## 2. Introducción al AOM (Accessibility Object Model) o Modelo de Objetos de Accesibilidad

### 2.1 Introducción al DOM

Para comprender el Modelo de Objetos de Accesibilidad (AOM), se necesita unos conocimientos previos del DOM. El DOM, también se conoce como el Modelo de Objetos del Documento, como la definición presentaba en el Capítulo 1, es un modelo que permite crear página web dinámica con HTML, CSS y JavaScript, con el uso de dicho modelo, a través de la programación de JavaScript, puede acceder y modificar el contenido de la página.

DOM no se indica a una tecnología nueva, sino una especificación estándar de W3C.[[1]](#footnote-1), es decir, en la industria del desarrollo web, entre los proveedores y compradores y usuarios de los productos y servicios web, se llegan a un acuerdo de todos sus requisitos.

Es también conocido como un API o Interfaz de Programación de Aplicación, en la ciencia de computación, se refiere a un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.[[2]](#footnote-2)

De una manera más fácil, puede entender el API como un estándar que se crea para facilitar la comunicación y cooperación de diferentes aplicaciones, sus equivalentes en el mundo real puede ser como el Código Morse, Tiempo universal coordinado o Tabla periódica de los elementos. (Jeremy Keith, 2005)

En el sentido de la web, DOM es un API de W3C como su definición: “DOM es una plataforma e interfaz de lenguajes neutrales (puede ser JavaScript, también PHP u otros lenguajes de programación) que permite las programas y scripts pueden acceder y renovar dinámicamente el contenido, la estructura y el estilo de un documento.” Se dividen en 3 partes diferentes[[3]](#footnote-3):

1. Core DOM: modelo estándar para todos tipos de documentos.
2. XML DOM: modelo estándar para documentos de XML.
3. HTML DOM: modelo estándar para documentos de HTML.

#### 2.1.1 El árbol de DOM

En el DOM, los documentos (***document***) HTML y los elementos (***element***) dentro de ellos son definidos como objetos (***object***), esto quiere decir que tienen determinadas propiedades (***property***) y métodos (***method***) destinados a ellos, puede asignar un valor a tal propiedad suya o hacerlos realizar una acción.(Puede ver todas las propiedades y métodos del objeto de los documentos en la web de W3C)[[4]](#footnote-4) Cuando un documento HTML es cargado dentro de un navegador web, se convierte a un objeto de documento, en este proceso, el DOM API sirve como su interpretador para que el navegador pueda entenderlo. Todos estos objetos en DOM son objetos nodos (***nodes***) y se distribuyen en forma de un árbol.

Los nodosen sentido general, puede ser los dispositivos en una red grande, por ejemplo, los PCs, móviles incluso impresores, o sea cualquier terminal que lleva una dirección IP. En la ciencia de computación, un nodo es una parte individual de una estructura de datos más complejas, que contiene una cantidad de información al mismo tiempo representa su relación con otros nodos (el nodo que se ubica más alto es el padre (***parent***) y el que se ubica debajo de él es el hijo (***child***)). La estructura de datos que se contribuyen por los nodos es normalmente un árbol binario, es decir, cada nodo solo puede tener dos hijos.[[5]](#footnote-5)

El árbol de DOM es un árbol binario de nodos, donde el mismo documento HTML es el ***root node***, el origen o dueño de todos restos de nodos, los elementos de HTML son objetos nodos que llevan varias propiedades y se puede aplicar métodos a ellos, al mismo tiempo puede ser definido por su relación con otros elementos como padre o hijo. No todos los nodos tiene hijos, algunos solo tienen padre, estos son ***leaf nodes***, otros pueden compartir un padre, son ***Sibling nodes***. Cada nodo en DOM lleva una propiedad de NodeType (tipo de nodo) para mostrar el tipo de interfaz y su relación con otros en el árbol, los tipos de nodos más usados en la maquetación web son como: nodo de elemento (ELEMENT\_NODE), nodo de atributo (ATTRIBUTE\_NODE) y nodo de texto (TEXT\_NODE), véase las capturas siguientes de un ejemplo del árbol de DOM. Realmente hay más tipología de nodos en DOM, puede consultar en el estándar de DOM publicado en la web de W3C[[6]](#footnote-6), en el ejemplo por motivo de simplificar la explicación solo usan los tres primeros.

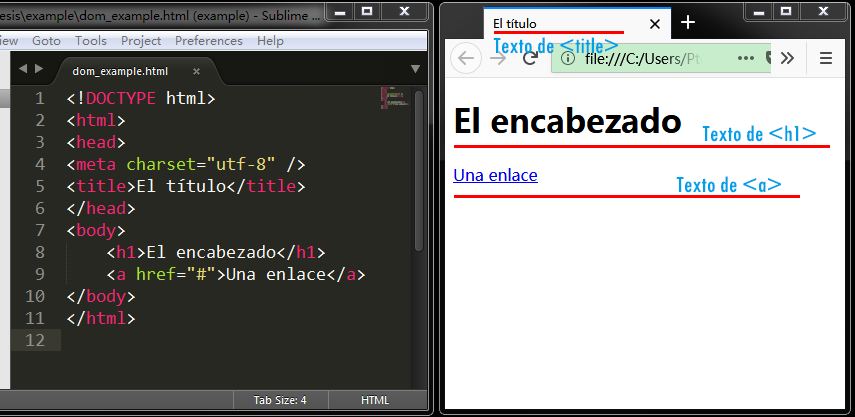


Figura 1. Código HTML e interfaz de usuario.

En la figura 1, se presenta el código de un documento HTML para la maquetación de una página web muy simple de ejemplo, el lado derecho es la interfaz de usuario (UI) que se presenta en el navegador web. Se puede ver el texto de título de la página “El título” dentro de las etiquetas del elemento <title> se presenta en la pestaña de la parte arriba del navegador, de la misma manera, el texto del encabezado y la enlace de las etiquetas del elemento <h1> y el <a> también se muestra en la parte principal del navegador con distintas apariencias, o sea, con interfaz diferente.

En el caso del elemento <meta>, es para añadir metadatos al documento HTML, datos para describir brevemente el contenido de la página, las palabras claves, el autor, etc. El metadato no se presenta en la UI, pero puede ser usado por los navegadores a la hora de presentar el contenido de la página. El funcionamiento de este elemento normalmente acompaña con varios tipos de atributos para proporcionar información adicional, en general vienen en forma de: nombre: “valor”. En el ejemplo, el elemento <meta> lleva un atributo “charset”, conjunto de caracteres, que representa el tipo de estándar de codificación de caracteres, lo asignamos un valor: “UTF-8”, UTF-8 (o Unicode) cubre casi todos los caracteres y símbolos en el mundo. Básicamente lo hemos hecho para informar al navegador que presenta también caracteres fuera del idioma de inglés, como el acento en el texto “El título”. Además, el elemento <a> también lleva un atributo “href” para indicar la dirección que va dirigir la enlace, en el ejemplo, asignamos un valor “#” para enlazar a la propia página.

El árbol de DOM de este documento HTML y las relaciones entre los objetos nodos se presentan en la siguiente figura.

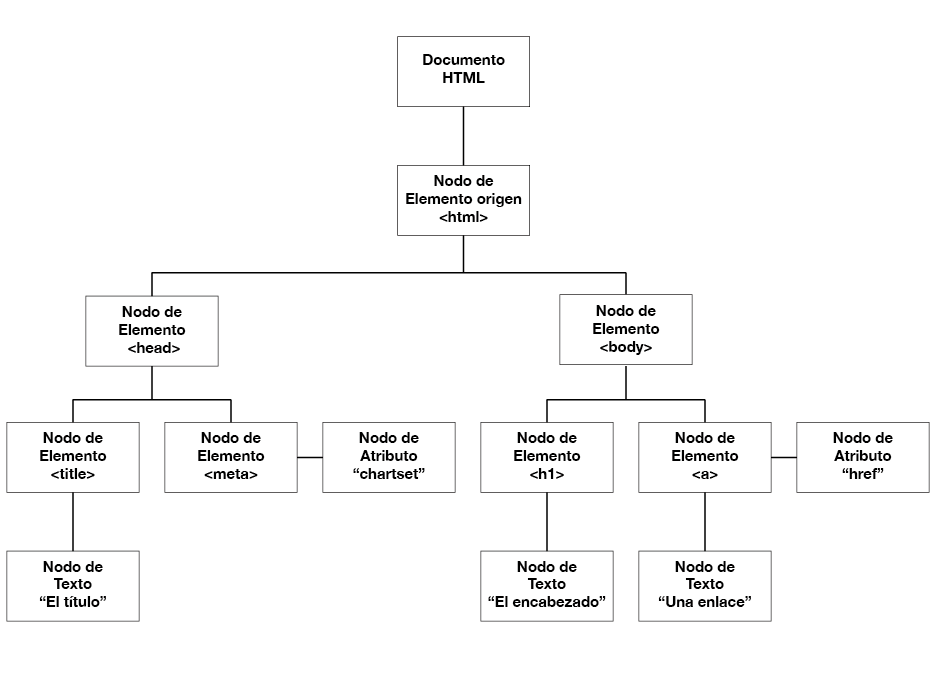


Figura 2. El árbol de DOM del documento HTML de ejemplo

Con el árbol, los objetos nodos y la relación entre ellos se ve muy claro a la vez la estructura del documento: con el documento HTML como el nodo de origen, surgen varios nodos con propiedad e interfaz diferente, como los tres principales mencionamos antes:

* Los nodos de elemento: <html> <body> <head>, etc.
* Los nodos de atributo: charset, href.
* Los nodos de texto: “El título”, “El encabezado”, “Una enlace”.

Además se muestra una estructura jerárquica entre ellos, el elemento <html> es el padre de todos sus hijos, como los elementos <body> <head>, vienen de <html> al mismo tiempo tienen sus hijos también, en este caso, <title> y <meta> son hijos de <head>, del mismo modo, <h1> y <a> son hijos de <body>.

Insertamos pocos códigos de JavaScript al documento HTML aplicando unos métodos del DOM (todos los métodos y propiedades en DOM[[7]](#footnote-7)) para mostrar estas relaciones en la consola del navegador.

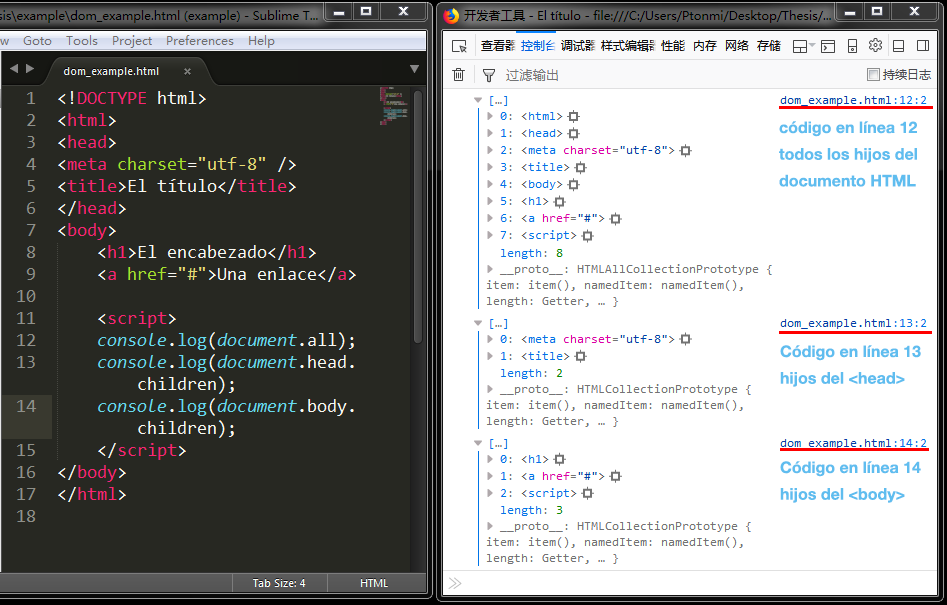


Figura 3. Mostrar la estructura de DOM en la consola del navegador

El DOM del este documento HTML no solo muestra relaciones jerárquicas entre los nodos de elemento, también incluye nodos de atributo y nodos de textos, desde el nodo de origen, ya podemos navegar entre estos nodos.



En el ejemplo anterior, hemos añadido unas líneas de JavaScript para mostrar un ejemplo de ambos nodos de texto y nodos de atributos. Debajo del código es la presentación en la consola, se encuentra que el atributo del elemento <meta> es “charset” y el contenido textual del elemento <title> es “El título”.

El lado derecho muestra el camino o los pasos que lleva para localizar y presentar los dos nodos. Primero creamos dos variables con nombre “meta” y “titulo”, según hemos presentado, el documento HTML es el nodo de origen, el padre de todos los elementos, en el ejemplo anterior (figura 3), se puede observar que hay 8 nodos de elemento en total en el documento, <meta> es con índice [2] y <title> es [3], así asignamos los dos nodos de elementos a las variables, y desde los dos nodos puede encontrar el nodo de texto y el nodo de atributo que están relacionado con ellos.

Cabe mencionar que el nodo de texto que usamos en el ejemplo no posee una posición jerárquica igual que el nodo de atributo. La relación entre el nodo de elemento <title> y el nodo de texto “El título” es padre e hijo, mientras el nodo de atributo “charset” está ubicada en el mismo nivel junto con el nodo de elemento <meta>, aunque los dos son tipos de nodo distintos. Por lo que coincide a la estructura en la figura 2.

#### 2.1.2 Los Eventos en DOM

1. “W3C Document Object Model,” 3 de enero, 2018, <https://www.w3.org/DOM/>. [↑](#footnote-ref-1)
2. “Interfaz de programación de aplicaciones,” Wikipedia, la enciclopedia libre, 22 de diciembre, 2017, <https://es.wikipedia.org/w/index.php?title=Interfaz_de_programaci%C3%B3n_de_aplicaciones&oldid=104391567>. [↑](#footnote-ref-2)
3. “JavaScript HTML DOM,” 3 de enero, 2018, <https://www.w3schools.com/js/js_htmldom.asp>. [↑](#footnote-ref-3)
4. “HTML DOM Document Objects,” 4 de enero, 2018, <https://www.w3schools.com/jsref/dom_obj_document.asp>. [↑](#footnote-ref-4)
5. “Binary Trees, Data Structures and Algorithms,” 9 de enero, 2018, <http://www.allisons.org/ll/AlgDS/Tree/>. [↑](#footnote-ref-5)
6. “Estándar DOM - 4.4 Interfaz de Nodo” 10 de enero, 2018, <https://dom.spec.whatwg.org/#dom-node-nodetype> [↑](#footnote-ref-6)
7. “JavaScript DOM Document,” 10 de enero, 2018, <https://www.w3schools.com/js/js_htmldom_document.asp>. [↑](#footnote-ref-7)