INTRODUCCIÓN AL DOM

Los sitios web han cambiado mucho desde que inició el siglo. Las viejas colecciones de páginas de texto básico evolucionaron a una red de experiencias planeadas meticulosamente, completadas con elementos como botones responsivos, desplazamiento en paralaje y contenido personalizado.

Desde luego, estas características del diseño web no surgen de la nada. Estas técnicas suelen ser bastante complejas de implementar, y la mayoría de las veces requieren diferentes archivos y lenguajes de programación que trabajen en conjunto para entregar una experiencia consistente.

Para crear páginas web bien diseñadas y atractivas, los archivos de tu sitio necesitan acceso entre ellos, y el modelo de objeto de documento (document object model o DOM, por sus siglas en inglés) lo hace posible. En esta guía explicaremos qué es DOM y lo que los propietarios de sitios deberían conocer sobre este componente clave para la experiencia web.

Antes de comenzar, te recomendamos familiarizarte con lo básico de HTML, CSS y JavaScript.

¿Qué es el DOM?

El modelo de objeto de documento (DOM) es una interfaz de programación para HTML. Traduce el contenido de un documento HTML a un objeto estandarizado, al que los lenguajes de programación funcionales como JavaScript tienen facilidad de acceso y modificación. Debido a que la mayoría de los eventos de páginas web son impulsados por código que no es HTML, todas las páginas web dinámicas dependen del DOM para visualizarse y funcionar correctamente.

Para comprender mejor por qué el DOM es útil, hablemos primero de lo que significa «objeto» en la programación computacional.

¿Qué es un objeto?

Si lo resumimos, la tarea principal del software de una computadora es realizar acciones con datos. Los datos pueden ser de diferentes tipos: números, caracteres y palabras, por mencionar algunos.

Cuanto más haga el programa, más datos necesita procesar y manejar. Sin un sistema escalable y bien organizado, la complejidad creciente causará errores.

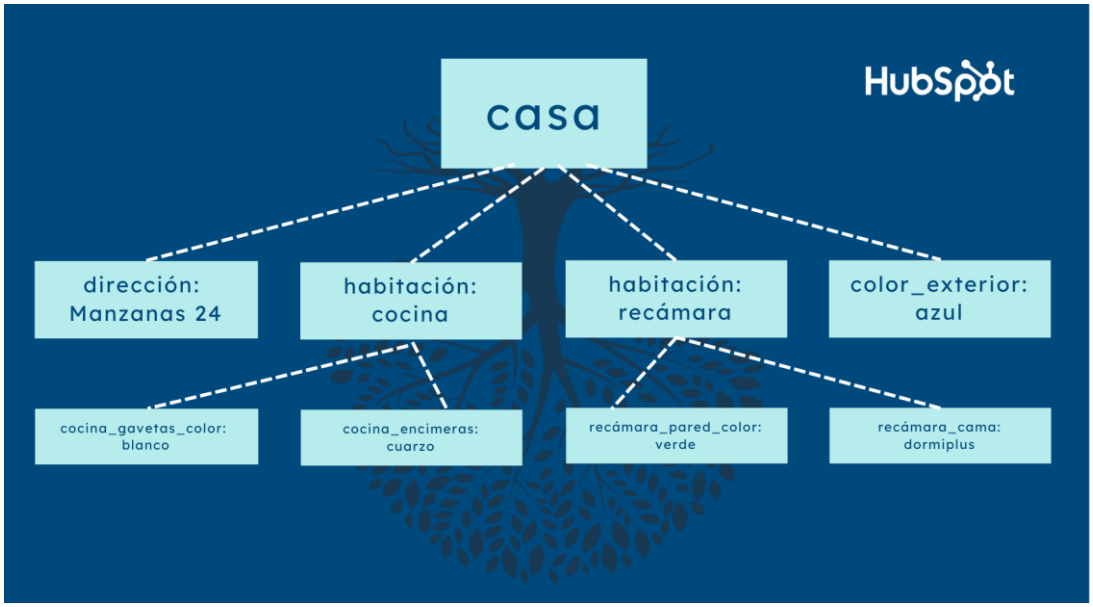
Así que para organizar piezas relacionadas de datos, los programadores usan un tipo de datos llamado «objeto». Los objetos se diferencian de otros tipos de datos porque su propósito es sostener a otros. Un objeto contiene piezas de datos relacionados bajo un concepto común, guardado en una jerarquía.

Este es un ejemplo: supón que estás construyendo una pieza de software que guarda información acerca de casas. En la vida real, las casas contienen muchas cosas: habitaciones, muebles, electrodomésticos y gente, sin mencionar información relevante como el vecindario, dirección y número de plantas.

Si deseas representar una variable de casa en un programa de software, no tendría sentido guardar todos estos datos relacionados con el tema de forma separada. En lugar de eso, creas el objeto «una casa» para cada casa en el sistema, y pones todas las cosas vinculadas a ella. Una «casa» puede contener datos para todos los elementos que posee.

Incluso puedes poner objetos dentro de otros objetos. En nuestro ejemplo, un objeto «casa» podría contener objetos «habitación». Puedes designar un objeto habitación como cocina, que a su vez contiene datos sobre las gavetas, encimeras, etc.

Aquí lo principal es que los objetos agrupan datos en una manera lógica y jerárquica. Para entenderlo mejor, imagina un árbol invertido. El nombre del objeto es como la raíz del árbol y todas sus ramas son los datos guardados dentro de ella. He aquí nuestro objeto casa:



¿Por qué necesitamos el DOM?

Antes de que te entusiasmes creando árboles y raíces, relacionemos el concepto de objeto al HTML. Mira este archivo HTML básico:

HTML

```html
<html>
  <head>
      <tittle>Mi Página Web</tittle>
  <head/>

      <body>
          <h1>Mi Página Web</h1>
          <h2>¡Bienvenidos a Mi Página Web!</h2>
          <p>Es genial tenerte de visita.</p>
          <img src="hola.jpg" alt="un gato levantando la patita delantera" width="1000" height="1000">
          <p>¡Haz clic en el botón de abajo para una sorpresa divertida!</p>
          <button>Clic aquí</button>
      </body>
      <ul id="list">
</html>
```

Observa que las partes también están estructuradas en una jerarquía. Dentro de nuestro HTML «casa», tenemos nuestras etiquetas <head> y <body>, que funcionan como habitaciones. Finalmente, existen los elementos más específicos: <title>, <h1>, <h2>, <p>, <img> y <button>, piensa en ellos como elementos particulares de una habitación, como la cama o una lámpara.

Los elementos en un documento HTML están estructurados de esa forma por la misma razón que los objetos: una estructura jerárquica ayuda a los programas de computadora a leer y procesar la información. En el caso de HTML, tu navegador web es el programa, y los lenguajes de secuencia de comandos (como JavaScript) son lo que cambia y modifica el código HTML.

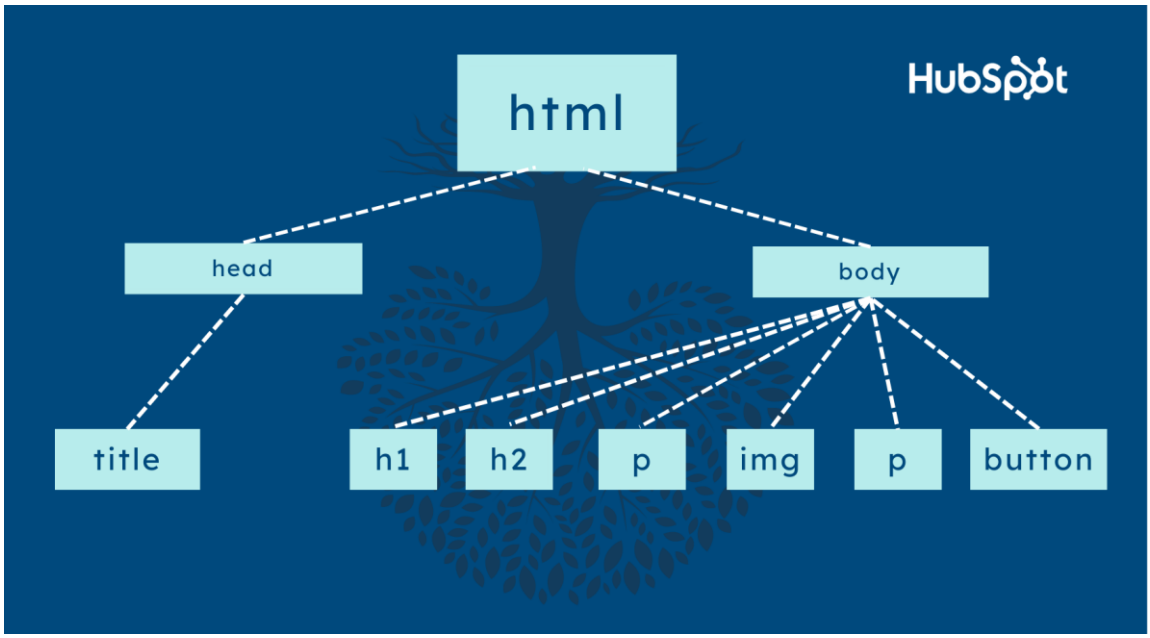
Pero existe un problema. Los documentos HTML por sí mismos no son objetos. No pueden ser leídos por JavaScript sin una especie de traducción archivo-a-objeto. Aquí es donde el DOM se involucra.

El DOM representa un documento HTML completo como un solo objeto. Para crear el DOM, el navegador web que lee el archivo HTML toma todas sus partes, desde el elemento <html> raíz hasta las etiquetas más pequeñas, y las devuelve como un objeto que el JavaScript comprende.

La palabra «representar» es importante en esta parte. El DOM no es una copia del archivo HTML, es simplemente una manera distinta en que el navegador web organiza la información HTML.

La forma en que el navegador crea el DOM es similar a cómo configura una página web. Cuando abres cualquier página en tu navegador, ves la representación visual del HTML subyacente. Ves el mismo contenido, pero organizado como una página que tu cerebro puede entender fácilmente. El DOM es otro medio para que tu navegador represente el HTML. En el DOM, el HTML está organizado en un objeto que el JavaScript puede entender sin problemas.

¿Recuerdas el ejemplo del árbol de más arriba? Así es como el DOM representaría nuestro simple archivo HTML como un objeto JavaScript:



Una cosa más que vale la pena resaltar antes de seguir con el tema: el DOM no es exclusivo de JavaScript. Cualquier lenguaje de programación (como Python, C++) puede utilizar el DOM para modificar páginas web. Sin embargo, JavaScript afecta casi todas las páginas que vemos en internet, así que es el único lenguaje de programación que necesitas conocer para entender el DOM.

Ya hemos hablado de árboles, ramas, raíces, casas y habitaciones, pero ¿qué tan relevante es todo eso para crear sitios web?

¿Qué es el DOM en el diseño web?

En el diseño web, JavaScript se utiliza para controlar el comportamiento de las páginas. El DOM vincula JavaScript al código HTML fuente, lo que permite que JavaScript ejecute sus funciones en elementos HTML individuales. Esta interacción JavaScript-HTML crea la experiencia de un diseño web interactivo.

Los objetos no solo son buenos para organizar datos, sino que también permiten que los programas tengan fácil acceso a piezas específicas de información en ellas.

De regreso a nuestro objeto casa, digamos que quieres editar el elemento «encimera» de una casa en particular. Para hacerlo, tu programa necesita encontrar la casa en cuestión, recorrer el objeto árbol hacia el objeto cocina, luego cambiar el elemento «encimera» de «cuarzo» a «mármol». Linda elección.

Del mismo modo, el DOM le da a JavaScript acceso a casi cualquier elemento en un documento HTML, desde la página entera hasta una pequeña línea de texto.

Cuando algo ocurre en una página web después de que se cargue (por ejemplo, dar clic a un botón abre un formulario o hace que un elemento cambie de color o tamaño), es el código JavaScript que apunta a un elemento específico a través

del DOM y hace el cambio. El código JavaScript puede localizarse en su propio archivo .js o dentro del propio documento HTML.

Para un ejemplo básico del DOM en acción, prueba esta demostración de W3Schools. El panel de la izquierda contiene el código fuente, y el de la derecha muestra el código como se ve en la página web. En el instante que presionas en ¡Haz clic aquí!, el siguiente JavaScript corre.

document.getElementById('demo').style.display='block'

En el código anterior:

- **document** es la forma en que JavaScript apunta al objeto archivo HTML a través del DOM. Le dice al navegador «quiero ir al archivo HTML a realizar un cambio».
- **.getElementByID('demo')** apunta al elemento de la página con el ID «demo». Este es el elemento que JavaScript desea modificar. Para llegar ahí, JavaScript comienza en la etiqueta <html> más alta, luego se mueve a través del DOM hasta que llega al elemento con el texto «¡Hola, JavaScript!»
- **.style.display='block'** es la acción. Una vez que el elemento al que se apunta es encontrado, revela el texto.

Las páginas web complejas ejecutan estos llamados cientos de veces (a medida que los usuarios interactúan con ellas). JavaScript utiliza el DOM en cada una de esas ocasiones.

El DOM: manteniendo interesante al internet

Ahora puedes notar por qué el modelo de objeto de documento es tan importante en los sitios web modernos. Sin él, JavaScript no sería capaz de alterar páginas de formas sorprendentes, sino que tendríamos un montón de páginas aburridas y estáticas.

Con suerte, nunca tendrás que preocuparte por programar un DOM, ya que los navegadores web hacen todo el trabajo por ti. Aun así, es útil que conozcas su funcionamiento, ya sea que pruebes nuevos efectos de página, requieras un rediseño de web, necesites eliminar fallos rápidamente o si simplemente deseas saber qué sucede tras bambalinas.