

Con Javascript podemos manipular lo que se ve en pantalla, no le llamamos en la jerga técnica "pantalla" a la porción de información que un usuario ve en su monitor y que tiene que ver con el sitio web. Esa función, esa porción, mejor dicho, es un objeto, es un objeto que se llama "documento" o "document", ¿si?. Dijimos antes que en Javascript casi todo era un objeto, salvo los tipos literales. Bueno, este es un objeto importantísimo, lo que el usuario puede ver. Se les llama "dom" "Document Object Model", ¿si? es el modelo del objeto documento, ¿por qué modelo? Bueno, precisamente porque hay un modelo, hay métodos, funciones para interactuar con él, para trabajar con él. Básicamente está compuesto por nodos, podríamos pensarlo como "nodos" o como ramas de un árbol, ¿si? Hay un objeto, el primero, que podría ser HTML, el propio documento o el "body", ¿recuerdan cuando veíamos, bueno, cuando seguimos viendo en HTML nuestras etiquetas y luego cada una de esas etiquetas, puede tener etiquetas dentro? Esos son los "child nodes" o "nodos hijos" y lo más interesante de esto, es que podemos recorrer el "dom" a través de Javascript, capturar cada uno de esos elementos, cambiarlos, transformarlos, eliminarlos, crear nuevos, toda esa magia podemos hacerla con Javascript. Recorriendo el "dom" aquí tenemos para comenzar nuestra prueba, un simple HTML con un texto, algunos párrafos, bueno, un botón un "input", ya veremos cómo interactuar con estos elementos, con estos nodos del "dom" directamente desde Javascript.

Comencemos por algo muy sencillo, tenemos aquí un párrafo que indica "este es un texto creado en HTML", bien, podríamos acceder a ese párrafo desde nuestro archivo, Javascript. Vamos a declarar una constante, vamos a ponerle el nombre "p", es igual, es indistinto, podríamos ponerle "para", por ejemplo, y esto va a ser igual a "document", porque este es el objeto "documento", recuerden que estamos hablando del "dom", del modelo del objeto documento.

Hay un objeto que se llama document que es nuestro objeto principalísimo en este caso bien entonces vamos a referirnos al objeto "document" y vamos a usar uno de sus muchísimos métodos y propiedades, vamos a usar uno de los uno de los métodos para capturar nodos, para capturar elementos del documento. Podríamos en este caso utilizar, vamos a utilizar un "query selector", bien, "query selector" va a encontrar el primer nodo de un tipo determinado. Aquí voy a decirle "busca en el documento un nodo del tipo p". Okey, tenemos dos, pero va a capturar sólo este, porque va a capturar el primero que encuentre y en este caso, eso nos sirve, por lo tanto vamos a dejarlo así. Bien para que ustedes sepan que esto está funcionando, podríamos chequearlo con un "console log", podríamos preguntar qué cosa es la constante "para". Aquí deberíamos abrir la consola, ¿verdad?, para que aquí debajo nos lo indique. Bien, para eso un párrafo, exactamente, es este primer párrafo, bien, entonces, ya lo hemos capturado. ¿Qué cosas podemos hacer? Bueno, podemos modificar todo de ese párrafo, pero comencemos por modificar su contenido, por ejemplo, con la propiedad "inner text" y vamos a indicar que el "inner text", es decir, el texto, el texto interior del párrafo, sea "Este texto ha sido modificado".

Y ahora, presten atención cuando grabé el archivo, esto tiene que cambiar de manera dinámica, perfecto, y ha cambiado de forma dinámica y de hecho, si aquí ven en el "console log" que estamos haciendo aquí, ya podemos ver que este párrafo, es un párrafo que además, tiene este contenido, ¿okey? Esto no nos interesa, por ahora vamos a quitarlo, muy bien, ya sabemos cómo acceder o por lo menos conocemos una manera para acceder a este

elemento. Bueno, como dato de color podríamos indicar que “text content” hace exactamente lo mismo que “inner text”, ¿okey? “Text content”.

Vamos a escribirlo bien, es decir “text content”, “inner texto” hace en la práctica lo mismo, cambia esta porción del elemento. Si hablamos de un párrafo cambia este valor, es decir, el texto de un párrafo, bien pero no solamente podemos acceder al contenido, también podemos acceder a otras, a otras propiedades, a otros valores, por ejemplo, podríamos cambiar el estilo de nuestro párrafo. Supongan que tenemos una clase, una clase en CSS, vamos a crear rápidamente aquí un archivo “style”, punto “C”, “S”, “S” y vamos a indicar que mi párrafo o, mejor aún, vamos a hacer una “clase”, punto, “para”, vamos a ponerle así. Cualquier elemento al que le aplique la clase “para” tendrá un color y aquí pongamos, bueno, si, “red”, no importa que sea un color muy, muy chillón, es solamente para fines educativos, así que, bueno, cualquier color está bien. Y vamos a cambiarle también el “font size”, como para que se note bien nuestro cambio, ¿verdad? Vamos a darle un “extra large”, okey, bueno, ahora ningún, ningún elemento tiene aplicada esta clase, vamos a indicar además aquí también, esto es muy importante, por supuesto, que queremos cargar a nuestro archivo CSS, bueno, dejemos este nombre porque es el mismo que hemos usado nosotros. Vamos a probar aquí, primero en caliente, a ver si esto funciona, pongámosle a este párrafo una clase y la clase se llama “para”, lo aplicamos y efectivamente la clase está funcionando, pero vamos a quitarla, porque queremos hacerlo de manera dinámica, por tanto, podríamos venir aquí y no solamente cambiar el texto, sino también aplicarle a “para” una clase, podríamos indicar “ad”, perdón, “class list” punto, “ad” y aquí vamos a indicar “para”, okey y agregamos la clase. Luego también podríamos quitarla, ¿cómo se quita una clase?, bueno, hagamos hagamos esto, vamos a ponerle a este último párrafo, a este segundo, vamos a aplicarle la clase “para” pero la aplicamos desde HTML, ah, bien, pero este párrafo no se está mostrando, bueno, hagamos una prueba, aquí simplemente pongamos un texto cualquiera, ya lo borraremos, entonces podemos acceder a este segundo párrafo y quitarle la clase, bien. Ahora el problema es que con ese selector, va a seleccionar el primer elemento del tipo indicado. Cuando tenemos más elementos iguales, si queremos capturarlos a todos, podríamos indicar “query selector all” y esto va a seleccionar a todos los párrafos, pero atención, noten que ahora ya no está reaccionando aquí y esto es porque “queries selector all”, no nos devuelve un solo elemento, en realidad, aquí tendríamos que haber puesto “paras”, para ser más exactos, no importa el nombre, ¿okey? Simplemente es para que ustedes tengan una idea de lo que aquí está pasando, lo voy a poner en plural, “paras” como “paragraph” o párrafos, ¿por qué?, porque si ustedes notan, e queries selector all”, todos los elementos del tipo “p” en nuestro documento.

Tenemos dos, pero podrían ser muchos más, por lo tanto esto no nos devuelve un solo elemento, sino una suerte de “array”, una lista de ellos. Miren este “console log”, aquí abrimos la consola y ustedes van a ver algo parecido a un arreglo, bien, esta línea no va a funcionar, porque ya este elemento no existe, así que la comentaré y aquí en otro lado también voy a comentar esto, pero noten que aquí me dice, “node list”, lista de nodos y tengo dos párrafos, el primero y el segundo. El primero, fíjense cómo en el “dom”, en la parte de arriba a la izquierda, se marca cada uno de los párrafos, okey, entonces ¿cómo podría seleccionarlos? Bueno, es una lista de nodos, un “out list”, aunque técnicamente funciona igual que un “array”, por lo tanto, tenemos un índice. La posición cero, será el párrafo de arriba y la posición uno, será el párrafo de abajo. Fácil, ¿verdad? Okey, entonces, el texto al que le queremos cambiar el, perdón, el párrafo al que queremos cambiar el texto es este, es el primero, por lo tanto, hablaremos de “paras” en la posición cero, ¡bum! y este texto ha sido

modificado, ya está funcionando y al que queremos ahora agregarle la clase es a este mismo "paras", en la posición cero y le agregamos la clase "para", pero aquí abajo, al párrafo, el segundo párrafo, le quiero quitar la clase. Bueno, eso es tan sencillo como indicar que en la posición uno de la lista de nodos o arreglo, en realidad, le podemos llamar así, que contiene todos los párrafos del documento, podríamos indicar "class list remove" y el nombre de la clase que queremos quitar y hemos quitado la clase de manera dinámica. Okey, muy bien ya sabemos cómo acceder a los elementos y cómo ponerle quitarle clases, podemos, en realidad, manipularlos totalmente a los elementos, podemos hacer lo que queramos con ellos, pero también podemos crear elementos.

Supongan que queremos crear debajo de este primer párrafo bueno, bueno, aquí al final no importa dónde, ahora no importa dónde, en este caso no importa vamos a hacer entonces un aquí "const", nuevo "para", nuevo párrafo, igual, "document" y ahora no es algún método para obtener un elemento, sino para crearlo, por el momento será un "create element". Vamos a crear un elemento, ¿qué elemento?, el elemento de tipo, bueno, otra vez, párrafo. Podríamos cambiar, podríamos crear una "h3" también, no hay ningún problema, bien, ya está creado, no está agregado al "dom", pero está creado. De hecho fíjense ustedes que el nuevo "para", aquí abajo, es un "h3" que no tiene, por ahora, nada, pero existe. Vamos a indicarle a nuestro nuevo "para", es que en realidad, aquí estoy cometiendo como un error, no porque sería un nuevo "title", reitero los nombres no importan en este caso, tanto, es decir importan porque hay que hacer el código legible y entendible, pero es lo mismo el nombre que yo indique, lo que importa es la lógica que estamos aplicando. Entonces nuevo "title", vamos a ponerle "inner text", será igual a, bueno, esto es un título dinámico, ¿por qué dinámico?, porque lo estamos agregando "on the fly", o sea, al vuelo desde Javascript. Título dinámico con Javascript, perfecto, y ahora fíjense que el "console log" bueno, aquí en algún sitio estoy haciendo referencia a nuevo "para", aún dónde, dónde déjenme ver, aquí nuevo "para" y es nuevo "title". Ya es un elemento completo, es decir, tiene las etiquetas de apertura y cierre, pero también un texto, bien, próximo paso, agregarlo al documento.

¿Cómo podemos agregarlo al documento? Bueno, de varias maneras, una forma sería agregarlo directamente al "body", por ejemplo, o que tendríamos que seleccionar allí, donde nosotros queremos agregarlo. Vamos a ir a nuestro "index" para hacerlo de manera prolija, podríamos hacerlo, bueno, aquí al final, por ejemplo, aquí, podríamos agregar un ejemplo, un "div", o podríamos capturar nuestro "body" directamente. Bien, en este caso, como es un ejemplo no hay problema lo podemos hacer así vamos a seleccionar nuestro "body" otra vez el nombre es simplemente convencional "document", punto, "query selector", luego verán que hay más formas de seleccionar, ¿verdad?, por ahora con estas estamos bien y ahora aquí, al final, podríamos indicar "body", punto, "appendChild", porque vamos a agregar un nodo hijo a la etiqueta "body" y, ¿cuál es la etiqueta que quiero? El nodo. El nodo, el elemento hijo que quiero agregar, el nodo hijo o el "child node", bueno, es este nuevo "title" que es un "h3" que tiene un contenido nuevo, "title", ¡boom! y aquí lo tenemos. Si quisiéramos agregarlo en otra sección, bueno, tendríamos que, por ejemplo, manipular el "dom" aquí, creando algún "div", bien, eso ustedes pueden verlo un poco más adelante, pero así es como, básicamente, podemos agregar un elemento, por lo demás, también podríamos eliminar elementos. Miren, vamos a hacer una cosa, vamos a hacer un bucle, vamos a algo un poco más divertido, okey. Más divertido aún.

Aquí, al final vamos a poner una, una lista, una lista ordenada, ¿okey?, pero no tiene ningún "list item", okey, o sea, que no vamos a ver nada en pantalla por ahora. De manera dinámica vamos a agregarlos, por lo tanto deberíamos capturar ese elemento para tenerlo como

referencia, ¿okey? “Document”, punto, “Query selector” y vamos a seleccionar es una lista ordenada, si mal no recuerdo, okey, la lista ordenada es la única que tenemos en el documento, así que no habrá ningún problema.

Y aquí vamos a hacer un bucle, “for, let i, equal, zero” y mientras sí sea menor que cinco, vamos a agregarle uno, a nuestro contador y ,¿qué vamos a hacer aquí, adentro del bucle “for”? Vamos a crear un “list item” por cada pasada, entonces, podríamos hacer “const, list, item, equal, document, punto create element”. ¿Qué elemento vamos a crear? Un elemento de tipo LI y luego, a este elemento le vamos a agregar un texto punto “inner text” sería igual a, elemento, perdón, aquí deberíamos hacer un texto, lo voy a hacer con “back ticks” para poder concatenar la variable “i” y poner elemento número y que quede un cero, uno dos, uno, dos tres, etcétera, elemento y aquí vamos a poner la variable, el contador, que la primera vez es cero, la segunda uno, la lista nos dirá elementos cero, uno, dos, etcétera. Si no quieren que arranque en cero, bueno, simplemente aquí hacemos y más uno, okey, perfecto, creo que estamos. Ahora, cada vez se ejecuta una de estas líneas, es decir, una se hace, una de las iteraciones del bucle “for”, se crea una variable, esta variable es asignada a un, se le asigna un elemento de tipo “LI”, se le agrega un texto a ese elemento de “LI” y finalmente ese elemento lo vamos a agregar a la lista que hemos capturado aquí arriba. Entonces, aquí, finalmente deberíamos hacer “OL” punto “appendChild”, ¡jupa!, ¿qué pasó?, no le ha gustado, “OL” no existe, es que le llame “lista”, nunca iba a funcionar, “appendChild”, “LI”, okey y aquí tenemos todos nuestros elementos. La verdad es que esto de ponerle número y que además sea una lista ordenada, no me gusta demasiado, vamos a tomarla como lista desordenada, así que aquí vamos a hacer “UL”, “UL”, okey, bien. Elemento 1, 2, 3, 4, 5, etcétera y ¿qué pasa si queremos borrar un elemento? o mejor aún, ¿qué pasa si queremos borrar todos los elementos?, bueno pues haríamos un “for” similar, por ejemplo haríamos un “for”, he copiado la línea que no correspondía aquí, “for”, bien y aquí tendríamos que en cada una de las pasadas, quitar uno de los elementos “LI” y por lo tanto, podríamos, aquí podríamos seleccionar antes del bucle, no se me ocurre, “const” o podríamos hacerlo aquí adentro, con esta “LI” igual “document”, punto, aquí no voy a usar un “query selector all”, voy a elegir sólo uno, es decir que capture el primero que encuentre “query selector LI” y aquí y ahora haríamos una “Li remove”. Okey, y desaparecen todos, bien y ahora vamos a limpiar todo esto, porque nos interesa, bueno, esto también aquí está sobrando, porque lo vamos a utilizar dentro de un instante nada más y vamos a quitarle también la clase, vamos a capturar un elemento cuando presionemos un botón y allí veremos también, entonces, el concepto de eventos. Bien tenemos aquí una caja de texto, un “input” y aquí un botón que, por supuesto, por ahora no hace nada. Vamos a comenzar capturando estos elementos, ¿cómo se llama nuestro “input”? veamos, bueno, no tiene ningún ID y por lo tanto, puedo capturarlo simplemente como “input”, ya que es el único que hay. Bueno, sí tiene una y dividen texto, ok, claro, tiene una ID, porque como le pusimos un “label” y recuerden que cada “label” tiene que estar referenciado con su “input” a través del atributo “for” en el “label” y el ID, en el “input”, por lo tanto, aquí podría mostrarles, incluso, aunque esto no ha sido planificado, pero nos viene bien, otra manera de capturar elementos del “dom”, ¿no? desde el “dom”. Hasta ahora lo hicimos con “query selector” y “query selector all”, pero también podemos capturar elementos según su ID y “get element by Id” y por lo tanto aquí, como tengo un elemento que posee una ID, es el “input” y tiene el ID texto lo voy a capturar de esa manera, cuidado es lo mismo que hacerlo “query selector input” no, no, no varía en absolutamente nada, el resultado es el mismo pero hay más de un modo de seleccionar, de escoger elementos del “dom”, simplemente eso quiero mostrarles. Ok, ya tenemos entonces el nuestro input, ahora

vamos a capturar el botón. Vamos a ponerle de nombre "btn" y le haremos un "document", punto, veamos si tiene una ID, usaremos el ID, no tiene, por lo tanto simplemente voy a capturarlo con "query selector". "query selector", "query selector" es un "button", bien, ahora lo que quiero lograr es que el texto que se escriba acá, pueda mostrarse aquí debajo, en un párrafo, que por ahora no tiene ningún contenido, pero que está ahí abajo y que tiene este ID y muestra texto y quiero, dinámicamente, mostrar ese texto aquí, pero sólo cuando se presiona el botón. Vamos a ver cómo sería esta lógica. Tendríamos que detectar el click del botón para poder disparar una función, eso podemos hacerlo con un "event listener", es decir, un método que posee cada elemento y que ayuda a detectar cuando un evento de algún tipo, hay unos 200 tipos de eventos en el "dom", bueno, cuando uno de los, de los eventos es disparado, es activado, a nuestro botón vamos a agregarle "ad event listener". "Event listener" recibe aquí un primer parámetro que es el nombre del evento, vamos a escuchar el evento click, es decir, cuando un usuario haga click en el botón click, así se llama simplemente, y, ¿qué va a ocurrir cuando el usuario haga click? Podríamos directamente indicar aquí un algoritmo o una, digamos, algún paso, alguna ejecución de código o directamente, esto es más prolijo aún, llamar a una función "show text", por ejemplo, okey, y ahora definamos qué hace esa función "show text". Entonces "function show text", y esa función va a capturar el texto del input y lo va a mostrar en el párrafo de abajo. Bien, capturamos este párrafo, vemos que tiene un ID, muestra texto, así que lo podríamos capturar por ID, es una buena práctica y capturando antes de su uso todos los elementos que vamos a necesitar, vamos a ponerle "para", va a ser igual a "document", ¡ups!, punto, "get element by id" y el ID era creo, "muestra texto", ¿verdad? "muestra texto". Perfecto. Entonces nuestra función "show text" que, reiteramos, se va a disparar, va a ser llamada cuando hagamos click en el botón, el botón es este botón, este que está aquí, aquí lo hemos capturado, aquí arriba, bien, entonces, ¿qué tiene que hacer?, bueno, tomar el elemento "para" y agregarle un texto, porque ahora no tiene nada, tener texto y el texto va a ser igual a, ¿qué cosa?, ¡ajá!, acá viene lo importante, el texto está en el input, pero, si nosotros aquí ponemos simplemente "input", vamos a obtener todo el elemento y nosotros solamente queremos su valor, esa es la propiedad que queremos obtener, por lo tanto, vamos a indicar "punto, value", ¿okey?, y ahora vamos a correr esto, escribamos aquí un texto, démosle click a nuestro botón y aquí tenemos nuestro texto. Si escribimos alguna otra cosa aquí, tenemos lo que hemos capturado, ¿okey?. Así es como podemos detectar un evento, lanzar una función y con esa función manipular el "dom".