez lo es del elemento TITLE. Un padre puede tener muchos hijos, aunque un hijo únicamente un padre.

Hermanos: Dos elementos se consideran hermanos cuando son hijos del mismo padre. En nuestro ejemplo, los dos elementos P son hermanos, ya que ambos son hijos de BODY.

Tipos de Nodos

Ya hemos comentado que hay diversos tipos de nodos, pero veamos un poco más específicamente sus características.

Nodos de Elemento

En un documento HTML se definen a través de etiquetas. Pueden ser hijos de otro nodo elemento, y también tener hijos (nodos elemento o nodos de texto). Su acceso y manipulación se realiza a través de la interfaz Node y el objeto document.

Nodos de Texto

Estos nodos siempre son hijos de un nodo elemento, y no pueden tener hijos. Su acceso y manipulación se realiza a través del nodo elemento del que son hijos.

Nodos de Atributo

Al igual que los de Texto, son siempre hijos de un nodo elemento, y no pueden tener hijos. Su acceso y manipulación se realiza a través del nodo elemento del que son hijos.

Nodo de Documento

Es un nodo único y diferente del resto. No tiene padre y únicamente tiene un hijo, el nodo elemento raíz, en un documento HTML es el elemento HTML. A través de los métodos y propiedades del objeto document podemos crear y acceder a nodos elemento.

Resumiendo:

Para acceder a un nodo elemento, lo hacemos desde el nodo documento o desde otro nodo elemento.

Para acceder a un nodo de texto o un nodo de atributo, lo hacemos desde el nodo elemento del que desciendan.

DOM ofrece una serie de interfaces, objetos, propiedades y métodos para recorrer, manipular, eliminar y crear nodos en el árbol del documento, que se muestra en la siguiente sección de referencia.

REFERENCIA

DOCUMENT

El acceso a nodos del documento:

`getElementById()`

Acceder a un elemento del documento a través de su atributo id.

`getElementsByTagName()`

Acceder a un conjunto (array) de elementos a través del nombre de su etiqueta.

`documentElement`

Accede al nodo raíz, es decir, al primer hijo de 'document', en el caso de un documento HTML este elemento se corresponde con el elemento HTML (<HTML>...</HTML>).

Creación de nuevos nodos en el documento:

`createElement()`

Crear un nuevo nodo de tipo elemento en el documento.

`createTextNode()`

Crear un nuevo nodo de tipo texto en el documento.

NODE

Acceso a nodos cercanos:

firstChild

Acceso al primer hijo del nodo.

lastChild

Acceso al último hijo del nodo.

childNodes

Acceso al conjunto (array) de hijos del nodo.

parentNode

Acceso al padre del nodo.

nextSibling

Acceso al siguiente hermano del nodo.

prevSibling

Acceso al anterior hermano del nodo.

Obtención de información sobre los nodos:

tagName

Nos informa del nombre de la etiqueta del nodo (en nodos de tipo Elemento).

nodeValue

Nos informa del valor del nodo (en nodos de tipo Texto).

hasChildNodes()

Nos devuelve true o false en caso de que tenga o no hijos el nodo.

Manipulación de los atributos de un nodo:

getAttribute()

Devuelve el valor del atributo indicado.

setAttribute()

Especifica un atributo para el nodo.

Añadir o eliminar nodos:

appendChild();

Añade un nodo hijo al nodo actual.

removeChild();

Elimina un nodo hijo al nodo actual.

ESTILO CSS

Tabla de correspondencia entre propiedades CSS y propiedades de 'style' en DOM.

Propiedades CSS	Style
background	background
background-color	backgroundColor
border	border
border-bottom, border-bottom-color, border-bottom-style, border-bottom-width	borderBottom, borderBottomColor, borderBottomStyle, borderBottomWidth
border-color	borderColor
border-left, border-left-color, border-left-style, border-left-width	borderLeft, borderLeftColor, borderLeftStyle, borderLeftWidth
border-right, border-right-color, border-right-style, border-right-width	borderRight, borderRightColor, borderRightStyle, borderRightWidth
border-style	borderStyle
border-top, border-top-color, border-top-style, border-top-width	borderTop, borderTopColor, borderTopStyle, borderTopWidth
border-width	borderWidth
bottom	bottom
clip	clip
color	color
display	display
font, font-family, font-size, font-style, font-variant, font-weight	font, fontFamily, fontSize, fontStyle, fontVariant, fontWeight
height	height
left	left
letter-spacing	letterSpacing
line-height	lineHeight
list-style, list-style-type, list-style-image, list-style-position	listStyle, listStyleType, listStyleImage, listStylePosition
margin, margin-left, margin-top, margin-right, margin-bottom	margin, marginLeft, marginTop, marginRight, marginBottom
overflow	overflow
padding, padding-left, padding-top, padding-bottom, padding-right	padding, paddingLeft, paddingTop, paddingBottom, paddingRight
position	position
right	right
text-align, text-indent, text-decoration, text-transform	textAlign, textIndent, textDecoration, textTransform
top	top
visibility	visibility
width	width
z-index	zIndex