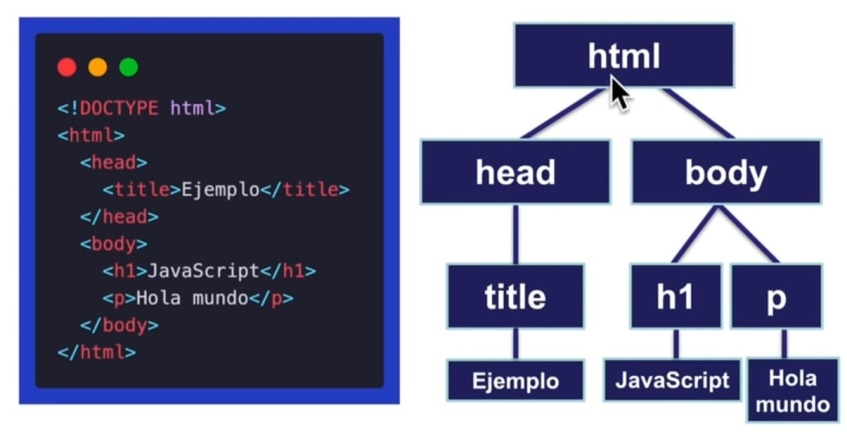
**5/9/2023-5/12/2023 Token github: ghp\_RHoOWifuu1CzmYBpaUOSlTIuDyn5p63LfOie**

**-Curso-JavaScript-DOM- 10/08/2023**

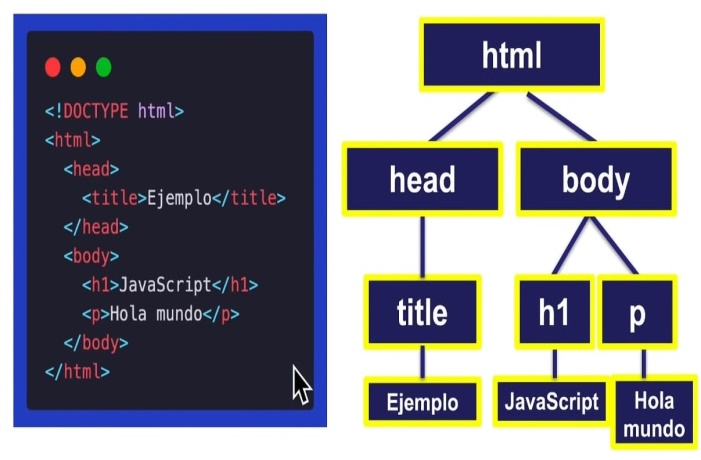
Que es el DOM? Ese es el modelo que vamos a usar durante el curso y está estrechamente relacionado con el desarrollo web, nos va a permitir crear páginas interactivas que necesitamos. Cuando se crea una página web normalmente se crea un archivo HTML para definir la estructura de la página. Esa estructura en el navegador está representada también por una jerarquía. Un elemento está dentro de otro elemento y esos elementos se van a poder representar como ciertos obj en nuestro programa. Toda esa jerarquía de distintos elementos que se van agrupar y estructurar para poder mostrarlos en el navegador es lo que conocemos como DOM pero se va creando en base a lo que definimos en el archivo HTML. El navegador crea una representación de c/u de esos elementos y su ubicación en una jerarquía especifica. El significado de DOM→Modelo de Objetos del Documento. Formal mente podemos definir el DOM como la representación de los obj, los distintos elementos que conforman la estructura de un documento en la web.

A partir de estos elementos podemos generar un diagrama similar a este que representa su jerarquía HTMT



En el grafico vemos al inicio la etiqueta <html> que es el elemento raíz a partir de ahí continúan todos los otros elementos que van hacer descendientes del elemento principal. Luego tenemos <head> y <body> que están en la misma línea, esto nos dice que son hermanos porque están ubicados en el mismo nivel en la jerarquía. Seguidamente dentro de <head> encontramos el elemento <title> y por debajo tenemos otro nivel que representa el texto en sí. El DOM llega hasta ese ínfimo detalle de representar el texto de forma individual. Con <body> ocurre lo mismo, pero en este caso <body> contiene dos elementos. Un elemento <h1> y un elemento <p> están al mismo nivel en la jerarquía pero están contenidos en el elemento <body> y así se puede ir leyendo este diagrama hasta que llegamos al texto que corresponde a cada uno si es que contienen textos. Por ejemplo si tenemos un elemento de imagen no tendríamos texto, esto es básicamente la idea del diagrama. ¿Porque es importante representar el documento a través del DOM? Representar el documento HTML a través del DOM nos permite *acceder* a sus elementos en nuestro programa usando JS y también *manipularlos* que significa crear elementos nuevos, eliminar elementos que ya están en la página, modificar sus atributos, agregar atributos nuevos, cambiar el texto. Todas esas operaciones mencionadas son ejemplos de cómo podemos manipular elementos una vez que ya están representadas en el DOM en el navegador. Es uno de los aspectos más importante que se tiene que saber. El DOM nos permite representar los elementos como *NODOS* y *OBJETOS* con los cuales podemos trabajar *JS* con sus propiedades, métodos y con distintos elementos. Por ej. cuando se hace clik sobre un botón todos esos eventos vamos a poder manejarlos y detectarlos en nuestro código gracias al DOM porque en si nuestro archivo HTML es solo eso un archivo pero el DOM es el que le da esa capacidad poderosa para poder trabajar con esos elementos en JS. hay que recordar estos dos conceptos *NODO y OBJETOS* son muy importantes para el DOM los Nodos son muy utilizados y prácticamente fundamentales en el área de estructuras de datos. También en el DOM vamos a tener Nodos pero esos Nodos en lugar de representar información como tal van a representar elementos con todos sus datos y toda esa información y los objetos van a estar presente cuando trabajemos con esos elementos en JS, van a tener propiedades y métodos.

Un Nodo es un punto específico del diagrama o del árbol de Nodos del DOM por ejemplo en el siguiente grafico tenemos la estructura del archivo HTML y el mismo diagrama anterior pero ves que todos tienen un borde amarillo es para indicarles que todos estos elementos son Nodos incluyendo el texto. Individualmente el texto en el DOM se representa como un Nodo y eso va ser importante cuando trabajemos con algunos métodos que nos puede permitir acceder a un elemento o a un Nodo como tal. Los Nodos en el DOM incluyen: El documento en sí, los elementos HTML que definamos en nuestro archivo, también cualquier texto que incluyas entre etiquetas se representa como un Nodo y también algo importante es que los comentarios que escribas en el archivo html también se representan como Nodos. Todo el proceso de transformar un archivo html a un DOM interactivo que podemos usar en nuestro archivo JS lo realiza el navegador o borwser en inglés.



**Conceptos Importantes:**

Al referirte al DOM y a los elementos representados en el dom dentro de su jerarquía, al igual que un árbol genealógico vamos a tener esos elementos representados como una jerarquía. Cual de esos elementos contiene a otros y cuales están contenidos dentro de otros elementos todo eso lo vamos a poder representar y cada tipo de Nodo va a tener un concepto diferente para referirnos a esa ubicación.



Por ejemplo como vemos en el diagrama lo primero que conseguimos en nuestra jerarquía en html es el Nodo raíz es el que contiene a todos los otros elementos. Luego comenzamos hablar sobre las distintas relaciones que puede haber. Primero tenemos parent node o (nodo padre) este nodo es el que contiene a otro nodo en la jerarquía del DOM.



En el gráfico de arriba tenemos señalado el elemento *head* ese nodo es el nodo padre de *title* porque esta contenido por *head* lo mismo ocurre con *body* es el nodo padre de *h1* y de *p.* Eso también nos lleva a que tenemos otra relación reciproca si un Nodo es el nodo padre el otro nodo contenido por este es el nodo hijo. Digamos entonces que el nodo hijo es el que esté conectado directamente un nivel por debajo el nodo padre. Y en este ejemplo podemos ver que en la misma jerarquía esta *h1 y p* que son contenidos por su nodo padre *body*



Ahora que ocurre si la jerarquía es más elaborada como indica el siguiente gráfico. Digamos que como muestra el grafico anterior body contiene a h1 y p y en lugar de eso como vemos en el siguiente grafico en lugar de h1 tenemos un div y dentro de ese div tenemos un h1 y un a, en ese caso ya sería relevante el concepto de Nodo descendiente. Al igual que una familia tendrías a tu tátara abuelo tendría muchos descendientes en la jerarquía que no son hijos directos pero si son sus descendientes porque son sus bisnietos y nietos. Ese mismo concepto del árbol genealógico también aplica aquí al DOM. En resun: html es el nodo padre de toda la estructura en la cual body es hijo de html a su vez body es padre de div y p y div es padre de h1 y a



Además otro concepto importante que ya hemos mencionado pero aquí tenemos el concepto formal. Esos nodos que están al mismo nivel como podemos observar en el siguiente diagrama se denominan Nodos hermanos y comparten el mismo Nodo padre.



NODOS Vs. ELEMENTOS

El concepto de Nodo es más amplio que el concepto de Elementos en el DOM, es importante saber cuál es la diferencia entre ambos. Estos conceptos son fáciles de intercambiar y es fácil confundirse entre un Nodo y un Elemento cuando ya estamos trabajando en el contexto del DOM

Herramientas de Desarrollo:

Veamos cómo usar las herramientas de google chrom con la cual vamos a trabajar en el curso, en estas herramientas va