

Clase 5: JavaScript (JS)

Introducción a Javascript

JavaScript es el lenguaje de programación más popular del mundo, es un lenguaje que permite a los desarrolladores crear páginas web interactivas y dinámicas. Existe desde hace más de veinte años, pero sigue ganando popularidad a medida que más gente se da cuenta de sus casos de uso.

JavaScript es esencialmente lo que hace que todos los sitios web sean interactivos. JavaScript puede utilizarse para hacer varias cosas, algunas de ellas son crear animaciones y juegos en un sitio web, reproducir efectos de sonido cuando alguien interactúa con una página. (Platzi).

Javascript, como también lo es HTML y CSS va teniendo cambios en sus especificaciones. Brevemente su historial es el siguiente:

- El JavaScript original ES1 ES2 ES3 (1997-1999)
- La primera revisión principal ES5 (2009)
- La segunda revisión ES6 (2015)
- Todas las adiciones anuales (2016, 2017, 2018, 2019, 2020)

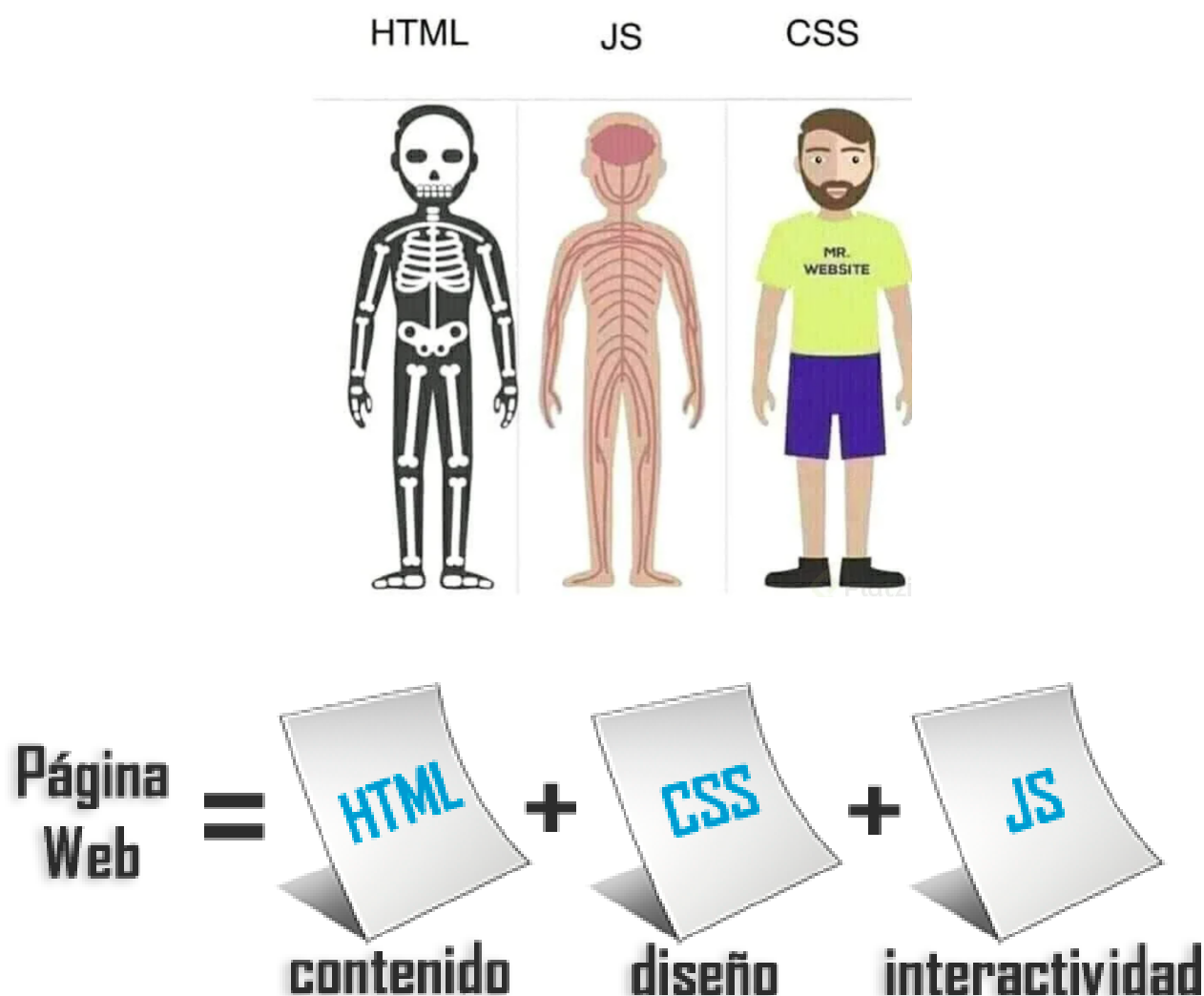
Hemos visto HTML, CSS y ahora vemos la tercera tecnología / lenguaje de las páginas web. Cada una de ellas tiene su propósito.

HTML define la estructura de las páginas web

CSS define su diseño y

JS especifica el comportamiento, las funcionalidades.





Haremos un recorrido práctico utilizando nuestro sitio librería, así mismo, en estos [videos de Javascript de TODO CODE](#) y en [w3schools.com/js](https://www.w3schools.com/js) hay un detalle que invitamos a recorrer. En el sitio de w3schools hay muchos ejemplos, como siempre recomendamos, es muy útil ver y probar con la opción que tiene del “Probalo vos mismo” donde nos permite ir haciendo pequeños cambios e ir viendo el resultado.

Volvemos a compartirles también otro video, en realidad un grupo de videos en Youtube de Kiko Palomares, es una lista de videos de introducción al desarrollo web, los videos de JavaScript van desde este, el 21 https://youtu.be/Ifvdh9wK_rl?list=PLnunbwZjHqMMQv-Gfj-NOAYLarTbMcwOe al 29. Los videos anteriores son sobre HTML y CSS, con lo cual, también es una lista interesante para los contenidos que hemos y estamos viendo.

¿Cómo obtengo JavaScript? ¿Dónde puedo descargar JavaScript? ¿JavaScript es gratuito?

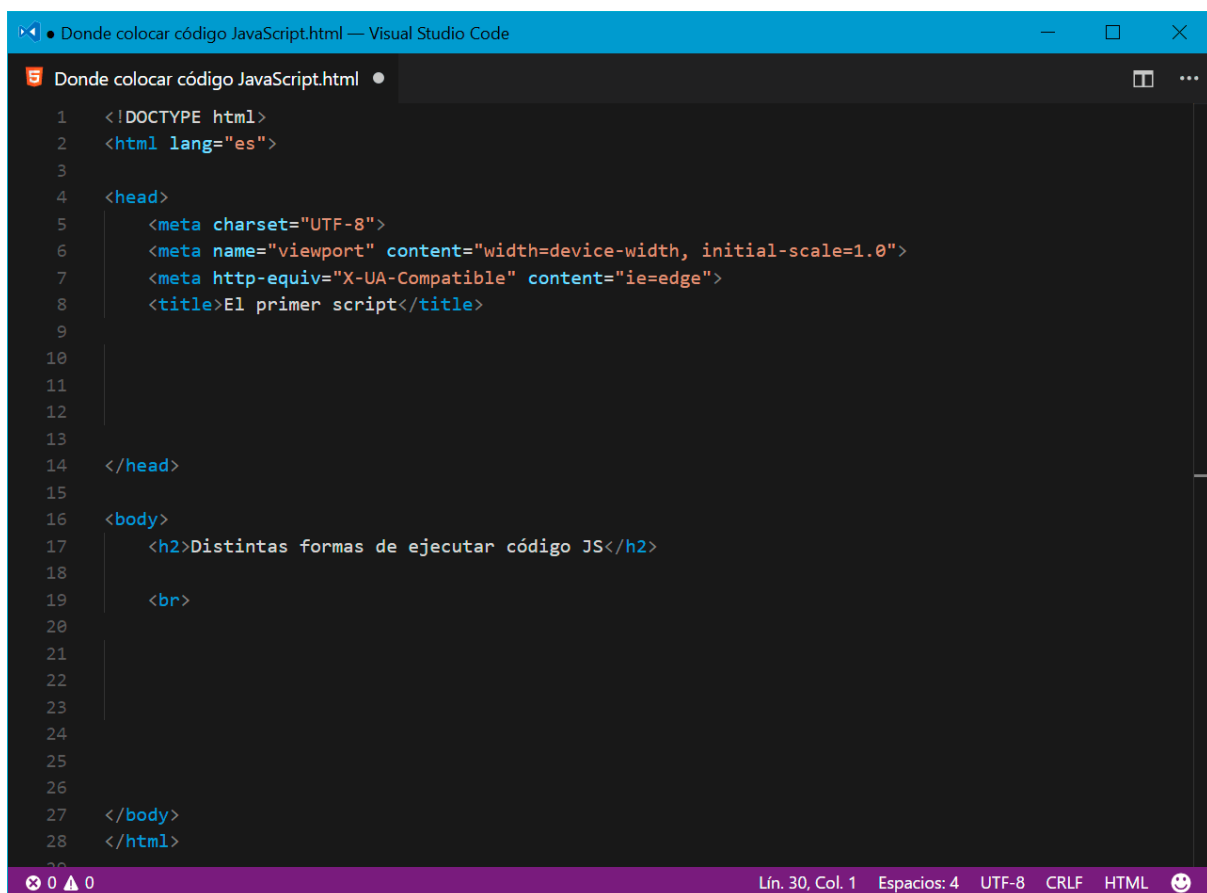
No es necesario que obtenga o descargue JavaScript. JavaScript ya se está ejecutando en su navegador en su computadora, en su tableta y en su teléfono inteligente. JavaScript es de uso gratuito para todos.

Por lo tanto, no necesitamos instalar nada, solo seguir utilizando lo que venimos usando, el editor de texto VS Code y un navegador web.

En el proyecto de librería, y algunos videos que dejaremos, iremos haciendo un breve recorrido por javascript. ¿Qué vamos a ver?

1 - ¿Cómo incorporar javascript en mi proyecto?

En el html entre etiquetas `<script>` o en archivo externo con extensión js



```
1 <!DOCTYPE html>
2 <html lang="es">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>El primer script</title>
9
10
11
12
13
14 </head>
15
16 <body>
17   <h2>Distintas formas de ejecutar código JS</h2>
18
19   <br>
20
21
22
23
24
25
26
27 </body>
28 </html>
```

2 - Analizamos las declaraciones y su sintaxis (comentarios, indentación)

3 - Opciones para mostrar lo que hacemos en javascript (console.log(), alert(), innerHTML). Herramientas del navegador (inspeccionar, consola)

4 - Veremos variables, constantes, como definir funciones, como usar estas funciones que definimos. Utilizar recursos como condicionales y bucles.

Las **variables** son contenedores para almacenar datos (almacenar valores de datos).

Una **función** de JavaScript es un bloque de código diseñado para realizar una tarea en particular.

Las declaraciones **condicionales** se utilizan para realizar diferentes acciones basadas en diferentes condiciones.

En JavaScript tenemos las siguientes declaraciones condicionales:

- Úselo **if** para especificar un bloque de código que se ejecutará, si una condición especificada es verdadera
- Se usa **else** para especificar un bloque de código a ejecutar, si la misma condición es falsa
- Úselo **else if** para especificar una nueva condición para probar, si la primera condición es falsa
- Úselo **switch** para especificar muchos bloques alternativos de código para ejecutar

JavaScript admite diferentes tipos de **bucles** (ciclos o repeticiones de ciertas instrucciones):

- **for**- recorre un bloque de código varias veces
- **for/in**- recorre las propiedades de un objeto
- **for/of**- recorre los valores de un objeto iterable
- **while**- recorre un bloque de código mientras una condición específica es verdadera
- **do/while**- también recorre un bloque de código mientras una condición específica es verdadera

Estos primeros 4 puntos se abordan en el proyecto con los archivos: **inicioJavascript.html** y **iniciando.js**

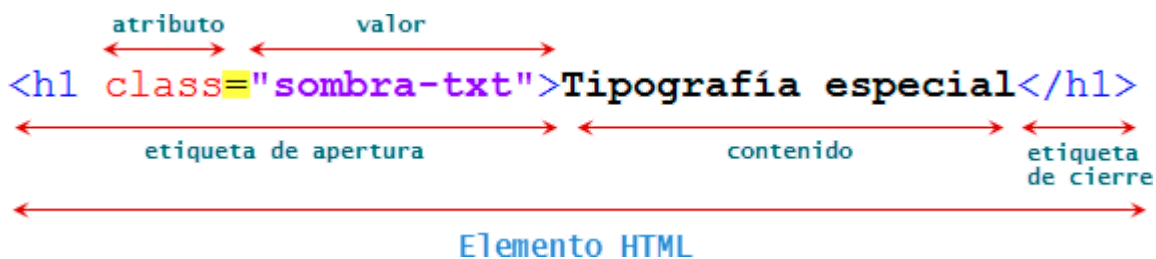
5 - Veremos algunos ejemplos con distintos tipos de datos (arreglos).

Archivo en el proyecto: **arreglos.js**

6 - Cómo interactuar con HTML, viendo las distintas formas de seleccionar los elementos HTML y hacer intervenciones sobre ellos. utilizando el DOM (Document Object Model), *GetId*, *getName*, *getClassName*, *querySelector* y *obtener los campos del formulario*. Usar *display*.

Este punto se abordan en el proyecto con los archivos: **inicioJavascript.html** y **obtenerElemento.js**

Para seleccionar elementos HTML desde javascript es necesario comprender bien la sintaxis HTML y CSS, ya que son utilizadas en este proceso de obtener el o los elementos HTML que cumplan alguna condición y luego hacer algo con ellos. Es por eso que vamos a repasar la sintaxis de HTML y de CSS:



<etiqueta atributo="valor">...</etiqueta>

Sobre atributos HTML: [más información aquí](#)

CSS

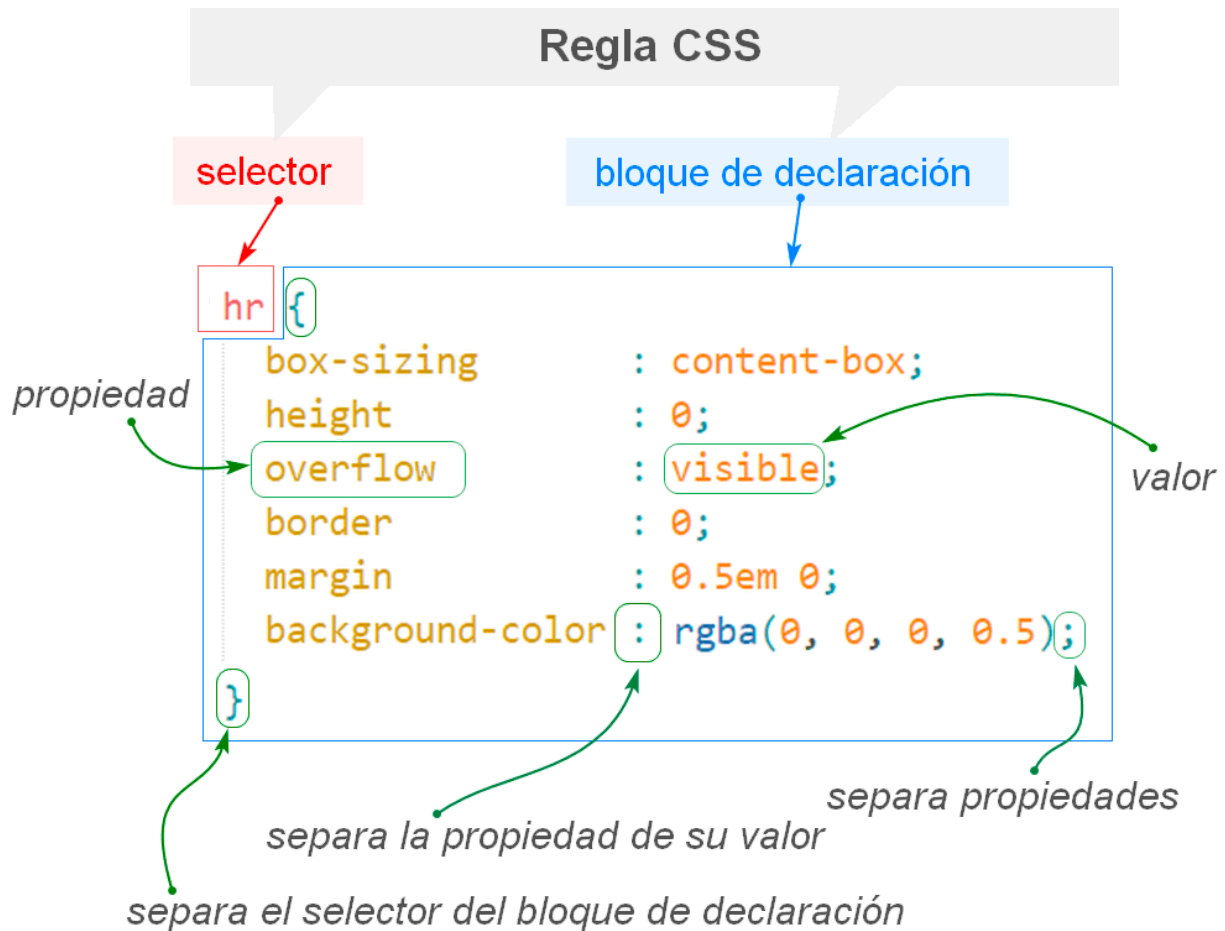
Una regla CSS se compone de un selector y de un bloque de declaración

Cada declaración está compuesta por uno o más pares propiedad/valor

El bloque de declaración empieza y acaba con los signos de llaves { ... }

Cada par propiedad/valor se separa por el signo punto y coma ;

Cada propiedad debe separarse de su valor por el signo dos puntos :



- **Regla:** cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de "selectores", un símbolo de "llave de apertura" {, otra parte denominada "bloque de declaración" y por último, un símbolo de "llave de cierre" }.
- **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
- **Bloque de declaración:** especifica los estilos que se aplican a los elementos. Está compuesto por pares propiedad CSS y su correspondiente valor.
- **Propiedad:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, color, etc. y cada propiedad tiene un conjunto de posibles valores.
- **Valor:** establece el nuevo valor de la propiedad modificada en el elemento

Se pueden definir varios selectores simultáneamente escribiendo los selectores separados por comas ,. Lo que no es correcto es definir varias propiedades simultáneamente.

Una hoja de estilo puede contener comentarios. Los delimitadores del comentario son / ... */ (como en el lenguaje de programación C). Los comentarios pueden extenderse a varias líneas.*

Un archivo CSS puede contener un número ilimitado de reglas CSS, cada regla se puede aplicar a varios selectores diferentes y puede contener un bloque de declaración con uno o más pares propiedad/valor.

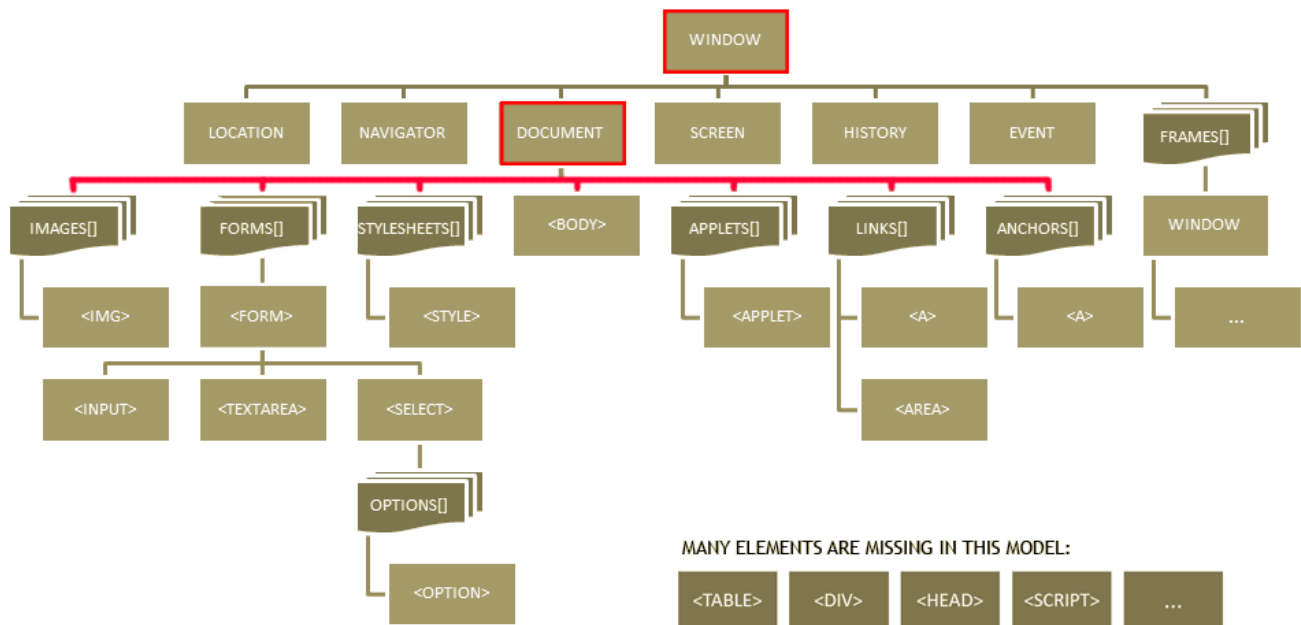
DOM

La creación del Document Object Model o DOM es una de las innovaciones que más ha influido en el desarrollo de las páginas web dinámicas y de las aplicaciones web más complejas. DOM permite a los programadores web acceder y manipular las páginas HTML.

Árbol de nodos

Una de las tareas habituales en la programación de aplicaciones web con JavaScript consiste en la manipulación de las páginas web. De esta forma, es habitual obtener el valor almacenado por algunos elementos (por ejemplo, los elementos de un formulario), crear o eliminar un elemento (p, div, img, etc.) de forma dinámica, manipular (poner/quitar/cambiar) una clase de un elemento, etc. Todas estas tareas habituales son muy sencillas de realizar gracias a DOM, pero para ello es necesario "cambiar" la página original.

Los navegadores web transforman automáticamente todas las páginas web en una estructura más eficiente de manipular, el DOM. Esta transformación la realizan todos los navegadores de forma automática y nos permite utilizar las herramientas de DOM de forma muy sencilla. DOM transforma todos los documentos HTML en un conjunto de elementos llamados nodos, que están interconectados y que representan los contenidos de las páginas web y las relaciones entre ellos. Por su aspecto, la unión de todos los nodos se llama "árbol de nodos".



La raíz del árbol de nodos de cualquier página HTML siempre es la misma: un nodo de tipo especial denominado **document**. A partir de ese nodo raíz, cada etiqueta HTML se transforma en un nodo de tipo "**Elemento**". Como se puede suponer, las páginas HTML habituales producen árboles con miles de nodos. Aun así, el proceso de transformación es rápido y automático.

Tipos de nodo

La especificación completa de DOM define 12 tipos de nodos, aunque en las páginas HTML se manipulan habitualmente cuatro tipos de nodos:

Document: nodo raíz del que derivan todos los demás nodos del árbol.

Element: representa cada una de las etiquetas HTML. Se trata del único nodo que puede contener atributos y el único del que pueden derivar otros nodos.

Attr: se define un nodo de este tipo para representar cada uno de los atributos de las etiquetas HTML, es decir, uno por cada par atributo=valor.

Text: nodo que contiene el texto encerrado por una etiqueta HTML.

Una vez que se ha accedido a un nodo, el siguiente paso consiste en acceder y/o modificar sus atributos y propiedades. Mediante DOM, es posible acceder de forma sencilla a todos los atributos HTML y todas las propiedades CSS de cualquier elemento de la página.

Los atributos HTML de los elementos de la página se transforman automáticamente en propiedades de los nodos. Para acceder a su valor, simplemente se indica el nombre del atributo XHTML detrás del nombre del nodo.

```
<a href="https://es.wikipedia.org">Wikipedia</a>
```

```
var enlace = document.getElementsByTagName('a');
```

```
alert(enlace[0].href); // mostrando la URL https://es.w...
```

En el ejemplo anterior, se obtiene el nodo DOM que representa el enlace mediante el método `document.getElementsByTagName()` que devuelve un array con todos los elementos “a” de la página. A continuación, se obtiene el atributo `href` del enlace mediante `enlace[0].href`. Para obtener por ejemplo el atributo `target`, se utilizaría `enlace[0].target`.

Las propiedades CSS no son tan fáciles de obtener como los atributos XHTML. Para obtener el valor de cualquier propiedad CSS del nodo, se debe utilizar el atributo `style`. Si el nombre de una propiedad CSS es compuesto, se accede a su valor modificando ligeramente su nombre. La transformación del nombre de las propiedades CSS compuestas consiste en eliminar todos los guiones medios (-) y escribir en mayúscula la letra siguiente a cada guión medio.

El único atributo HTML que no tiene el mismo nombre en HTML y en las propiedades DOM es el atributo `class`. Como la palabra `class` está reservada por JavaScript, no es posible utilizarla para acceder al atributo `class` del elemento HTML. En su lugar, DOM utiliza el nombre `className` para acceder al atributo `class` de HTML.

Otro sitio recomendado es 3con14.biz que tiene un buen resumen de cada uno de estos tres temas: [HTML](#), [CSS](#), [JAVASCRIPT](#)

7 - Por último, utilizaremos eventos.

Los eventos HTML son "cosas" que les suceden a los elementos HTML.

Cuando se usa JavaScript en páginas HTML, JavaScript puede "reaccionar" a estos eventos.

Este punto se aborda en el proyecto con los archivos: **inicioJavascript.html** y **obtenerElemento.js** y también en **listadoLibros.html** y **main.js** mediante el **onClick()**, y en **clientes.html** y **main.js** mediante **onBlur()** y **onSubmit()**.

Habiendo visto todo lo anterior, veamos el uso de javascript en el proyecto.

“Agregamos dos columnas en el listado de libros (tabla) y hacemos un cálculo del **carrito de compras**”: aca tenemos evento cada vez que hace clic en la cantidad, recalculamos el carrito, por lo que hay que seleccionar distintos elementos del carrito, obtener datos de algunos, actualizar en otros. Usaremos funciones, variables, ...


Este punto se aborda en el proyecto con los archivos: **listadoLibros.html** y **main.js** mediante **onClick()**.

*En la página del formulario de un cliente vamos a “**Controlar que el mail sea el mismo**”. Acá obtenemos ambos elementos y hacemos una comparación, si son incorrectos avisamos y posicionamos el cursor en el segundo campo de mail, sino continua sin advertencias.*

*Este punto se aborda en el proyecto con los archivos **clientes.html** y **main.js** mediante **onBlur()** y **onSubmit()**.*

Hay mucho más por aprender de JavaScript, hay mucha información en internet. Nosotros hemos intentado dar los primeros pasos en cada una de estas tres tecnologías para inciar en el desarrollo de sitios web.

Les dejamos un video donde se diseña en javascript el juego de piedra papel y tijera en 10 minutos. Se utilizan la mayoría de las funcionalidades que hemos visto.

- Juego: Piedra, papel y tijera.  [Cómo crear un videojuego con JavaScript en 10 minutos](#)

En la sección Actividades invitamos a realizar la consigna, la cual invita a realizar una función en javascript, en un archivo javascript y desde una de las páginas html de tu sitio web poder hacer uso, invocar a dicha función desde un evento, por ejemplo, haciendo click en un botón.

Mucha suerte, hay mucho por hacer :)