



# DESARROLLO WEB

## PRIMEROS PASOS



**Leonardo Cardoza - Nelson Cerros - José Palacios**

## HTML.

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

Se trata de un formato abierto que surgió a partir de las etiquetas SGML (Standard Generalized Markup Language). Concepto traducido generalmente como «Estándar de Lenguaje de Marcado Generalizado» y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto.

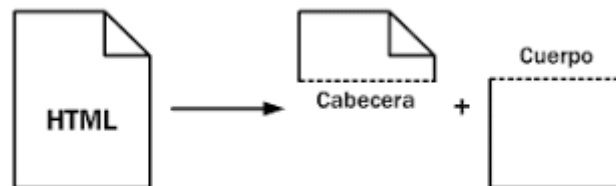
EL HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, etc).

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos.

Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos).

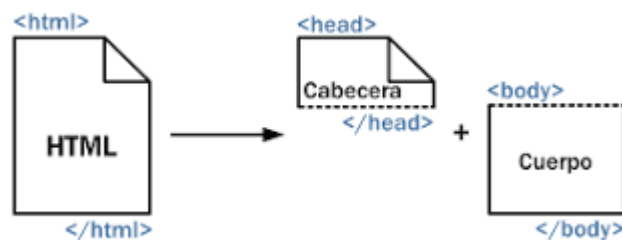
## Partes de un documento HTML

Un documento HTML ha de estar delimitado por la etiqueta <html> y </html>. Dentro de este documento, podemos distinguir dos partes principales: La cabecera y el cuerpo.



El encabezado, delimitado por <head> y </head> donde colocaremos etiquetas de índole informativo como por ejemplo el título de nuestra página.

El cuerpo, por las etiquetas <body> y </body>, que será donde colocaremos nuestro texto e imágenes delimitados a su vez por otras etiquetas.



`<html>`: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta `<html>`. En el interior de la etiqueta `<html>` se definen la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta `<html>` se ignora.

`<head>`: delimita la parte de la cabecera del documento. La cabecera contiene información sobre el propio documento HTML, como por ejemplo su título y el idioma de la página. Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta `<title>`, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la parte superior izquierda de la ventana o pestaña del navegador.

`<body>`: delimita el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, tablas). En general, el `<body>` de un documento contiene cientos de etiquetas HTML, mientras que el `<head>` no contiene más que unas pocas.

## Etiquetas del Head

### Link:

El elemento HTML `<link>` especifica la relación entre el documento actual y un recurso externo. Los usos posibles de este elemento incluyen la definición de un marco relacional para navegación. Este elemento es más frecuentemente usado para enlazar hojas de estilos.

```
<head>
  <title>Document</title>
  <link rel="stylesheet" href="">
</head>
```

### TITLE:

Un elemento del HEAD visible desde el navegador, muy importante para los buscadores pues es el texto que se visualiza en los resultados de búsqueda.

```
<head>
  <title>Document</title>
</head>
```

### Lang:

Se utiliza el atributo `lang` para establecer el idioma del documento.

```
<html lang="en">
```

### Favicon:

Un favicon es la pequeña imagen que se muestra en la pestaña del navegador o en la lista de marcadores (favoritos).

```
<head>
```

```
<title>Document</title>
<link rel="stylesheet" href="">
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
</head>
```

### Meta:

Las metaetiquetas, etiquetas meta o elementos meta (también conocidas por su nombre en inglés, metatags o meta tags) son etiquetas HTML que se incorporan en el encabezado de una página web y que resultan invisibles para un visitante normal, pero de gran utilidad para navegadores u otros programas que puedan valerse de esta información.

Su propósito es el de incluir información (metadatos) de referencia sobre la página: autor, título, fecha, palabras clave, descripción, etc.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <link rel="stylesheet" href="">
  <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
</head>
```

## Texto

En un documento HTML existen dos tipos de etiquetas HTML: las etiquetas que contienen fragmentos de texto (se utilizan para dar significado a esas palabras o fragmentos) y las etiquetas que agrupan conjunto de información (fragmentos de texto y/u otras etiquetas).

La etiqueta <p> y </p> se utiliza para párrafos.

La etiqueta <strong> se utiliza para fragmentos de texto importante o palabras clave.

La etiqueta <span> se utiliza para fragmento de texto sin significado (útil para seleccionar).

La etiqueta <cite> se utiliza para fragmento de texto con el título de un trabajo creativo: obras, libros...

La etiqueta <mark> se utiliza para fragmento de texto resaltado, simulando estar marcado con rotulador amarillo.

La etiqueta <small> se utiliza para anotaciones menores, pequeñas puntualizaciones.

## Imágenes

Para poner una imagen simple en una página web, utilizamos el elemento `<img>`. Este es un elemento vacío (lo que significa que no contiene texto o etiqueta de cierre) que requiere de por lo menos un atributo para ser utilizado — `src` (a veces llamado completamente como, `source`). El atributo `src` contiene una ruta que apunta a la imagen que quieres poner en la página, que puede ser una URL relativa o absoluta.

```
<img src="" alt="">
```

## Tablas

En documentos HTML una tabla puede ser considerada, resumidamente, como un grupo de filas donde cada una contiene a un grupo de celdas. Esto es conceptualmente distinto a un grupo de columnas que contiene a un grupo de filas, y esta diferencia tendrá un impacto en la composición y comportamiento de la tabla.

Como muchas otras estructuras de HTML, las tablas son construidas utilizando elementos. En particular, una tabla básica puede ser declarada usando tres elementos, a saber, `table` (el contenedor principal), `tr` (representando a las filas contenedoras de las celdas) y `td` (representando a las celdas).

```
<table>
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
    <td>Celda 3</td>
  </tr>
  <tr>
    <td>Celda 4</td>
    <td>Celda 5</td>
    <td>Celda 6</td>
  </tr>
</table>
```

## Contenedores(Div)

`div` de "division" –división. Sirve para crear secciones o agrupar contenidos.

```
<div></div>
```

## Form

`Form` es una etiqueta de HTML que representa un formulario. En este formulario se agregan los diferentes campos de entrada de datos o de confirmación, así como los botones, que como mínimo ha de haber uno, el de envío. También pueden agregarse más botones como el de Restablecer que pone el formulario en blanco.

```
<form action=""></form>
```

## Controles (Buttons, Input)

-La etiqueta de HTML `<button>` representa un elemento *cliqueable* de tipo botón que puede ser utilizado en formularios o en cualquier parte de la página que necesite un botón estándar y simple de aplicar. De forma predeterminada, los botones HTML se presentan con un estilo similar en todas las plataformas, estos estilos se pueden cambiar utilizando CSS.

```
<button></button>
```

- El elemento HTML `<input>` se usa para crear controles interactivos para formularios basados en la web, que reciban datos del usuario. La forma en que `<input>` funciona varía considerablemente dependiendo del valor de su atributo `type`.

```
<input type="text">
```

## CSS.

CSS, las siglas para *Cascading Style Sheets*, u Hojas de estilo en cascada, es una tecnología que nos permite crear una página web de manera más exacta; o, siendo más específicos, que nos permite editar una página web. Si bien se puede decir que HTML (por favor, leer antes el manual de HTML) es la estructura de una página web, CSS es la forma de darle ese atractivo visual a la página, permitiéndonos ordenar los textos, darles color, poner un fondo, crear banners, etc., en fin, todo para una edición de página web.

## Formas de utilizar CSS.

Si bien es recomendable utilizar un editor de texto más profesional, crear hojas de estilo con CSS (cabe recalcar que CSS es una forma de trabajar hojas de estilo, pero existen más) con el simple bloc de notas que traen todas las computadoras.

La base, y lo primordial, es crear un archivo con extensión ".css". Luego de esto, ya podemos comenzar a editar nuestra hoja de estilo. Existen tres formas para utilizar CSS:

La primera es añadir el atributo "style" en la apertura de una etiqueta, y luego definir la característica que esa etiqueta tendrá, por ejemplo:

```
1  <!DOCTYPE html>
2  <html lan="es-ES">
3  <head>
4      <meta charset="utf-8">
5  </head>
6  <body>
7      <p style=color:red>
8          La letra de este párrafo se pondrá en rojo.
9      </p>
10 </body>
11 </html>
```

Nótese primero "<p" para aperturar la etiqueta de párrafo, luego la palabra "style", seguido de la cualidad "color:red".

La siguiente forma, es crear la hoja de estilo, pero dentro la cabecera, o etiqueta "head" del archivo html, por ejemplo:

```
1  <!DOCTYPE html>
2  <html lan="es-ES">
3  <head>
4      <meta charset="utf-8">
5      <style type="text/css">
6          p{
7              color:red;
8              background-color:blue;
9          }
10     </style>
11 </head>
12 <body>
13     <p>
14         La letra de este párrafo se pondrá en rojo y
15         el fondo será azul.
16     </p>
17 </body>
18 </html>
```

Aquí ya apreciamos como se trabaja una hoja de estilo, con la única diferencia que se tiene que especificar lo que es (una hoja de estilo), en este caso, con la etiqueta "<style type='text/css'" (recuerden cerrar esta etiqueta "/style").

Luego, se escribe la o las (en este caso "p") etiquetas que se verán afectadas por este estilo (si fuera las, se enlazan con una coma (,) ) y se abre llave ({), luego, la cualidad, o cualidades que tendrá esta (o estas) etiqueta.

La tercera y última forma, y la que utilizaremos para este manual; es crear un archivo con la extensión ".css", y referenciarlo dentro del archivo html, así:

```
1  <!DOCTYPE html>
2  <html lan="es-ES">
3  <head>
4      <meta charset="utf-8">
5      <link rel="stylesheet" href="ejemplos.css">
6  </head>
7  <body>
8      <p>
9          La letra de este párrafo se pondrá en rojo y
          el fondo será azul.
10     </p>
11 </body>
12 </html>
```

Dentro de la etiqueta link (no se cierra esta etiqueta más que con el signo ">") definimos el tipo de relación con la etiqueta "rel", en este caso "stylesheet", u hoja de estilo en español, y con la etiqueta "href" ponemos la referencia, o sea, la hoja de estilo a utilizar. En el caso de este ejemplo se utilizó el nombre "ejemplos.css". Luego, todos los cambios hechos en la hoja de estilo serán automáticamente aplicados a este archivo html; si la hoja de estilo no estuviera en la misma carpeta que el archivo html, se debe especificar la ruta, así que es recomendable mantener estos dos archivos (y los demás, si se usan, archivos html) juntos.

## Estructura de un documento CSS.

El documento CSS, a diferencia del html, no necesita algo que lo especifique como tal (en el caso de html, "<!DOCTYPE html>" y la etiquetas propias <html> y </html>, simplemente el documento tiene que tener la extensión ".css" al final.

Este documento, que llamaremos ahora "hoja de estilo", cumple un orden jerárquico, las "reglas", llamadas así en CSS, más específicas se aplicarán primero, luego las generales, o sea, si tengo un cambio de color en la etiqueta "body", pero también en las etiquetas "p", este último cambio será el que se aplique.

Sabiendo esto, debe quedar claro que estas reglas son aplicables en las etiquetas que harán un cambio visual en la página web, o sea, "header", "footer", "body", "p", "pre", "div", todas las etiquetas "h" con su número posterior, por supuesto, asterisco (\*) para especificar que el cambio será en todo el documento, etc.; también se puede ser mucho más específico, por ejemplo, se pueden anidar etiquetas dentro de etiquetas, por



ejemplo, si quisiera aplicar un estilo a todos los párrafos dentro de las etiquetas div, sería “div p”, también, los estilos son aplicables en las clases, simplemente se escribe un punto (.) seguido del nombre de la clase, igualmente, se pueden aplicar en los id’s, utilizando, en vez de un punto, el numeral (#), seguido del nombre del id. Otra forma de aplicar hojas de estilo es con el selector de adyacencia, con esto, se aplicará al primer elemento que sigue al especificado de primero, o sea, si yo quisiera aplicarle un estilo a la primera etiqueta “p” que le siga a la finalización de un “div”, sería así: “div + p”, o sea, se aplicará al primer elemento “p” que esté al mismo nivel de los elementos “div”; entre muchas otras formas.

Estas reglas, se abren y cierran con llaves; dentro de estas se escribirán especificaciones que se cumplirán en la página al cumplirse esta regla, por ejemplo, así:

```
1 p{  
2   color:red;  
3   background-color:blue;  
4 }  
5 |
```

La estructura termina siendo: El selector, que sería la etiqueta “p”; la propiedad que sería “color” y el valor, que sería “red”, no olviden en juntar estos dos mediante dos puntos (:) y terminar cada declaración (que sería cada característica que poseerá la regla, “color:red”) con un punto y coma (;).

## Propiedades.

### Color.

En los ejemplos anteriores, se ha utilizado la propiedad “color” varias veces, esta sirve para cambiar el color de algo; sin ninguna propiedad acompañándola, “color” funcionará para cambiar el color de la letra solamente. La sintaxis es:

color: “color a elegir”.

Se puede utilizar el nombre (en inglés del color), o por cantidad de rojos, verdes y azules. Un ejemplo más específico sería:

color: red;

color: #363636;

### Background.

Esta propiedad sirve para editar el fondo de la página; por sí sola, “background” funcionará para definirle un color de fondo (a menos que con la propiedad url se especifique que es una imagen –se verá más adelante), de la misma manera, se le puede agregar la propiedad “color” para ser más específico; estos dos ejemplos harán lo mismo:

background: red;

background-color: red;

Acompañada de la propiedad “image”, se definirá una imagen de fondo, de esta manera:

background-image: url("Desktop/imagenes/imagenbonita.jpg");

Se utiliza "url" para especificar la ruta donde se encuentra la imagen, si la imagen se encuentra en la misma carpeta, simplemente se pondrá el nombre de la misma.

También se pueden las propiedades a esa imagen, acompañando "background" con algunas de las siguientes propiedades: "repeat", "position" y "attachment". Por ejemplo:

background-image: url("Desktop/imagenes/imagenbonita.jpg");

background-repeat: no-repeat;

background-position: 2em 1.5cm;

background-attachment: fixed;

Aquí se especifica la ruta de la imagen, que esta no se repetirá (o sea que sólo saldrá una imagen en el fondo), la posición de la misma (em es reemplazable por cualquier otra unidad de medida de CSS, px, %, em, rem, etc.) y que esta tendrá un comportamiento "fixed", o sea, que no bajará con el *scroll* de la página, estará estática.

Todas estas propiedades pueden ser definidas en una misma declaración, sin importar mucho el orden de las propiedades de la imagen de esta manera:

background: url("Desktop/imagenes/imagenbonita.jpg") no-repeat 2em 1.5cm fixed;

El orden de estas no importa.

## **Border.**

La propiedad "border" sirve para crear un borde. Esta se acompaña de la propiedad "style", seguido del tipo de borde como valor. Existen los siguientes tipos: "dotted", "dashed", "solid", "double", "groove", "ridge", "inset", "outset", "none", "hidden". Por ejemplo:

border-style: solid;

Se pueden definir cuatro tipos de borde por declaración, en el orden de arriba, derecha, abajo, izquierda, por ejemplo:

border-style: solid dashed double ridge;

También se puede acompañar de la propiedad "color" para darle un color (Si no se especifica un color, este se hereda):

border-color: red;

También se puede especificar la anchura del borde, con la propiedad "width", utilizando las medidas anteriormente mencionadas (px, %, em, rem, etc.), o usando uno de los tres valores pre-definidos: "thin", "medium" o "thick". Igualmente, con el color, se puede poner hasta cuatro valores de anchura, de esta manera:

border-width: 2px 5px 10px 1px;

También, se puede ser más específico, escribiendo antes de las propiedades anteriormente mencionadas, las propiedades "top", "right", "bottom" y "left". Por ejemplo:

border-top-style: solid;

Si se utiliza un valor, todos los bordes lo tendrán. Si se especifican dos valores, el borde de arriba y el de abajo tendrán el primer valor y el segundo valor lo tendrán los de izquierda y derecha. Si se especifican tres, el primer valor lo tendrá el borde de arriba, el segundo valor lo tendrán los bordes de izquierda y derecha, y el tercero lo tendrá el borde de abajo. Si se utilizan cuatro, será como se mencionó anteriormente.

También es posible aplicar todas las propiedades en una sola declaración:

```
border-top: 5px solid red;
```

Por último, el borde puede ser redondeado, agregándole un radio al mismo, con la propiedad “radius”, por ejemplo:

```
border-radius: 5px;
```

## **Margins y Padding.**

La propiedad “margin” sirve para crear distancia entre un borde definido y el espacio dado a la etiqueta. Se trabaja con unidades de medida y de la misma forma que los bordes, o sea, se pueden definir hasta cuatro, siguiendo la misma lógica anteriormente explicada, o especificando que margen será el afectado, por ejemplo:

```
margin: 25px;
```

```
margin-top: 25px;
```

```
margin: 25px 10px 8px 15px;
```

También tiene valores pre definidos, “auto”, para centrar el elemento horizontalmente dentro de su contenedor. Y la propiedad “inherit” para heredar un margen.

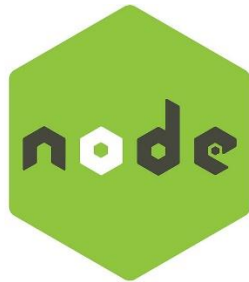
La propiedad “padding” sirve para crear una distancia entre los bordes y el contenido. Funciona de la misma manera que el “margin” pero sin los valores pre definidos “auto” ni “inherit”.

## **Width y Height.**

La propiedad “width” se utiliza para definir la anchura de un elemento, utilizando la unidad de medida que se prefiera y valores predefinidos. La propiedad “height” sirve para definir la altura de un elemento, funcionando de la misma manera que la propiedad “width”.

## JavaScript.

JavaScript es un lenguaje de tipo script, interpretado, y cuya plataforma de ejecución más frecuente son los navegadores. JavaScript también puede correr fuera de los navegadores gracias a entornos como Node JS, que permite ejecutar código JavaScript para tareas que requieran integración de módulos o ejecución de código en servidores.



## ¿Para qué nos sirve JavaScript en el desarrollo Web?

En la actualidad JavaScript es lenguaje más utilizado en el mundo del desarrollo, ya que está presente en prácticamente todo el internet. Hasta el momento sabes que con los otros dos lenguajes pilares de la Web, HTML y CSS, podíamos estructurar y estilizar respectivamente, páginas web, ahora con JavaScript podemos darle interactividad.

A continuación, se mencionan los usos más frecuentes que se le da a JavaScript:

- **Modificación de la estructura.** Con JS podemos modificar de forma dinámica la estructura y propiedades de un sitio Web. Podemos agregar y eliminar atributos de etiquetas HTML e incluso las etiquetas mismas.
- **Procesos asíncronos.** Una de las características de JavaScript es poder realizar procesos independientes al flujo principal. JavaScript trabaja con un único hilo de ejecución y el asincronismo viene a cumplir con la función de ejecutarse procesos paralelos sin detener el flujo principal y único. La mayoría de ejecuciones de esta naturaleza se realizan cuando se solicitan datos a servidores y se cargan a la página modificando su estructura, esto se da, por ejemplo, cuando una aplicación consigue información sin recargar el sitio.
- **Eventos.** Con JavaScript podemos asignar a eventos a elementos HTML, y que cuando ocurra cierto escenario, se ejecuten indicaciones para realizar algún cambio u otra actividad.
- **Validaciones.** JavaScript se utiliza con mucha frecuencia para validar métodos de entrada de información desde el área del cliente. Estas pueden ser consultas asíncronas o verificaciones a partir del contenido HTML.

## ¿Qué se va a abarcar?

En esta ocasión omitiremos los detalles básicos del lenguaje (que podrían ser de mayor utilidad para otras tareas) para enfocarnos en cómo usar de forma práctica el lenguaje para tareas comunes en el desarrollo Web.

Entre lo que se va a explicar se encontrar las distintas formas de incorporar código JavaScript a un documento HTML, modificar elementos seleccionándolos desde el DOM, asignar un evento a un elemento HTML.

## ¿Cómo agregar JavaScript?

Básicamente, podemos agregar de dos formas: directamente en el HTML con la ayuda de etiquetas `<script></script>` o bien con un archivo externo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>

<body>
  <div class="box" id="box"></div>

  <script>

    var box = document.getElementById("box");
    box.style.height = "400px";
    box.style.width = "400px";
    box.style.background = "green";

  </script>
</body>
</html>
```

```


JS prueba.js ▸ ...
1 // prueba.js
2
3
4 var box = document.getElementById("box");
5 box.style.height = "400px";
6 box.style.width = "400px";
7 box.style.background = "green";

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>

<body>
  <div class="box" id="box"></div>

  <script src="prueba.js"></script>
</body>
</html>

```



## Seleccionando un elemento con JavaScript.

Para poder seleccionar un elemento es necesario asignarle una propiedad identificativa, por lo general se escoge un id:

```
<div class="box" id="box"></div>
```

Ahora podemos seleccionarlo desde el un archivo JavaScript:

```

var box = document.getElementById("box");
box.style.height = "400px";
box.style.width = "400px";
box.style.background = "green";
box.style.border = "solid red 1px";

```

En la asignación de la variable “box” escribimos “document” para hacer referencia al archivo desde el cual es utilizado el archivo. Luego, con `getElementById` especificamos el id del elemento que queremos obtener.

Ahora, la variable `box` representa al elemento que teníamos en el archivo HTML. Más que variable, “box” es un objeto, y como objeto contiene propiedades. La propiedad “style”, como podemos ver en la captura nos permite asignarle estilos como si bien se tratara de un archivo CSS, sólo procura usar un editor con autocompletado para hacer esto más fácil. Te aparecerán las propiedades de estilos, y las reconocerás si ya dominas CSS.

Ahora veámoslo en el navegador:

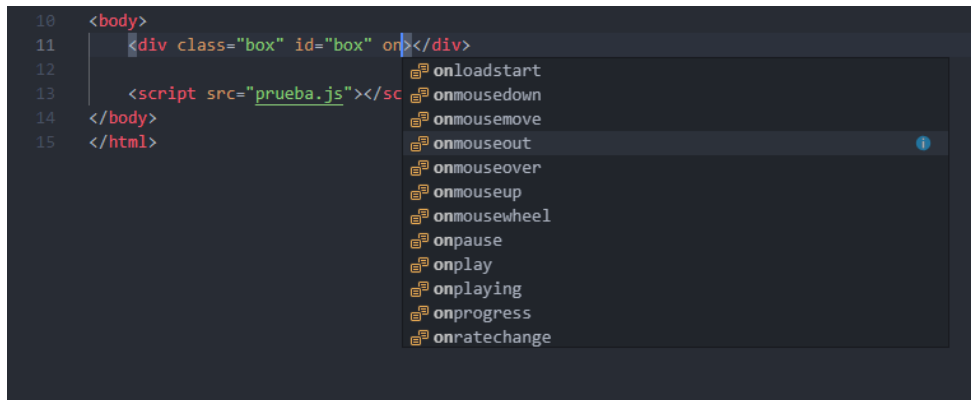


Los estilos se han aplicado a como lo especificamos. También podemos usar la función `getElementsByClassName`, sólo que este nos asociará todos los elementos que contengan la clase especificada.

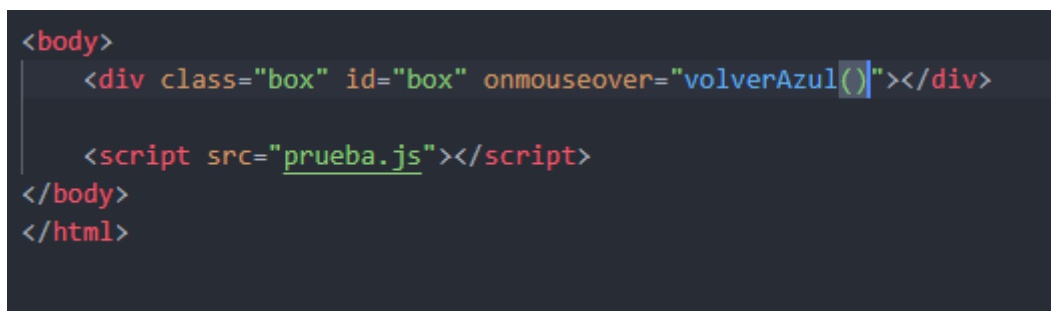
Ya probamos con estilos, ¿te parece si probamos ahora con eventos?

## Eventos con JavaScript.

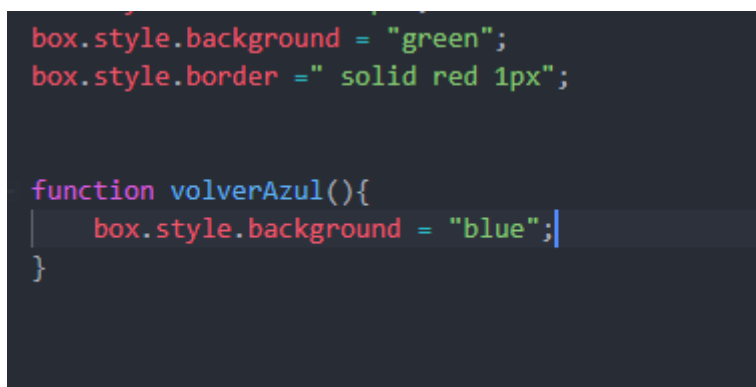
Para este paso podemos utilizar los eventos de HTML, en los cuales se especifica la función de JavaScript que se ejecutará cuando el evento se detecte.



Como podemos ver, tenemos gran cantidad de eventos escribiendo tan solo las iniciales “on”, yo escogeré onMouseOver que sería un equivalente a :hover en CSS.

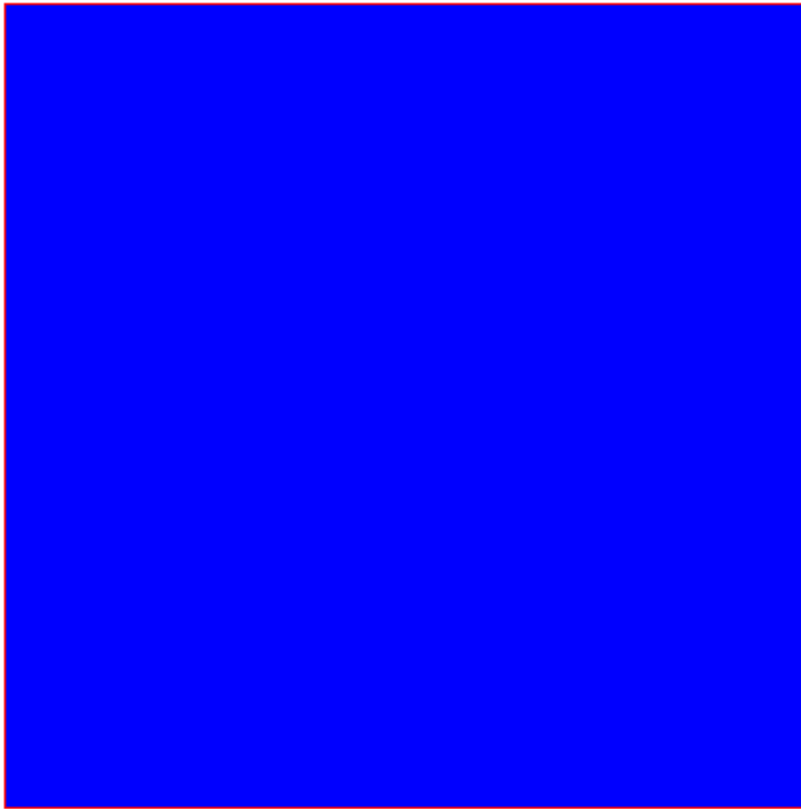


Una vez especificada la función, procedemos a codificarla en el archivo JS:





Veamos qué pasa al poner el mouse sobre el cuadro:



¡Puedes comprobarlo tú mismo si quieres!

Ahora, probemos otra alternativa. Usando `addEventListener`:

```
/* function volverAzul(){  
  box.style.background = "blue";  
} */  
  
box.addEventListener("mouseover", () =>{  
  box.style.background = "blue";  
});
```

El primer parámetro es el tipo de evento que esperamos, que, si cuentas con un editor con autocompletado, no se te hará difícil este paso.

El resultado es completamente el mismo que el anterior ¡Pruébalo!

## Referencias.

<https://definicion.de/html/>

<http://disenoyelaboraciondepaginasweb.blogspot.com/2012/04/partes-de-un-documento-html.html>

<https://developer.mozilla.org/es/docs/Web/HTML/Elemento/link>

<http://www.mclibre.org/consultar/htmlcss/html/html-favicon.html>

<https://lenguajehtml.com/p/html/semantica/etiquetas-html-de-texto>

[https://developer.mozilla.org/es/docs/Learn/HTML/Multimedia\\_and\\_embedding/Images\\_in\\_HTML](https://developer.mozilla.org/es/docs/Learn/HTML/Multimedia_and_embedding/Images_in_HTML)

<http://www.htmlquick.com/es/tutorials/tables.html>

<https://developer.mozilla.org/es/docs/Web/HTML/Elemento/div>

[https://es.wikipedia.org/wiki/Form\\_\(etiqueta\\_HTML\)](https://es.wikipedia.org/wiki/Form_(etiqueta_HTML))

<https://developer.mozilla.org/es/docs/Web/HTML/Elemento/button>

<https://developer.mozilla.org/es/docs/Web/HTML/Elemento/input>

CSS width Property, recuperado de: [https://www.w3schools.com/cssref/pr\\_dim\\_width.asp](https://www.w3schools.com/cssref/pr_dim_width.asp)

CSS height Property, recuperado de: [https://www.w3schools.com/cssref/pr\\_dim\\_height.asp](https://www.w3schools.com/cssref/pr_dim_height.asp)

CSS Padding, recuperado de: [https://www.w3schools.com/css/css\\_padding.asp](https://www.w3schools.com/css/css_padding.asp)

CSS Borders, recuperado de: [https://www.w3schools.com/css/css\\_border.asp](https://www.w3schools.com/css/css_border.asp)

CSS Margins , recuperado de: [https://www.w3schools.com/css/css\\_margin.asp](https://www.w3schools.com/css/css_margin.asp)

Propiedad Background, recuperado de: <https://uniwebsidad.com/libros/referencia-css2/background>

Tutorial CSS3: El uso de reglas, recuperado de: <https://www.dariobf.com/tutorial-css3-regla-cs/>

CSS Selector Reference, recuperado de: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

