

EXERCISES QUE TRABAJAREMOS EN LA CUE

- EXERCISE 1: INCORPORANDO ESTILOS
- EXERCISE 2: MODELO DE CAJAS
- EXERCISE 3: UNIDADES DE MEDIDA
- EXERCISE 4: INSPECTOR DE ELEMENTOS DE UN NAVEGADOR

EXERCISE 1: INCORPORANDO ESTILOS

Como previamente comentamos, **CSS** u Hojas de Estilo en Cascada es el lenguaje que se utiliza para dar todo el aspecto visual y volver atractivos nuestros sitios web.

Este lenguaje se utiliza en conjunto con algún lenguaje de marcado como es **HTML**. La relación entre ambos (**HTML** y **CSS**) es bastante fuerte y, aunque técnicamente hablando no es una obligación el uso de **CSS** en nuestros proyectos, es común su incorporación para favorecer los diseños.

Para comenzar a trabajar con **CSS** e incorporar estilos, debemos crear un proyecto y en él, antes que todo, un archivo **HTML** para poder dar estructura.

Una vez que tengamos definido el archivo principal con su estructura básica, vamos a colocar una etiqueta `<div>` y dentro de ella, una etiqueta `<p>` con `lorem ipsum`.

Esto lo repetiremos una vez más, quedando de la siguiente manera:

```
1  <div>
2      <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
3  Sunt, architecto.
4      Repudiandae labore asperiores non,
5      dolores ipsa dolor amet suscipit quas laborum at iste
6  mollitia natus ea
7      sint quos illo enim!</p>
8  </div>
9
10 <div>
11     <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
12 Sunt, architecto.
```

```
13         Repudiandae labore asperiores non,  
14         dolores ipsa dolor amet suscipit quas laborum at iste  
15 mollitia natus ea  
16         sint quos illo enim!</p>  
17     </div>
```

De esta forma hemos creado nuestra primera estructura **HTML**. Abramos nuestro archivo para observar en el navegador como se ve.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Con esto tenemos la primera estructura. Evidentemente, si bien cumple la función de comunicar algo, estéticamente no es muy atractiva.

Existen tres formas de aplicar estilo a un elemento, siendo la tercera forma que veremos, la que usaremos de aquí en adelante y consideraremos las otras dos formas, como una mala práctica que solo debemos conocer.

APLICANDO ESTILO EN LA ETIQUETA

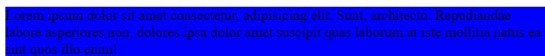
Una mala práctica y el motivo por el cual se creó **CSS** es la aplicación de estilos directamente en la etiqueta. Esto genera que el código se vuelva difícil de mantener a medida que avanza el proyecto y dificulta los procesos de creación del mismo.

Vamos a definir, en la primera etiqueta **<div>** un estilo de color de fondo azul con un ancho de **600px** (las medidas las veremos más adelante). Nuestro código se verá de la siguiente forma:

```
1 <div style="background-color: blue; width: 600px;">  
2     <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.  
3 Sunt, architecto.  
4         Repudiandae labore asperiores non,  
5         dolores ipsa dolor amet suscipit quas laborum at iste  
6 mollitia natus ea  
7         sint quos illo enim!</p>  
8     </div>
```

```
9
10     <div>
11         <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
12 Sunt, architecto.
13         Repudiandae labore asperiores non,
14         dolores ipsa dolor amet suscipit quas laborum at iste
15 mollitia natus ea
16         sint quos illo enim!</p>
17     </div>
```

Y si guardamos los cambios y actualizamos la página en nuestro navegador veremos el siguiente resultado:



Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Podemos ver que hemos dado estilo al elemento `<div>` y, por lo tanto, a todo lo que este elemento envuelve, utilizando el atributo `style`.

Parece muy simple y uno se podría preguntar ¿sí mi código es así de pequeño por qué no utilizar esta forma?

Supongamos que nuestro segundo `<div>` requiere el mismo estilo, debemos volver a escribir exactamente lo mismo en la otra etiqueta `<div>` para lograrlo.

```
1     <div style="background-color: blue; width: 600px;">
2         <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
3 Sunt, architecto.
4         Repudiandae labore asperiores non,
5         dolores ipsa dolor amet suscipit quas laborum at iste
6 mollitia natus ea
7         sint quos illo enim!</p>
8     </div>
9
10    <div style="background-color: blue; width: 600px;">
11        <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
12 Sunt, architecto.
13        Repudiandae labore asperiores non,
14        dolores ipsa dolor amet suscipit quas laborum at iste
15 mollitia natus ea
```



```
16      sint quos illo enim!</p>
17    </div>
```

Obteniendo como resultado:

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

En otras palabras, consiguiendo el objetivo, pero, teniendo que darnos a la tarea de escribir las instrucciones dos veces. Esto, a medida que el código se vuelve más grande y complejo, se convierte en una tarea tediosa y, repetir instrucciones innecesariamente nunca es buena opción (y se considera una muy mala práctica).

APLICANDO ESTILO EN EL DOCUMENTO HTML

Otra forma de realizar esto, es utilizando etiquetas `<style>` y dentro de ellas colocando la misma indicación. En este caso, la mantención también se hace algo compleja, ya que el código **HTML** y el código **CSS** van en conjunto en el mismo documento.

Antes de continuar debemos detenernos a hablar de la sintaxis de **CSS**.

SINTAXIS DE CSS

CSS se compone de dos partes: La primera parte se conoce como selector y la segunda parte se conoce como declaración.



El selector indica sobre quien se realizará una acción determinada y la declaración indica que es lo que se hará.

Primero que todo, vamos a eliminar las indicaciones que tenemos en nuestros `<div>` y a ambos le colocaremos `class` "contenedor", mientras que al segundo `<div>` también le colocaremos un `id` "división".

```

1 <div class="contenedor">
2     <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
3 Sunt, architecto.
4         Repudiandae labore asperiores non,
5         dolores ipsa dolor amet suscipit quas laborum at iste
6 mollitia natus ea
7         sint quos illo enim!</p>
8 </div>
9
10 <div class="contenedor" id="division">
11     <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
12 Sunt, architecto.
13         Repudiandae labore asperiores non,
14         dolores ipsa dolor amet suscipit quas laborum at iste
15 mollitia natus ea
16         sint quos illo enim!</p>
17 </div>

```

Luego de esto, vamos a agregar las etiquetas `<style>` al inicio del `<body>` y vamos a comenzar a dar estilo a nuestros elementos mientras, al mismo tiempo, iremos conociendo los tipos de selectores.

TIPOS DE SELECTOR

- **Selector universal:** Se utiliza para seleccionar todos los elementos de una página. No se utiliza habitualmente porque no es común aplicar el mismo estilo a toda la página. Se simboliza con un asterisco (*).

```
1 *{  
2     margin: 0;  
3     padding: 0;  
4 }
```

- **Selector de etiqueta o tipo:** selecciona todos los elementos del documento que coinciden con el valor del selector. Todos los elementos coincidentes compartirán el mismo estilo

```
1 div{  
2     background-color: green;  
3 }  
4
```

Lorem ipsum dolor sit amet consectetur adipiscing elit. Suspendisse nulla. Repellendus lobortis sapien. Sed dolor sit amet suscipit. Quis lobortis at iure mollitia. Nunc ex nisi quis. Illo enim.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Suspendisse nulla. Repellendus lobortis sapien. Sed dolor sit amet suscipit. Quis lobortis at iure mollitia. Nunc ex nisi quis. Illo enim.

- **Selector de clase:** selecciona todos los elementos que contengan un atributo **class** con un valor determinado. Se simboliza indicando un punto, más el valor del **class**, por ejemplo **.inicio**. Todos los elementos que coincidan con la **class** indicada compartirán el mismo estilo.

```
1 .contenedor{  
2     background-color: blue;  
3 }  
4
```

Lorem ipsum dolor sit amet consectetur adipiscing elit. Suspendisse nulla. Repellendus lobortis sapien. Sed dolor sit amet suscipit. Quis lobortis at iure mollitia. Nunc ex nisi quis. Illo enim.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Suspendisse nulla. Repellendus lobortis sapien. Sed dolor sit amet suscipit. Quis lobortis at iure mollitia. Nunc ex nisi quis. Illo enim.

- **Selector de ID:** selecciona un elemento del documento que contenga un determinado valor de **id**. El **id** es único e irrepetible. Se simboliza con un signo numeral más el valor del **id**, por ejemplo **#titulo**

```
1 #division{
2     color: coral;
3 }
```

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Sobre el estilo que hemos aplicado previamente, **background-color** es la indicación que utilizaremos para dar color de fondo a un elemento. **Color** es la indicación que utilizaremos para definir un color de letra del texto de un elemento.

El código completo con la última indicación se verá así:

```
1 <body>
2
3     <style>
4         .contenedor{
5             background-color: blue;
6         }
7     </style>
8     <div class="contenedor">
9         <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
10      Sunt, architecto.
11             Repudiandae labore asperiores non,
12             dolores ipsa dolor amet suscipit quas laborum at iste
13      mollitia natus ea
14             sint quos illo enim!</p>
15     </div>
16
17     <div class="contenedor" id="division">
18         <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
19      Sunt, architecto.
20             Repudiandae labore asperiores non,
21             dolores ipsa dolor amet suscipit quas laborum at iste
22      mollitia natus ea
23             sint quos illo enim!</p>
24     </div>
```

```
25 </body>
```

APLICANDO ESTILO CON UNA HOJA DE ESTILO

Finalmente, veremos cual sería la forma correcta de aplicar el estilo a través de una hoja de estilo, es decir, un archivo de extensión **CSS**.

Lo primero que haremos será eliminar las etiquetas **<style>** y su contenido y mantendremos las **class** y el **id** que indicamos previamente en los **<div>**.

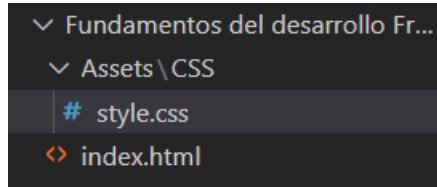
```
1 <body>
2
3   <div class="contenedor">
4       <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
5       Sunt, architecto.
6           Repudiandae labore asperiores non,
7           dolores ipsa dolor amet suscipit quas laborum at iste
8       mollitia natus ea
9           sint quos illo enim!</p>
10  </div>
11
12  <div class="contenedor" id="division">
13      <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
14      Sunt, architecto.
15          Repudiandae labore asperiores non,
16          dolores ipsa dolor amet suscipit quas laborum at iste
17      mollitia natus ea
18          sint quos illo enim!</p>
19  </div>
20 </body>
```

Ahora nuestro documento principal ha quedado limpio y solo contiene código **HTML**, lo que facilita su lectura.

Luego, en nuestro proyecto vamos a crear la carpeta **Assets** que contendrá imágenes u otros archivos. Dentro de esa carpeta crearemos la carpeta **CSS**.

Finalmente, dentro de esa carpeta crearemos un nuevo archivo llamado **"style.css"**. La extensión **CSS** nos indica que es un archivo de estilos.

La estructura de carpetas quedará de la siguiente forma:



Ahora ya tenemos el archivo creado, pero aún no se encuentra ligado con el archivo **HTML**. Debemos realizar esto utilizando la etiqueta `<link>` en el `<head>` del **HTML**.

Esta etiqueta tiene el atributo `rel` que se utiliza para describir la relación entre los dos documentos y el atributo `href` que se utiliza para indicar la ruta al documento enlazado.

```
1 <link rel="stylesheet" href="Assets/CSS/style.css">
```

De esta forma, el archivo **CSS** y el archivo **HTML** han quedado relacionados.

Finalmente, nuestro archivo **HTML** quedará así:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-
8 scale=1.0">
9     <link rel="stylesheet" href="Assets/CSS/style.css">
10    <title>CSS</title>
11 </head>
12
13 <body>
14
15     <div class="contenedor">
16         <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
17 Sunt, architecto.
18         Repudiandae labore asperiores non,
19         dolores ipsa dolor amet suscipit quas laborum at iste
20 mollitia natus ea
21         sint quos illo enim!</p>
22     </div>
```

```
23
24     <div class="contenedor" id="division">
25         <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
26 Sunt, architecto.
27         Repudiandae labore asperiores non,
28         dolores ipsa dolor amet suscipit quas laborum at iste
29 mollitia natus ea
30         sint quos illo enim!</p>
31     </div>
32 </body>
33
34 </html>
```

En nuestro archivo **CSS** vamos a comenzar a darle estilo a nuestra página web, pero, como ya sabemos utilizar los selectores, vamos a continuar aprendiendo la Jerarquía en **CSS**.

ORDEN JERARQUICO EN CSS

Cuando definimos un estilo, la última instrucción será la que prevalezca siempre que estamos refiriéndonos a un mismo selector, pero, en caso contrario, la indicación más específica será la que prevalecerá.

Vamos a indicar, utilizando el selector de etiqueta **<div>** que nuestros elementos tendrán un color de fondo rojo y luego un color de fondo amarillo.

```
1 div{
2     background-color: red;
3 }
4 div{
5     background-color: yellow;
6 }
```

De esta forma, obtenemos como resultado que ambos elementos tendrán un color de fondo amarillo, ya que prevalece la última indicación.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Ahora, eliminaremos las indicaciones previas y vamos a llamar al **id** división con color de letra azul, luego, llamaremos a la **clase** contenedor con color de letra rojo y finalmente a la etiqueta **<div>** con color verde.

```
1 #division{
2     color: blue;
3 }
4 .contenedor{
5     color: red;
6 }
7 div{
8     color: green;
9 }
```

Obteniendo como resultado:

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!
 Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

El color de letra del elemento que tiene indicado un **id** se mantiene en azul a pesar de no ser la última indicación, y tener posteriormente una indicación con su **clase** y tu **tag**, ya que el **id** es el más específico.

Para el caso del elemento con **clase** definida, se mantiene con la indicación dada a la **clase** y se ignora la indicación posterior dada a la **etiqueta**, porque se respeta la indicación más específica.

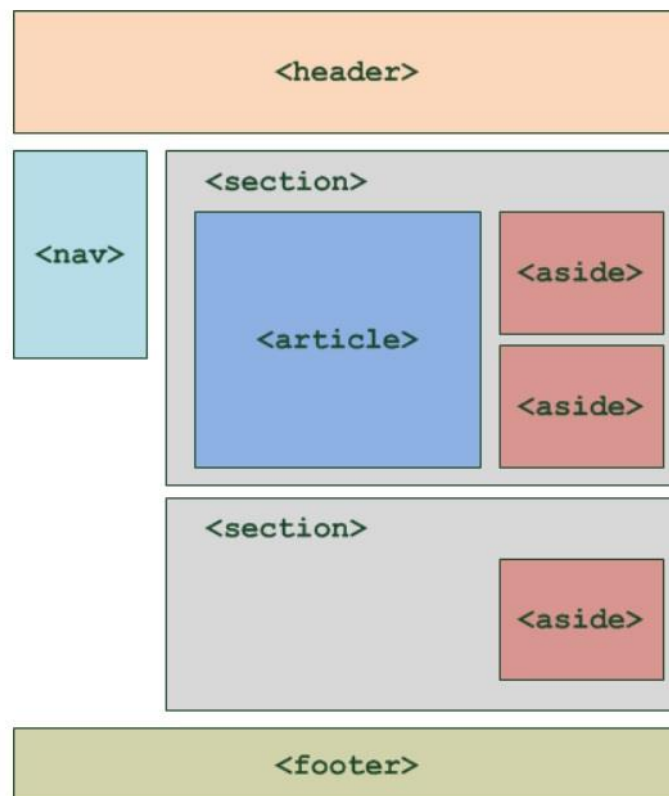
EXERCISE 2: MODELO DE CAJAS

Es fundamental, al trabajar con **CSS** hablar del modelo de cajas.

Las cajas de una página web se crean automáticamente cada vez que se inserta una etiqueta **HTML**. Estas son cajas rectangulares que encierran los contenidos del elemento insertado.

Las cajas con una característica muy importante de CSS ya que condiciona todo el diseño de una página web. Estas cajas no son visibles a simple vista.

En la siguiente imagen, podemos ver un esquema general de las cajas de una página web utilizando las etiquetas semánticas.



Para realizar este ejemplo, deberemos modificar levemente el documento **HTML** que tenemos hasta ahora, incorporando en él, encima de las etiquetas **<div>** dos etiquetas **<a>** para generar hipervínculos.

Para esta ocasión, no colocaremos nada en el atributo **href** ya que nos centraremos en otros aspectos.

Nuestro código **HTML** quedará así:

```
1 <body>
2   <a href="">Hola mundo</a>
3   <a href="">Hola mundo</a>
4
5
```

```
6   <div class="contenedor">
7       <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
8   Sunt, architecto.
9       Repudiandae labore asperiores non,
10      dolores ipsa dolor amet suscipit quas laborum at iste
11   mollitia natus ea
12      sint quos illo enim!</p>
13   </div>
14
15   <div class="contenedor" id="division">
16       <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
17   Sunt, architecto.
18       Repudiandae labore asperiores non,
19       dolores ipsa dolor amet suscipit quas laborum at iste
20   mollitia natus ea
21       sint quos illo enim!</p>
22   </div>
23 </body>
```

En el archivo **style.css** eliminaremos las indicaciones dadas previamente, actualizaremos nuestra página web y obtendremos:

[Hola mundo](#) [Hola mundo](#)

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

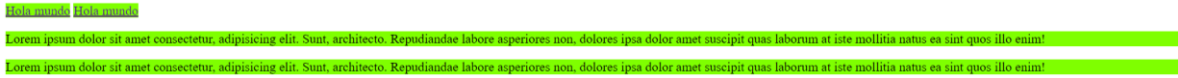
Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!

Ahora que tenemos nuestros 4 elementos, en el archivo **CSS** les indicaremos un color de fondo.

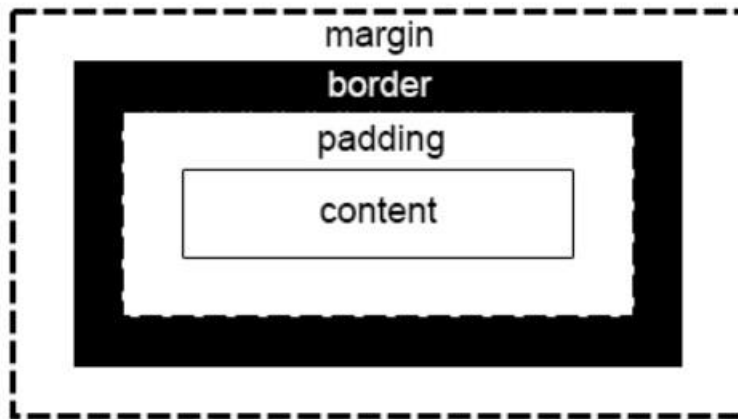
Para agregar un color de fondo a más de un elemento a la vez, solo separaremos los elementos por coma.

```
1 div, a {
2     background-color: chartreuse;
3 }
```

Al actualizar nuestra página web (siempre después de guardar los cambios) obtendremos lo siguiente:



Como podemos observar, todo lo pintado en verde es una caja completa. Las cajas de cada elemento incluyen el contenido, el *padding*, el margen y el borde.



Analizando un poco más en detalle estos cuatro componentes de una caja, podemos decir que:

- **Contenido:** Es el código HTML con texto, imágenes, listas, etc. que se encuentra dentro de la caja.
- **Padding:** Espacio libre entre el contenido y el borde.
- **Border:** Línea que encierra el contenido y el relleno.
- **Margin:** Espacio libre entre el borde y otros elementos adyacentes (si los hubiera) o el borde de su elemento contenedor.

Analizar esto nos permitirá comprender como ocupar el espacio de un elemento.

Al `<div>` con `id` división, le realizaremos una modificación de `padding`, borde y margen para poder revisar como se modifican estos puntos.

```
1 #division{  
2   margin: 10%;  
3   padding: 10%;
```

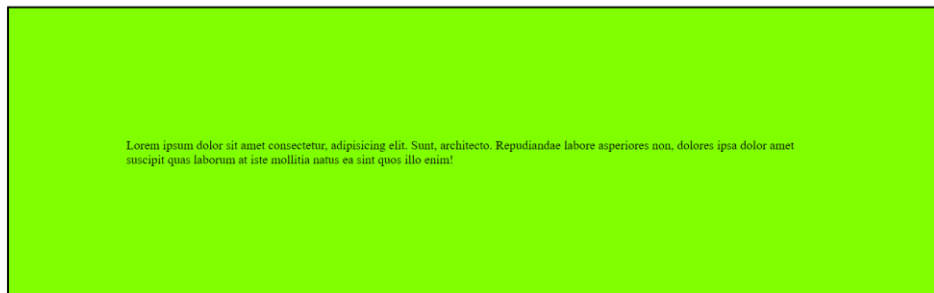
```
4 border-color: black;  
5 border-style: solid;  
6 }
```

Cabe destacar que, en el caso del *margin* y el *padding* se puede dar una indicación específica para cada uno de los lados (*top* para arriba, *left* para izquierda, *right* para derecha o *bottom* para abajo) o una indicación general para los 4 lados, como ocurre en este ejemplo. Para el caso del borde, indicamos que sea sólido para que se vea realmente como una línea y definimos un color.

Si actualizamos nuestra página web, podremos encontrar el siguiente resultado:

[Ver código](#) [Ver resultado](#)

>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sunt, architecto. Repudiandae labore asperiores non, dolores ipsa dolor amet suscipit quas laborum at iste mollitia natus ea sint quos illo enim!



Acá podemos ver como el elemento seleccionado cuenta con un **margen**, que es la separación entre el borde del elemento y el resto de elemento. Un **borde** definido y un **padding** que es la distancia entre el contenido del elemento y su borde.

De esta forma, podemos manipular las distancias dentro de cada caja de cada elemento.

BLOCK V/S INLINE

Cuando hablamos del modelo de caja no solamente debemos referirnos a los componentes de una caja si no también, a los tipos de cajas.

En general hay dos tipos de cajas: en bloque y en línea. Las cajas en bloque fuerzan un salto de línea al llegar al final. Por defecto usan **block** como tipo de visualización y ocupan el 100% del espacio disponible. Las cajas en línea no fuerzan un salto de línea y usan la visualización **inline**. No se les puede dar un alto y un ancho y ocupan solamente el espacio necesario.

Para profundizar este tema un poco más, observaremos nuevamente nuestros elementos **<a>** y **<div>**. Podemos ver que los dos elementos **<a>** se han posicionado uno al lado del otro, es porque los elementos como los enlaces son de tipo **inline** o en línea.

Este tipo de elementos ocupa solamente el espacio necesario y se alinean siempre uno al lado del otro, sin la necesidad de un salto de línea si aún queda espacio en esa línea.

Es importante tener en cuenta que a esta clase de elementos no se les puede dar un ancho ni un alto (es decir, el tamaño de su caja no es modificable con **CSS**).

Por el contrario, si observamos el primer **<div>** que no hemos modificado, podemos notar que la caja ocupa el 100% del espacio disponible (incluso si el contenido termina antes) y siempre realiza un salto de línea al final, agrupándose siempre uno debajo del otro.

Este tipo de elemento se conoce como elemento **block** o en bloque y con **CSS** podemos modificar su ancho y alto.

```
1 div{  
2   height: 200px;  
3   width: 300px;  
4 }
```

En nuestro navegador obtendremos:

Hola mundo Hola mundo

Lorem ipsum dolor sit amet consectetur,
adipiscing elit. Sunt, architecto. Repudiandae
labore asperiores non, dolores ipsa dolor amet
suscipit quas laborum at iste mollitia natus ea
sint quos illo enim!

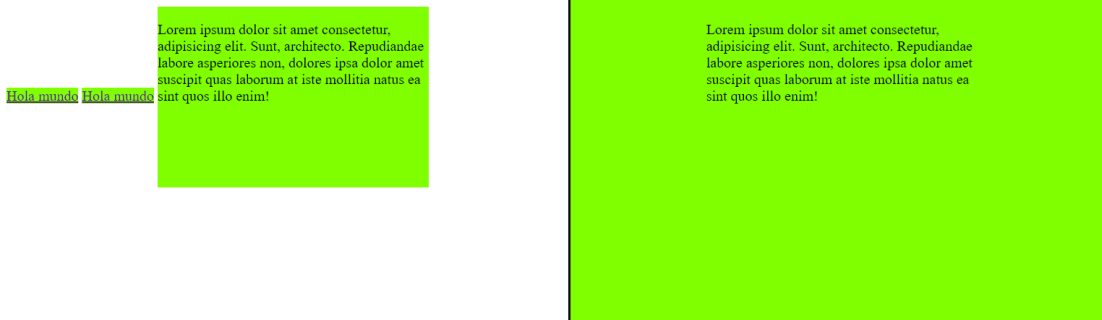
Evidentemente, hemos podido modificar, haciendo uso de instrucciones **CSS**, el tamaño de la caja en bloque.

Pero, no solo eso podemos hacer. Como previamente dijimos, este elemento es en bloque, por lo que por defecto se posicionará siempre debajo del resto y generará un salto de línea incluso quedando espacio disponible.

En ocasiones necesitaremos que estos elementos, en cambio, se posicionen uno al lado del otro, es decir, se comporten como elementos en línea. Para eso, podemos utilizar la propiedad **display** con el valor **"inline-block"** para forzar que se comporten como elementos en línea.

```
1 div{  
2   height: 200px;  
3   width: 300px;  
4   display: inline-block;  
5 }
```

Al utilizar esta propiedad con este valor, en nuestra página web obtendremos como resultado:



Pudiendo modificar así el tipo de elemento.

Esto se puede realizar con los elementos en línea y convertirlos en elementos en bloque usando la propiedad **display** con el valor **block**.

EXERCISE 3: UNIDADES DE MEDIDA

Las unidades de medida son aquellas que nos permiten indicar un tamaño (por ejemplo, ancho) de una caja o imagen. Hasta el momento hemos trabajado con porcentaje(%) o pixeles (px), pero ¿Cuáles otras unidades de medida existen?,¿Cual nos conviene utilizar?

UNIDADES ABSOLUTAS

Una medida indicada con unidades absolutas son un tipo de medida fija que no cambia, ya que su valor no depende de ningún otro valor de referencia. Son unidades poco flexibles e inadecuadas para la web actual ya que no se adaptan a las distintas resoluciones de pantalla. A su vez, la ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar.

| Unidad | Significado | Medida aproximada |
|--------|--------------|-------------------|
| in | Pulgadas | 1in = 25.4mm |
| cm | Centímetros | 1cm = 10mm |
| pc | Picas | 1pc = 4.23mm |
| mm | Milímetros | 1mm = 1mm |
| pt | Puntos | 1pt = 0.35mm |
| px | Píxels | 1px = 0.26mm |
| Q | Cuarto de mm | 1Q = 0.248mm |

UNIDADES RELATIVAS

Las unidades relativas son unidades cuyo valor siempre esta referenciado a otro valor como la densidad de pantalla, la resolución, etc. Son las más utilizadas en el diseño web y son ideales para trabajar en dispositivos con diferentes tamaños por la flexibilidad con que se adaptan a diferentes medios.

| Unidad | Significado | Medida aproximada |
|------------|--------------|--|
| em | «M» | 1em = tamaño de fuente establecida en navegador |
| ex | «X» (~0.5em) | 1ex = ~ mitad del tamaño de fuente del navegador |
| ch | «zero width» | 1ch = tamaño de ancho del cero (0) |
| rem | «root M» | 1rem = tamaño fuente raíz |
| % | Porcentaje | Relativa a herencia (contenedor padre) |

Las unidades **em** y **ex** son unidades que no fueron creadas por **CSS** si no que llevan décadas siendo utilizadas en el campo de la tipografía.

La unidad **em** hace referencia al tamaño actual de la fuente que ha sido establecido en el navegador, comúnmente un valor aproximado a 16px. Por lo tanto, 1em es el valor establecido por el usuario, 2em es justo el doble y 0.5em es justo la mitad de aquel valor.

Por su parte, la unidad **rem** permite establecer un tamaño para el documento en general, utilizando por ejemplo el elemento **<body>**. En otras palabras, es un tamaño escalado del **Font-size** que le indiquemos al **<body>**.

Para los porcentajes, hay que tener en cuenta al elemento padre, esto quiere decir que, si un elemento está dentro de otro elemento y le indicamos un tamaño del 80%, ese tamaño corresponderá al 80% del tamaño total del elemento padre.

UNIDADES FLEXIBLES

Las unidades flexibles se consideran unidades de nueva generación cuyo tamaño depende del **viewport** (*región visible de la página web en el navegador*). Con estas unidades podemos hacer referencia a un porcentaje concreto del tamaño específico que tengamos en la ventana del navegador, independientemente de si es redimensionado o no.

| Unidad | Significado | Medida aproximada |
|-------------|------------------|--|
| vw | viewport width | 1vw = 1% ancho de navegador |
| vh | viewport height | 1vh = 1% alto de navegador |
| vmin | viewport minimum | 1vmin = 1% de alto o ancho (el mínimo) |
| vmax | viewport maximum | 1vmax = 1% de alto o ancho (el máximo) |

La unidad **vw** hace referencia al ancho del **viewport**, mientras que **vh** hace referencia al alto. Por ejemplo, si utilizamos **100vw** estaremos haciendo referencia al 100% del ancho del navegador, o sea, todo lo que se está viendo de ancho en pantalla, mientras que si indicamos **50vw** estaremos haciendo referencia a la mitad del ancho del navegador.

EXERCISE 4: INSPECTOR DE ELEMENTOS DE UN NAVEGADOR

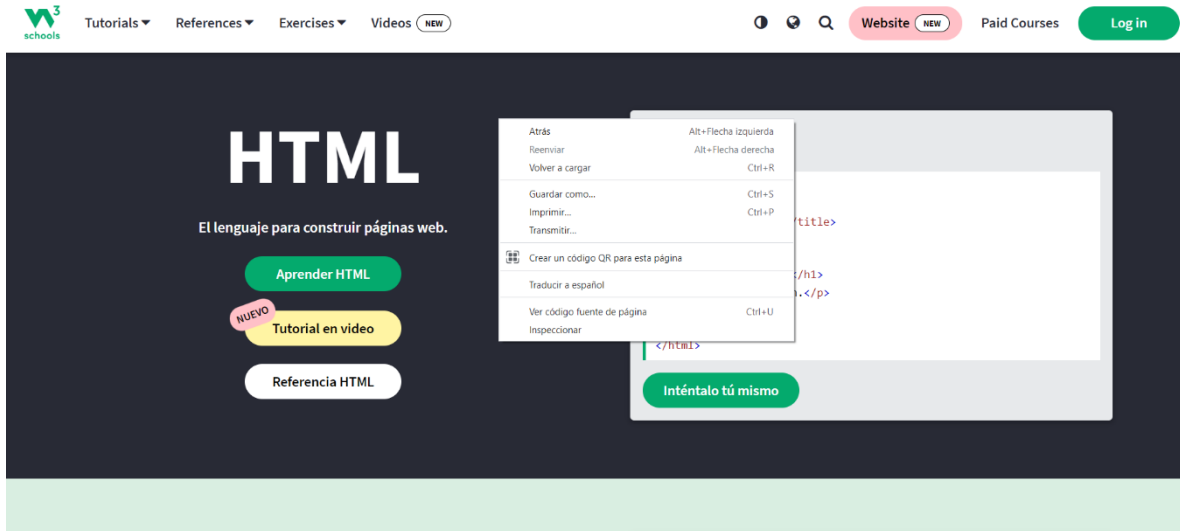
El inspector de elementos es una herramienta que nos permite analizar los contenidos de un sitio web. Utilizando esta herramienta podemos acceder a ver el código fuente de la página web que estemos visitando, así como analizar diferentes elementos que la conforman.

Es bastante útil para conocer como está hecha una determinada página web o para analizar nuestros propios sitios y modificar estilos de manera transitoria para ir evaluando como podemos alcanzar aquello que deseamos.

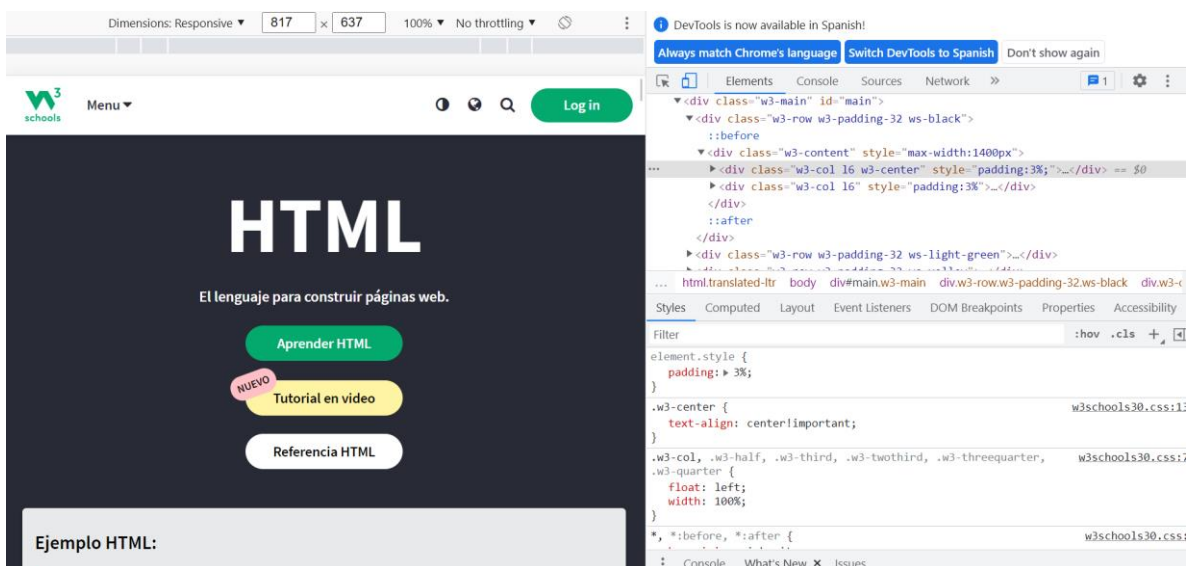
Todos los navegadores incluyen un inspector de elementos y en todos se pueden realizar las mismas acciones.

ABRIR EL INSPECTOR DE ELEMENTOS

Lo primero que debemos hacer es acceder al sitio web que deseamos analizar. En nuestro caso accederemos al sitio web de la **w3school** y en él haremos clic en el botón secundario del mouse y seleccionaremos "inspeccionar".

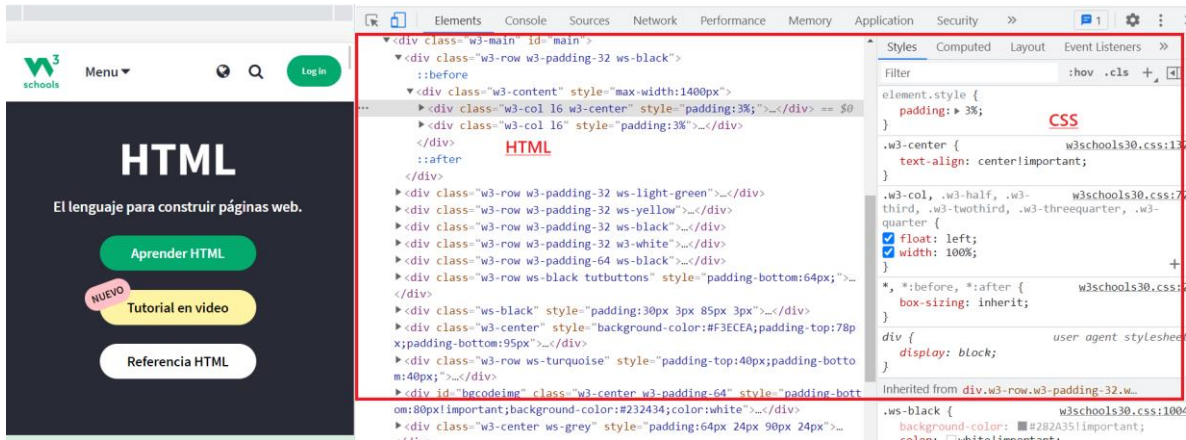


automáticamente se nos desplegará otra pestaña o aparecerá a un costado o debajo de la página que estamos analizando:



Este inspector de elementos, como previamente dijimos, permite visualizar el código fuente de la página web que estamos visitando y ver como se encuentra construida.

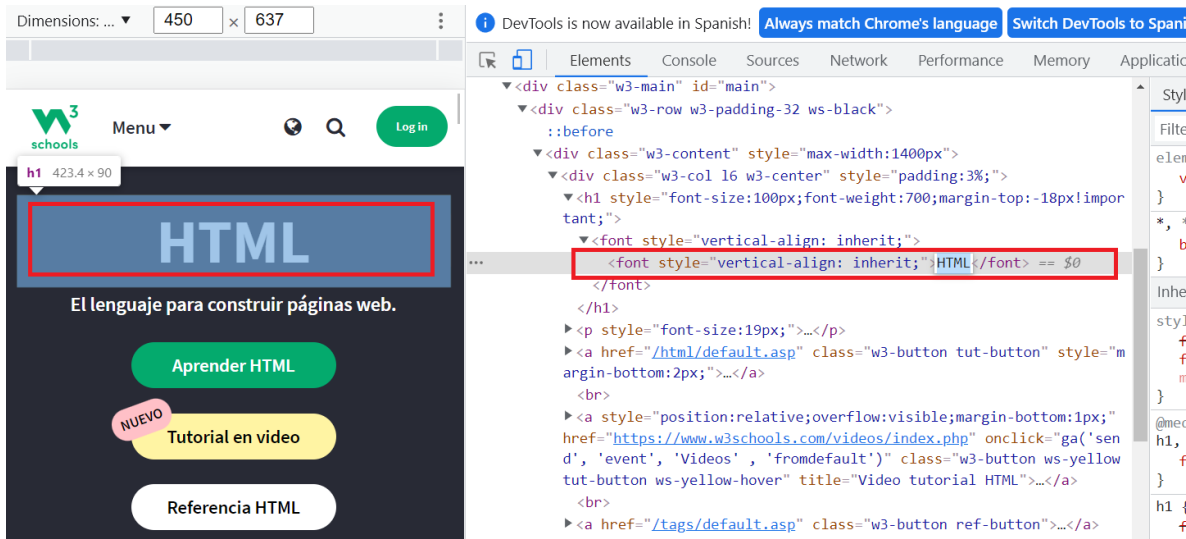
Podremos ver el **HTML** y el **CSS** de la misma.



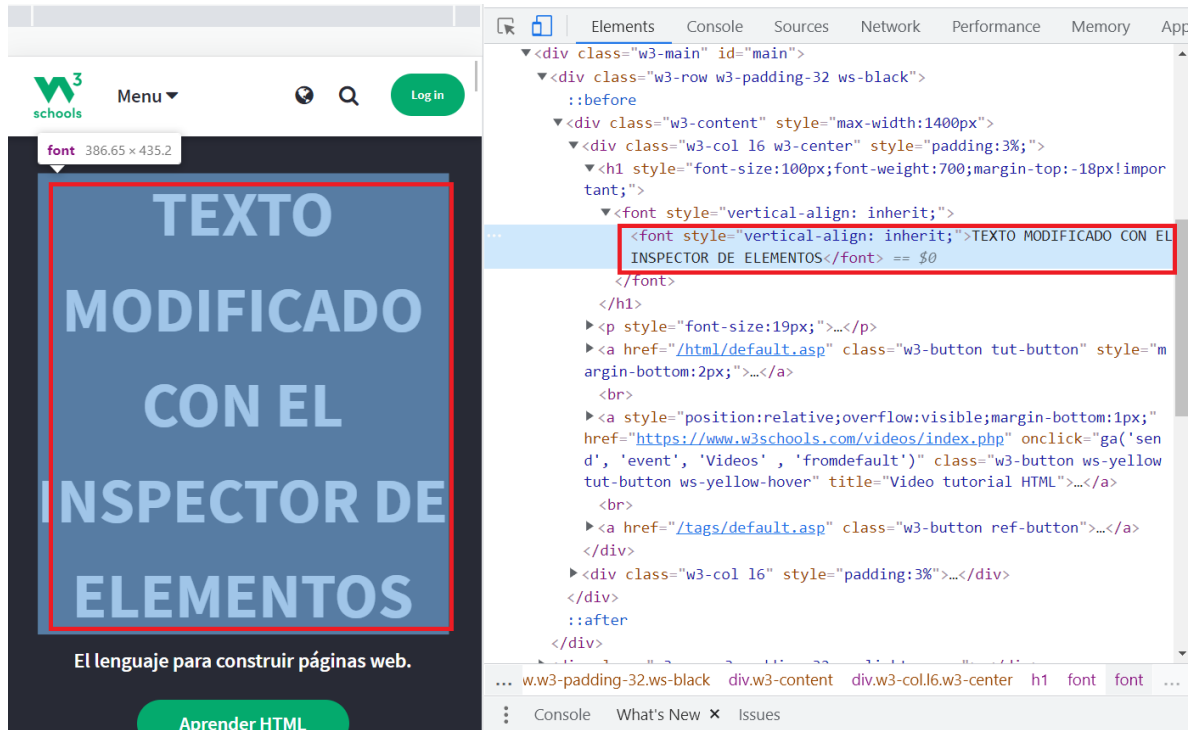
Además de eso, nos permite hacer modificaciones a una página web que serán transitorios, esto quiere decir que esas modificaciones existirán solamente mientras analizamos la página con el inspector de elementos y no se verán reflejados para otras personas e, incluso, al refrescar la página web se perderán los cambios.

Generalmente usaremos esta posibilidad para analizar sitios web propios, en los cuales queramos hacer un cambio y una vez que lo realicemos de manera transitoria, si nos gusta, lo llevaremos al código fuente para realizarlo de manera definitiva.

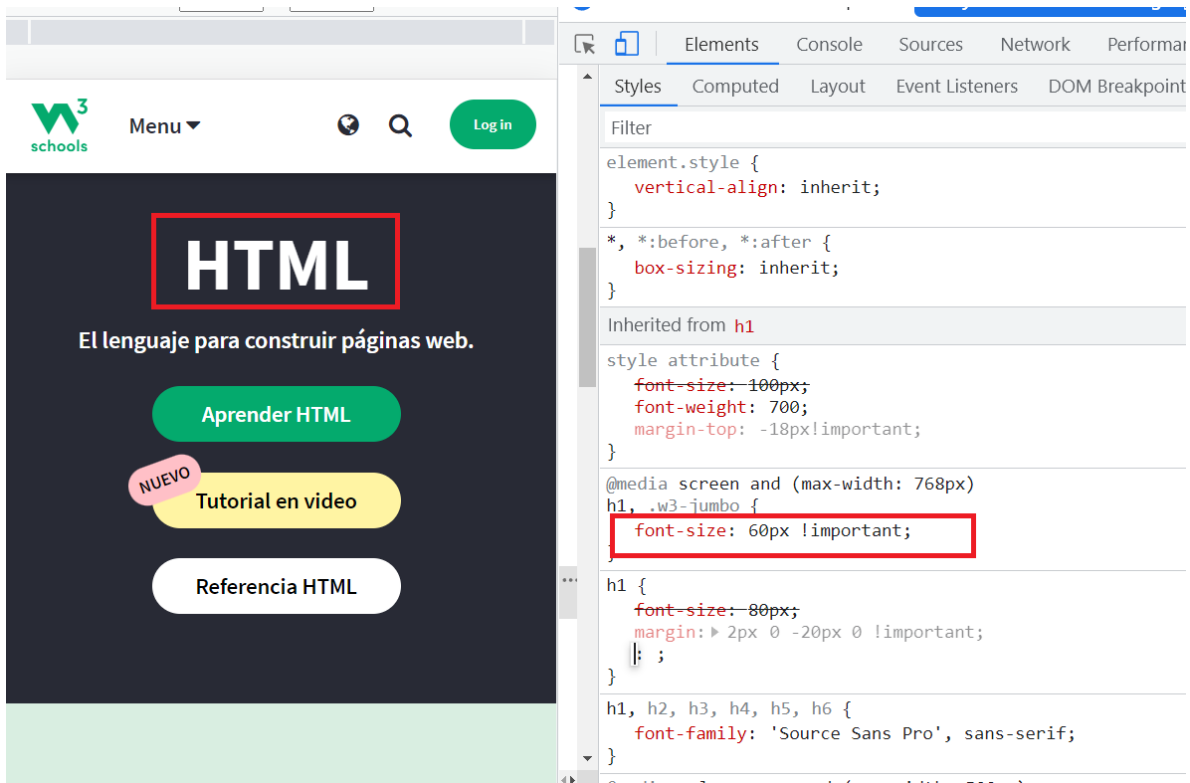
¿Cómo realizamos esto? Simplemente modificamos manualmente un elemento.



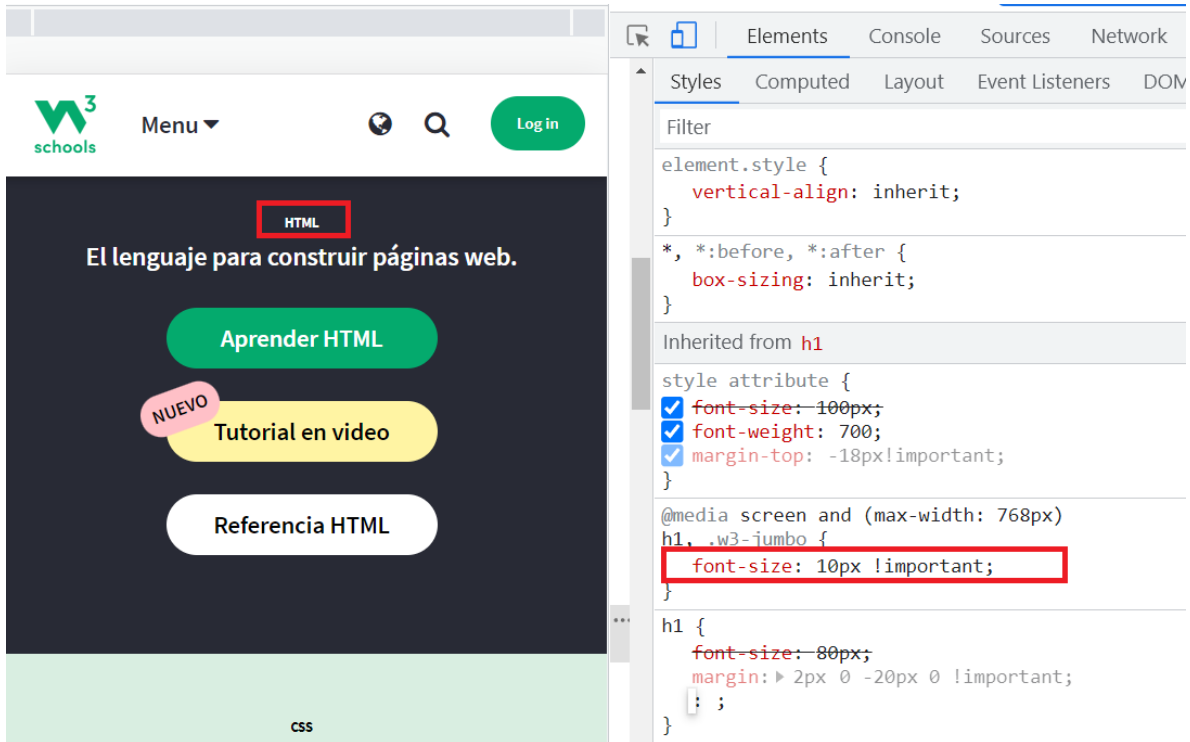
Modificaremos el texto "HTML" y escribiremos en su lugar algún texto de nuestra preferencia. Debemos hacer **Enter** para concretar el cambio.



También podemos modificar los estilos en el espacio de **CSS**.



Y estos cambios también se verán reflejados de manera transitoria.



De esta forma, podemos modificar y dar estilo a nuestras páginas web, “jugando” con el inspector de elementos y comprendiendo el modelo de cajas para poder dar una interfaz grafica agradable de usar a todos nuestros proyectos.