



Lección 3

Introducción a CSS

En la lección anterior, hemos aprendido a crear un documento HTML, a organizar su estructura, y a determinar qué elementos son más apropiados para representar su contenido. Esta información nos permite definir el documento, pero no determina cómo se mostrará en pantalla. Desde la introducción de HTML5, esa tarea es responsabilidad de CSS.

CSS es un lenguaje que facilita instrucciones que podemos usar para asignar estilos a los elementos HTML, como colores, tipos de letra, tamaños, etc. Los estilos se deben definir con CSS y luego asignar a los elementos hasta que logramos el diseño visual que queremos para nuestra página.

Objetivos de la lección

Al finalizar esta lección, los participantes podrán:

- Conocer las distintas formas de aplicar estilos CSS
- Definir selectores CSS
- Conocer el modelo de caja
- Crear una página responsiva.

CSS básico

CSS (Hojas de Estilo en Cascada) es el código que usas para dar estilo a tu página web. CSS Básico te lleva a través de lo que tú necesitas para empezar. Contestará a preguntas del tipo: ¿Cómo hago mi texto rojo o negro? ¿Cómo hago que mi contenido se muestre en tal y tal lugar de la pantalla? ¿Cómo decoro mi página web con imágenes de fondo y colores?

Entonces ¿qué es CSS, realmente?

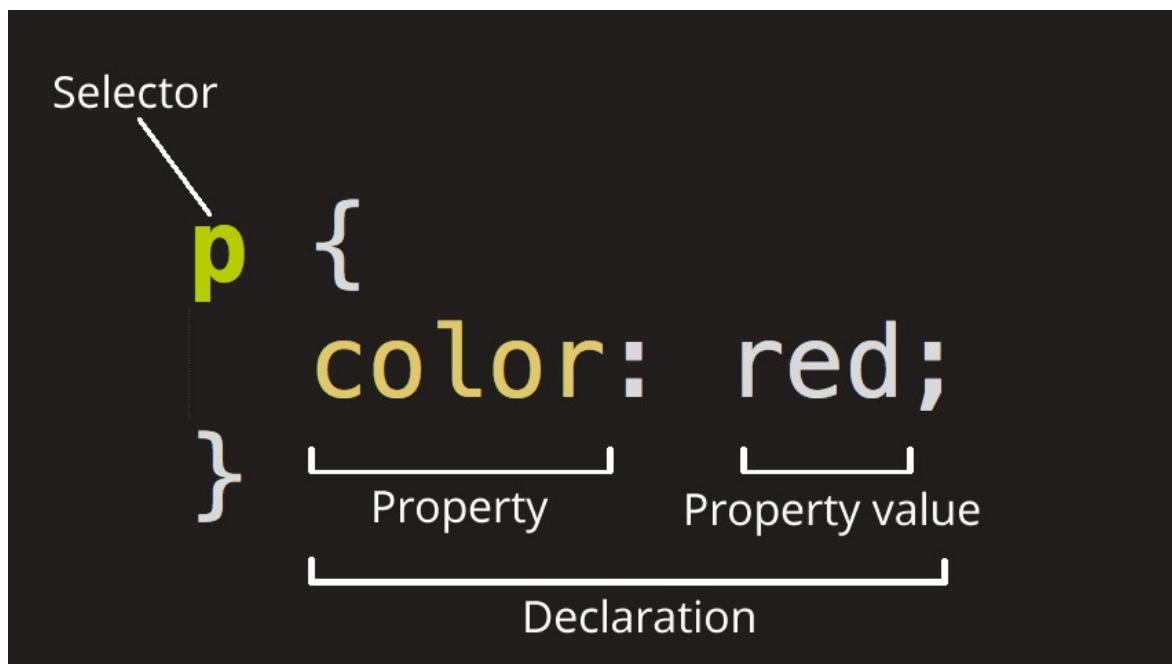
Como HTML, CSS (Cascading Style Sheets) u Hojas de estilo en cascada en español, no es realmente un lenguaje de programación, tampoco es un lenguaje de marcado. Es un lenguaje de hojas de estilo, es decir, te permite aplicar estilos de manera **selectiva** a elementos en documentos HTML. Por ejemplo, para seleccionar todos los elementos de párrafo en una página HTML y volver el texto dentro de ellos de color rojo, podemos escribir este CSS:

```
p{  
  color: red;  
}
```

A este código también es conocido como reglas CSS.

Anatomía de una regla CSS

Observa el código CSS de arriba, un poco más a detalle:



Partes de una declaracion de css

La estructura completa es llamada regla predeterminada (pero a menudo «regla» para abreviar).

Nota también los nombres de las partes individuales:

- **Selector:** El elemento HTML en el que comienza la regla. Esta selecciona el(los) elemento(s) a dar estilo (en este caso, los elementos <p>). Para dar estilo a un elemento diferente, solo cambia el selector.
- **Declaración:** Una sola regla como color: red; especifica a cuál de las propiedades del elemento quieres dar estilo.
- **Propiedad:** Maneras en las cuales puedes dar estilo a un elemento HTML. (En este caso, color es una propiedad del elemento <p>). En CSS, seleccionas qué propiedad quieres afectar en tu regla.
- **Valor:** A la derecha de la propiedad, después de los dos puntos (:), tienes el valor de la propiedad, para elegir una de las muchas posibles apariencias para una propiedad determinada (hay muchos valores para color además de red).

Nota las otras partes importantes de la sintaxis:

Cada una de las reglas (aparte del selector) deben estar encapsuladas entre llaves ({}).

Dentro de cada declaración, debes usar los dos puntos (:) para separar la propiedad de su valor.

Dentro de cada regla, debes usar el punto y coma (;) para separar una declaración de la siguiente.

De este modo para modificar varios valores de propiedad a la vez, solo necesitas escribirlos separados por punto y coma (;), así:

```
p{  
  color: red;  
  border: 1px;  
  width: 300;  
}
```

Diferentes tipos de selectores

Existen muchos tipos diferentes de selectores. Antes, solo viste los selectores de elementos, los cuales seleccionan todos los elementos de un tipo dado en los documentos HTML. Sin embargo, puedes hacer selecciones más específicas que esas. En seguida están algunos de los tipos de selectores más comunes:

Nombre del selector	Qué selecciona	Ejemplo
Selector de elemento (llamado algunas veces selector de etiqueta o tipo)	Todos los elementos HTML del tipo especificado.	p Selecciona <p>
Selector de identificación (ID)	El elemento en la página con el ID especificado (en una página HTML dada, solo se permite un único elemento por ID).	#mi-id Selecciona <p id="mi-id"> y
Selector de clase	Los elementos en la página con la clase especificada (una clase puede aparecer varias veces en una página).	.mi-clase Selecciona <p class="mi-clase"> y
Selector de atributo	Los elementos en una página con el atributo especificado.	img[src] Selecciona pero no
Selector de pseudoclase	Los elementos especificados, pero solo cuando esté en el estado especificado, por ejemplo cuando el puntero esté sobre él.	a:hover Selecciona <a>, pero solo cuando el puntero esté sobre el enlace.

Existen muchos más selectores para explorar, y podrás encontrar una lista más detallada en la [guía de Selectores](#).

Pseudoclases y pseudoelementos

El conjunto de selectores que estudiaremos en este artículo, se conocen como **pseudoclases** y **pseudoelementos**. Hay muchos y a menudo sirven para fines muy específicos.

¿Qué es una *pseudoclase*?

Una **pseudoclase** es un selector que marca los elementos que están en un **estado** específico, por ejemplo, aquellos por los que el cursor les pasa por encima. Tienden a actuar como si hubieras aplicado una clase en una parte determinada del documento y, a menudo, ayudan a reducir el exceso de clases y proporcionan un marcado más flexible y fácil de mantener.

Las **pseudoclases** son palabras clave que comienzan con dos puntos. Todas las **pseudoclases** se comportan del mismo modo. Seleccionan un fragmento del documento que está en un estado determinado y se comportan como si se hubiera añadido una clase a su HTML.

Ejemplos:

- `:last-child`
- `:only-child`
- `:invalid`

Algunas pseudoclases solo intervienen cuando el usuario interactúa con el documento de alguna manera. Estas pseudoclases de acción de usuario, que también reciben el nombre de pseudoclases dinámicas, actúan como si se añadiese una clase al elemento cuando el usuario interactúa con él. Algunos ejemplos son:

- **:hover**: solo interviene si el usuario pasa el cursor sobre un elemento, normalmente un enlace.
- **:focus**: solo interviene si el usuario selecciona el elemento con los controles del teclado.

¿Qué es un *pseudoelemento*?

Los **pseudoelementos** se comportan de manera similar. Sin embargo, actúan como si hubieras añadido un elemento HTML totalmente nuevo en el marcado, en lugar de haber aplicado una clase nueva a los elementos presentes. Los *pseudoelementos* empiezan con un doble signo de dos puntos: `::`.

Hay un par de *pseudoelementos* especiales que se utilizan junto con la propiedad de **content** para introducir contenido en el documento usando el CSS.

Puedes utilizarlos para insertar una cadena de texto. Y con la propiedad *content* puedes cambiar el contenido ya sea utilizando el pseudoelemento **::before** o **::after** y verás que el contenido se inserta al principio o al final del elemento.

Las pseudoclases

Selector	Descripción
:active	Selecciona un elemento cuando el usuario lo activa (por ejemplo, con un clic).
:any-link	Selecciona los estados :link y :visited de un enlace.
:blank	Selecciona un elemento <input> cuyo valor de entrada está vacío.
:checked	Selecciona un botón de opción o casilla de verificación en el estado que determines.
:current	Selecciona el elemento que se muestra en ese momento, o un ancestro de ese elemento.
:default	Selecciona uno o más elementos de interfaz de usuario cuyo valor es el predeterminado de entre un conjunto de elementos similares.
:dir	Selecciona un elemento según su direccionalidad (valor del atributo dir de HTML o propiedad direction de CSS).
:disabled	Selecciona elementos de la interfaz de usuario que están en estado inactivo.
:empty	Selecciona un elemento que no tiene elementos hijo, excepto por algún espacio en blanco opcional.
:enabled	Selecciona elementos de la interfaz de usuario que están en estado activo.
:first	En Paged Media, selecciona la primera página.
:first-child	Selecciona el primero entre elementos hermanos.
:first-of-type	Selecciona el primero entre un tipo determinado de elementos hermanos.
:focus	Selecciona el elemento que tiene el foco.
:focus-visible	Selecciona el elemento que tiene el foco cuando el foco tiene que estar visible para el usuario.
:focus-within	Selecciona el elemento que tiene el foco y el elemento con un descendiente que tiene el foco.
:future	Selecciona los elementos que van después del elemento en curso.
:hover	Selecciona un elemento cuando el usuario interactúa con él.
:indeterminate	Selecciona elementos de interfaz de usuario cuyo valor está en un estado no determinado, por lo general se trata de casillas de verificación.
:in-range	Selecciona un elemento cuyo valor se encuentra dentro de un rango de valores determinado.
:invalid	Selecciona un elemento, como por ejemplo un <input>, cuyo estado es no válido.
:lang	Selecciona un elemento según el idioma (valor del atributo lang de HTML).
:last-child	Selecciona el último elemento de entre sus elementos hermanos.
:last-of-type	Selecciona el último de entre los elementos hermanos de un tipo determinado.
:left	En Paged Media selecciona las páginas de la izquierda.
:link	Selecciona los enlaces no visitados.
:local-link	Selecciona los enlaces que dirigen a páginas que se encuentran en la misma página web que el documento activo.

:is()	Selecciona cualquiera de los selectores de la lista de selección que se pase como valor de este selector.
:not	Selecciona elementos que otros selectores no han seleccionado antes y que se han pasado como valor de este selector.
:nth-child	Selecciona elementos de entre una lista de elementos hermanos. Los elementos hermanos están relacionados por una fórmula del tipo $an + b$ (por ejemplo, $2n + 1$ seleccionaría los elementos 1, 3, 5, 7, etc., es decir, todos los impares).
:nth-of-type	Selecciona elementos de entre una lista de elementos hermanos de un tipo determinado (por ejemplo, todos los elementos <code><p></code>). Los elementos hermanos están relacionados por una fórmula del tipo $an + b$ (por ejemplo, $2n + 1$ relacionaría en la secuencia ese tipo de elementos, los números 1, 3, 5, 7, etc., es decir, todos los impares).
:nth-last-child	Selecciona elementos de entre una lista de elementos hermanos, contando hacia atrás desde el final. Los elementos hermanos están relacionados por una fórmula del tipo $an+b$ (por ejemplo, $2n + 1$ relacionaría en la secuencia el último de los elementos de este tipo con el que se encuentra dos por delante, y así sucesivamente. Todos los impares, contando desde el final).
:nth-last-of-type	Selecciona los elementos de entre una lista de elementos hermanos que son de un tipo determinado (por ejemplo, elementos <code><p></code>), contando hacia atrás desde el final. Los elementos hermanos están relacionados por una fórmula del tipo $an+b$ (por ejemplo, $2n + 1$ relacionaría en la secuencia el último de los elementos de ese tipo con el que se encuentra dos por delante, y así sucesivamente. Todos los impares, contando desde el final).
:only-child	Selecciona un elemento que no tiene elementos hermanos.
:only-of-type	Selecciona un elemento que es el único de su tipo entre sus elementos hermanos.
:optional	Selecciona los elementos de formulario que son innecesarios.
:out-of-range	Selecciona un elemento cuyo valor está fuera de rango.
:past	Selecciona los elementos que se encuentran antes del elemento activo.
:placeholder-shown	Selecciona el elemento de entrada que muestra texto de marcador de posición.
:playing	Selecciona un elemento que representa un audio, un vídeo o un recurso similar que se puede «reproducir» o «pausar», cuando el elemento está «en reproducción».
:paused	Selecciona un elemento que representa un audio, un vídeo o un recurso similar que se puede «reproducir» o «pausar» cuando el elemento está «pausado».
:read-only	Selecciona los elementos que el usuario no puede modificar.
:read-write	Selecciona los elementos que el usuario puede modificar.
:required	Selecciona los elementos de formulario que son necesarios.
:right	En Paged Media selecciona las páginas de la derecha.
:root	Selecciona un elemento que es la raíz del documento.
:scope	Selecciona cualquier elemento de ámbito.
:valid	Selecciona un elemento como <code><input></code> , en un estado válido.
:target	Selecciona el elemento al que apunta la URL activa (es decir, cuyo ID coincide con el identificador de fragmento de la URL activo).

:visited	Selecciona los enlaces visitados.
-----------------	-----------------------------------

Pseudoelementos

Selector	Descripción
::after	Selecciona el elemento al que se puede aplicar estilo que aparece a continuación del contenido del elemento que lo origina.
::before	Selecciona el elemento al que se puede aplicar estilo que aparece antes del contenido del elemento que lo origina.
::first-letter	Selecciona la primera letra del elemento.
::first-line	Selecciona la primera línea del elemento de contenido.
::grammar-error	Selecciona una parte del documento que contiene un error de gramática indicado por el navegador.
::selection	Selecciona la parte del documento que ha sido seleccionada.
::spelling-error	Selecciona una parte del documento que contiene un error de ortografía indicado por el navegador.

Modelo de caja

Todo en CSS tiene una “caja” alrededor, y comprender estas cajas es clave para poder crear diseños con CSS o para alinear elementos con otros elementos.

Cajas en bloque y en línea

En CSS, en general, hay dos tipos de cajas: cajas en **bloque (block)** y cajas **en línea (inline)**. Estas características se refieren al modo cómo se comporta la caja en términos de flujo de página y en relación con otras cajas de la página:

Si una caja se define como **block**, se comportará de las maneras siguientes:

- La caja fuerza un salto de línea al llegar al final de la línea.
- La caja se extenderá todo el ancho para llenar todo el espacio disponible que haya en su contenedor. En la mayoría de los casos, esto significa que la caja será tan ancha como su contenedor, y llenará el 100% del espacio disponible.
- Se respetan las propiedades **width** y **height**.
- El relleno, el margen y el borde mantienen a los otros elementos alejados de la caja.

Si una caja tiene una visualización externa de tipo **inline**, entonces:

- La caja no fuerza ningún salto de línea al llegar al final de la línea.
- Las propiedades **width** y **height** no se aplican.

- Se aplican relleno, margen y bordes verticales, pero no mantienen alejadas otras cajas en línea.
- Se aplican relleno, margen y bordes horizontales, y mantienen alejadas otras cajas en línea.

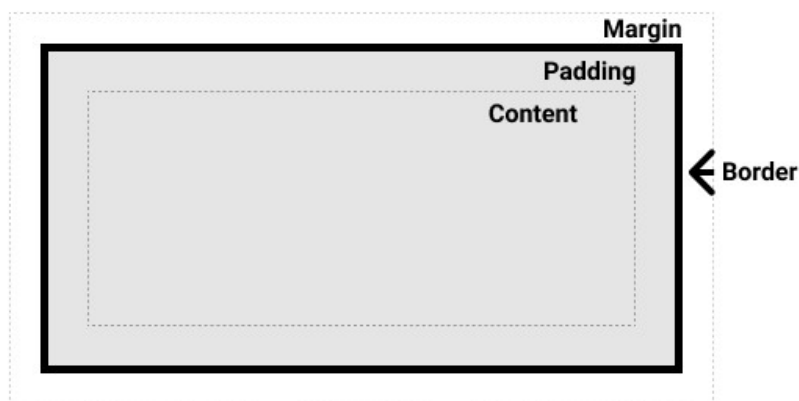
¿Qué es el modelo de cajas CSS?

El modelo de cajas CSS completo se aplica a cajas que presentan comportamiento en bloque; las cajas con comportamiento en línea solo usan una parte del comportamiento definido en el modelo de cajas. El modelo define cómo funcionan juntas las diferentes partes de una caja (margen, borde, relleno y contenido) para crear una caja que puedas ver en tu página. Para complicarlo un poco más, hay un modelo de cajas estándar y un modelo de cajas alternativo.

Partes de una caja

Al hacer una caja de tipo **block** en CSS tenemos los elementos siguientes:

- El contenido de la caja (o content box): El área donde se muestra el contenido, cuyo tamaño puede cambiarse utilizando propiedades como width y height.
- El relleno de la caja (o padding box): El relleno es espacio en blanco alrededor del contenido; es posible controlar su tamaño usando la propiedad padding y otras propiedades relacionadas.
- El borde de la caja (o border box): El borde de la caja envuelve el contenido y el de relleno. Es posible controlar su tamaño y estilo utilizando la propiedad border y otras propiedades relacionadas.
- El margen de la caja (o margin box): El margen es la capa más externa. Envuelve el contenido, el relleno y el borde como espacio en blanco entre la caja y otros elementos. Es posible controlar su tamaño usando la propiedad margin y otras propiedades relacionadas.



Diseño responsivo

El término diseño responsivo fue acuñado por *Ethan Marcotte en 2010*, y describía el uso combinado de tres técnicas.

La primera es la idea de las cajas fluidas, algo que ya exploraba Gillenwater, y que puede leerse en el artículo de Marcotte, Fluid Grids (publicado en 2009 en A list apart).

La segunda técnica es la idea de las imágenes fluidas. Usando una técnica muy simple de establecer la propiedad de `max-width` al 100%, las imágenes se reducían si su columna de contención se volvía más estrecha que el tamaño intrínseco de la imagen, pero nunca se expandía. Esto permite reducir una imagen para que se ajuste a una columna de tamaño flexible, en lugar de que desborde, pero no se expande ni se pixela si la columna se ensancha más que la imagen.

El tercer componente clave es el uso de **media query**. Las *media query* permiten cambiar el tipo de cambio de diseño que Cameron Adams había explorado previamente usando JavaScript, pero usando solo CSS. En lugar de tener un único diseño para todos los tamaños de pantalla, el diseño podría cambiarse. Las barras laterales pueden reposicionarse en una pantalla más pequeña, o puede mostrarse una navegación alternativa.

Es importante comprender que el diseño web responsivo no es una tecnología independiente: es un término utilizado para describir un enfoque para el diseño web, o un conjunto de buenas prácticas utilizado para crear un diseño que puede responder según el dispositivo que se utiliza para ver un contenido.

Media query

El diseño responsivo solo pudo surgir gracias a la definición de *media query*. La especificación de creación de *media queries* se convirtió en una candidata a Recomendación en 2009, lo que significa que se consideró lista para su implementación en los navegadores. Una *media query* nos permite ejecutar una serie de pruebas (por ejemplo, si la pantalla del usuario es mayor que un ancho o una resolución determinados) y aplicar CSS selectivamente para diseñar la página de manera que resulte adecuada a las necesidades del usuario.

Puedes añadir múltiples *media queries* dentro de una hoja de estilo, y ajustar todo tu diseño o solo partes de él para que se adapte mejor a los diferentes tamaños de pantalla. Los puntos en los que se introduce una **media query** y se cambia el diseño se conocen como puntos de interrupción o **break points**.

Un enfoque común cuando se usan *media queries* es crear un diseño sencillo de una sola columna para dispositivos de pantalla estrecha (por ejemplo, teléfonos móviles), luego implementar un diseño

en columnas para pantallas más grandes cuando se sabe que hay suficiente ancho de pantalla para manejarlo. Esto se describe a menudo como diseño **mobile-first**.

El siguiente código muestra la definición de una media query,

```
@media(max-width:1400px){  
  .main-content{  
    max-width: 1100px;  
  }  
  
  .resume{  
    width: 70%;  
  }  
}
```

Una media query, consiste en:

- Palabra reservada **@media**
- Tipo de medio o **media type**: el cual le avisa al navegador el tipo de medio al que tiene que aplicar el conjunto de reglas descrito. Los media types son: **all**, **print** y **screen**
- **Expresión**: la cual es la regla, que debe ser consultada para que se aplique el conjunto de CSS
- **CSS**: conjunto de reglas CSS que se van a aplicar en caso de que la expresión sea verdadera.

Referencias:

Modelo de caja	<ul style="list-style-type: none">• El modelo de caja - Aprende desarrollo web MDN (mozilla.org)• The CSS Box Model CSS-Tricks
Flexbox	<ul style="list-style-type: none">• Flexbox - Aprende desarrollo web MDN (mozilla.org)• A Complete Guide To Flexbox CSS-Tricks• Flexbox Froggy - Un juego para aprender CSS flexbox
Grid	<ul style="list-style-type: none">• CSS Grid Layout - CSS MDN (mozilla.org)• Layoutit Grid: Learning CSS Grid Visually With A Generator CSS-Tricks• Grid Garden - Un juego para aprender CSS grid (cssgridgarden.com)