

3.3. Modelo de caja y posicionamiento



LENGUAJES DE PROGRAMACIÓN NIVEL 3. UNIDAD 3

CSS - ESTILOS PARA LA PRESENTACIÓN VISUAL DE LA PÁGINA WEB

Contenido

Introducción.....	3
El modelo de caja de CSS.....	5
Conceptos básicos.....	5
Elementos de una caja	6
Ejemplo	6
Box-Sizing	7
Posicionamiento en CSS	9
Tipos de posicionamiento.....	9
static:.....	9
relative:	10
absolute:	10
fixed:.....	10
sticky:	11
Layouts con modelo de caja y posicionamiento	12
Creación de un layout	12
Responsive design	13
Relación con el modelo de caja y el posicionamiento.....	13
Conclusión	15
Recursos adicionales.....	16
Enlaces a documentación oficial y tutoriales.....	16
Herramientas en línea para practicar CSS	16

Introducción

Te doy la bienvenida al módulo sobre el Modelo de Caja y Posicionamiento en CSS, un componente esencial en el diseño y la maquetación de páginas web. Este módulo se adentra en dos conceptos fundamentales que son piedras angulares del diseño web: el Modelo de Caja de CSS, que define cómo se representan los elementos en la página, y las técnicas de posicionamiento, que determinan dónde se ubican esos elementos en el diseño general.

El modelo de caja de CSS es un concepto crucial que todo desarrollador web debe entender profundamente. En esencia, este modelo trata a cada elemento HTML como si estuviera contenido dentro de una caja rectangular. Esta "caja" incluye márgenes, bordes, relleno y el contenido real, cada uno afectando el espacio que el elemento ocupa en la página. Se trata de un modelo dinámico en la que el posicionamiento de cada elemento afecta al resto de elementos incluidos dentro de esa caja. Comprender cómo manipular estos aspectos es vital para controlar el layout y la estética de tus sitios web.

Por otro lado, el posicionamiento en CSS es igualmente importante, ya que te permite controlar la ubicación exacta de los elementos en la página. Desde el flujo normal del documento hasta técnicas más avanzadas como el posicionamiento absoluto y fijo, este módulo te mostrará la manera en que puedes utilizar el posicionamiento para crear diseños dinámicos y responsivos.

Dicho lo anterior, y al final de este módulo, estarás equipado con un conocimiento detallado sobre:

- *El modelo de caja de CSS: Entenderás cómo cada elemento se representa como una caja y cómo las propiedades de margen, borde, relleno y contenido influyen en la disposición de los elementos.*
- *Técnicas de posicionamiento: Aprenderás a utilizar las propiedades de posicionamiento de CSS para colocar elementos en la página de manera efectiva, permitiéndote crear diseños complejos y reactivos.*

Este módulo te proporcionará las herramientas y técnicas necesarias para manipular el espacio y la posición de los elementos en tus páginas web, permitiéndote crear diseños atractivos y funcionales que mejoren la experiencia del usuario.

El modelo de caja de CSS



El modelo de caja es un concepto fundamental en el diseño web y es esencial para entender cómo CSS maneja el diseño y la disposición de los elementos en una página web. Este modelo describe cómo cada elemento de página se representa como una caja rectangular, y cómo estas cajas interactúan entre sí y con el contenedor de la página.

Conceptos básicos

Se puede concebir una página web como un conjunto de cajas anidadas dentro de otras cajas, donde cada elemento, desde el más grande como un `<div>` contenedor hasta el más pequeño como un `` de texto, se representa como una caja individual en el modelo.

Por ejemplo, imagina un simple artículo de blog compuesto por un título (`<h1>`), un subtítulo (`<h2>`), y varios párrafos de texto (`<p>`). En el modelo de caja, el título, el subtítulo y cada párrafo se consideran cajas individuales, cada una con su propio contenido, relleno, borde y margen, que pueden ser estilizados y posicionados de manera independiente en la página.

Esta estructura permite a los diseñadores web un control preciso sobre la disposición y el estilo de cada elemento, facilitando la creación de diseños complejos y responsivos.

Elementos de una caja

Cada "caja" en el modelo de caja de CSS consta de cuatro componentes principales:

- **Contenido (Content):** El área donde se muestra el contenido real del elemento, como texto o imágenes. Las propiedades width y height controlan el tamaño de esta área.
- **Relleno (Padding):** El espacio entre el contenido del elemento y su borde. Aumenta el área total del elemento, pero no el tamaño del contenido en sí. Se puede controlar individualmente con padding-top, padding-right, padding-bottom, y padding-left.
- **Borde (Border):** Encierra el contenido y el relleno. Puede ser visible o invisible y se controla con propiedades como border-style, border-width, y border-color.
- **Margen (Margin):** El espacio alrededor de la caja, separándola de otros elementos. Al igual que el relleno, puede controlarse individualmente para cada lado del elemento.

La suma de estas áreas (contenido, relleno, borde y margen) determina el tamaño total y la posición de un elemento en la página.

Ejemplo

Imagina que estás diseñando una tarjeta de perfil para un sitio web, donde cada tarjeta es un elemento <div> que contiene una foto, un nombre y una breve descripción. En términos del Modelo de Caja:

- **Contenido:** Sería la foto, el nombre y la descripción dentro de la tarjeta.
- **Relleno:** El espacio entre los bordes de la foto, el nombre y la descripción y los límites de la tarjeta. Añadir padding alrededor del contenido garantiza que no esté pegado directamente a los bordes de la tarjeta, haciendo que el diseño sea más atractivo visualmente.
- **Borde:** Si decides que cada tarjeta debería tener un borde para distinguirla claramente del fondo de la página o de otras tarjetas, este sería el borde alrededor del contenido y el relleno. Podrías usar border-style, border-width, y border-color para definir su apariencia.
- **Margen:** El espacio alrededor de la tarjeta que la separa de otras tarjetas o elementos en la página. Usar margin asegura que las tarjetas no estén demasiado apretadas unas contra otras, mejorando la legibilidad y la estética general del diseño.

Box-Sizing

Tras comprender los componentes fundamentales del Modelo de Caja — contenido, relleno, borde y margen—, surge naturalmente la pregunta de cómo estos elementos interactúan para formar el tamaño total de una caja. Aquí es donde el concepto de box-sizing se vuelve crucial.

Por defecto, el ancho y la altura de una caja se aplican solo al área de contenido, lo que significa que el relleno y el borde se añaden al tamaño final del elemento, a menudo llevando a confusiones y complicaciones en el diseño, especialmente cuando se intenta que los elementos encajen en dimensiones específicas. ¿Esto qué quiere decir?

Por defecto, CSS utiliza el valor content-box para la propiedad box-sizing, lo que implica que cuando especificas el width (ancho) y el height (altura) de un elemento, estas medidas se aplican únicamente al área del contenido de la caja. Esto significa que el relleno (padding) y el borde (border) que añades a este elemento no se incluyen dentro de las dimensiones especificadas para el ancho y la altura; en cambio, se suman a estas automáticamente, aumentando el tamaño total del elemento en la página.

Por ejemplo, si tienes un elemento con un width de 100px y un height de 100px, y le añades un padding de 10px por todos lados y un border de 2px, el tamaño total del elemento no será de 100x100px. En cambio, debido al relleno y al borde adicionales, el tamaño final será de 124px de ancho (100px de contenido + 10px de relleno a la izquierda + 10px de relleno a la derecha + 2px de borde a la izquierda + 2px de borde a la derecha) y 124px de alto, siguiendo la misma lógica.

Este comportamiento puede ser no intuitivo, especialmente para quienes están aprendiendo CSS o para aquellos casos en los que se requiere un control preciso del layout, ya que el tamaño final de los elementos puede terminar siendo mayor de lo esperado, afectando la disposición de estos en la página. Por ello, es esencial comprender cómo el valor predeterminado de box-sizing influye en el cálculo de las dimensiones de los elementos y cómo puedes modificar este comportamiento utilizando la propiedad box-sizing para ajustarse mejor a tus necesidades de diseño.

La propiedad `box-sizing` ofrece una solución elegante a este dilema, permitiéndonos controlar cómo se calculan las dimensiones de las cajas y hacer que el proceso de diseño sea más intuitivo y predecible.

Por tanto, la propiedad `box-sizing` en CSS es crucial para controlar cómo se calculan las dimensiones de un elemento. Hay dos valores principales:

- `content-box` (valor por defecto): El ancho y la altura del elemento se aplican solo al contenido, y el relleno y el borde se añaden al tamaño total del elemento.
- `border-box`: El ancho y la altura del elemento incluyen el contenido, el relleno y el borde, lo que facilita el diseño al hacer que los elementos se comporten de manera más predecible.

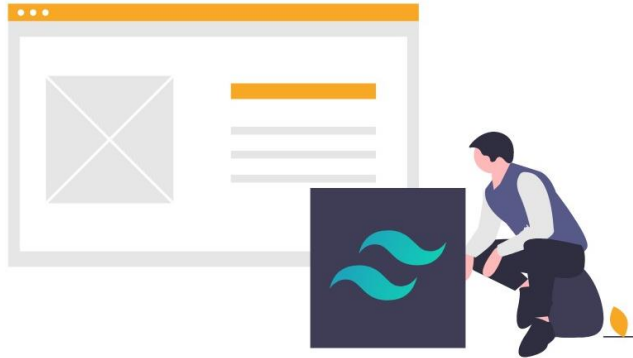
```
.box {  
  box-sizing: border-box;  
}
```

1

¹ Accesibilidad:

```
.box {  
  box-sizing: border-box;  
}
```


Posicionamiento en CSS



El posicionamiento en CSS es una herramienta poderosa que afecta cómo y dónde se muestran los elementos en la página web. Comprender los diferentes tipos de posicionamiento y cómo interactúan entre sí es crucial para crear diseños precisos y dinámicos.

Este módulo explora los fundamentos del posicionamiento en CSS, la importancia del z-index en el apilamiento de elementos y proporciona prácticas recomendadas para abordar desafíos comunes de diseño.

Tipos de posicionamiento

CSS ofrece varios tipos de posicionamiento, cada uno con su propósito y comportamiento específicos:

static:

Es el valor predeterminado. Los elementos se colocan en el flujo normal del documento, uno tras otro, según aparecen en el HTML.

- Uso típico: No necesitas definirlo porque todos los elementos HTML lo tienen por defecto. Es ideal cuando quieres que los elementos sigan el flujo normal de la página, uno tras otro.
- Ejemplo: Un párrafo de texto seguido por una imagen, tal como aparecen en tu código HTML.

relative:

Posiciona un elemento en relación con su posición original en el flujo del documento. Puedes mover un elemento con top, right, bottom, y left, pero el espacio original que ocupaba el elemento permanece.

- Uso típico: Perfecto para hacer pequeños ajustes de posición a un elemento sin afectar el layout general.
- Ejemplo: Mover ligeramente un botón hacia abajo para alinearlo mejor con otros elementos de su alrededor.

absolute:

Extrae el elemento del flujo normal del documento y lo posiciona en relación con su contenedor posicionado más cercano. Si no hay tal contenedor, se posiciona en relación con el elemento <html>.

- Uso típico: Ideal para posicionar un elemento exactamente donde lo deseas en la página, como una ventana modal o un tooltip.
- Ejemplo: Un mensaje emergente (tooltip) que aparece sobre un elemento para ofrecer información adicional.

fixed:

Similar a absolute, pero el elemento se posiciona en relación con la ventana del navegador, permaneciendo fijo durante el desplazamiento.

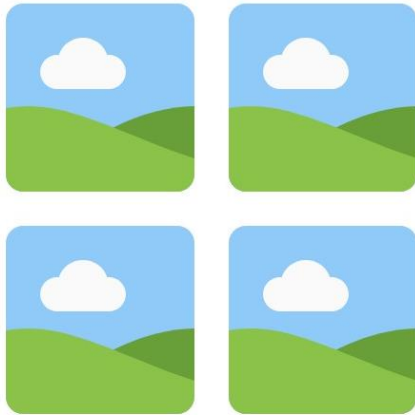
- Uso típico: Usado comúnmente para elementos de navegación o botones que necesitas que se queden visibles todo el tiempo, sin importar el desplazamiento de la página.
- Ejemplo: Una barra de navegación que permanece en la parte superior de la pantalla incluso cuando desplazas hacia abajo.

sticky:

Un híbrido entre relative y fixed. El elemento se posiciona como relative hasta que alcanza un punto de desplazamiento específico, momento en el cual se convierte en fixed.

- Uso típico: Útil para elementos que quieres que se comporten de manera normal hasta que llegues a un punto específico de desplazamiento, momento en el cual se fijan en su lugar.
- Ejemplo: Un encabezado de sección en una lista o tabla que se queda en la parte superior mientras desplazas por esa sección.

Layouts con modelo de caja y posicionamiento



El modelo de caja y las técnicas de posicionamiento en CSS son fundamentales para crear estructuras de página efectivas y responsivas. Este módulo explora cómo estos conceptos se pueden aplicar para diseñar encabezados, pies de página, columnas, barras laterales y más, y cómo asegurar que estos layouts se adapten de manera óptima a diferentes tamaños de pantalla mediante el uso de media queries.

Creación de un layout

Un "layout" en el contexto del diseño web se refiere a la disposición o estructura de los elementos en una página web. Incluye cómo se organizan y se presentan visualmente los distintos componentes, como encabezados, pies de página, columnas de contenido, barras laterales, imágenes, botones y enlaces, entre otros. El objetivo de un buen layout es presentar la información de manera clara y accesible, asegurando una experiencia de usuario coherente y atractiva.

A la hora de crear ese layout:

- el modelo de caja define cómo se estructuran y dimensionan los elementos en una página, considerando margen, borde, relleno y contenido. Es, por tanto, ideal para ajustar el espacio interior y exterior de los elementos, controlar su tamaño y gestionar cómo interactúan entre sí en el flujo normal del documento.
- el posicionamiento, por otro lado, determina la ubicación exacta de estas cajas en el layout de la página. Es ideal para ajustar el espacio

interior y exterior de los elementos, controlar su tamaño y gestionar cómo interactúan entre sí en el flujo normal del documento.

Juntos, estos conceptos permiten un control detallado sobre cómo y dónde se muestran los elementos.

Responsive design

El diseño responsivo, o "Responsive Design", es un enfoque de diseño web que busca asegurar que las páginas se vean y funcionen bien en una amplia gama de dispositivos y tamaños de pantalla, desde teléfonos móviles hasta computadoras de escritorio. La idea es que el diseño de un sitio web debería responder al entorno del usuario, adaptándose a las características del dispositivo, como su tamaño de pantalla, plataforma y orientación.

Relación con el modelo de caja y el posicionamiento

El modelo de caja y el posicionamiento en CSS son fundamentales para implementar un diseño responsivo, ya que determinan cómo se organizan y visualizan los elementos en la página.

- **Modelo de caja:** La forma en que definimos el margen, el borde, el relleno y las dimensiones de los elementos afecta directamente cómo se adaptan a diferentes tamaños de pantalla. Por ejemplo, el uso de unidades relativas (como %, em, rem, vw, vh) en lugar de unidades fijas (como px) permite que los elementos se escalen de manera más fluida. Así, un elemento puede ocupar width: 100% en un teléfono móvil para aprovechar todo el ancho disponible, mientras que en una pantalla más grande puede limitarse a width: 50% para no extenderse demasiado.
- **Posicionamiento:** Las técnicas de posicionamiento permiten controlar la ubicación de los elementos en la página. En un diseño responsivo, podemos cambiar el posicionamiento de los elementos según el tamaño de la pantalla mediante media queries. Por ejemplo, una barra lateral position: fixed; en una pantalla de escritorio puede convertirse en position: static; en dispositivos móviles para fluir normalmente dentro del contenido y no ocupar espacio valioso en la pantalla.

Básicamente, se define un layout predeterminado que generalmente está orientado a cierto tamaño de pantalla (a menudo el tamaño más común o el

que consideras principal para tu audiencia) y luego se definen otros layouts aplicables en otros tamaños de pantalla (por ejemplo, un Tablet o un iphone). Son las media queries incluidas en la programación de la página quienes gobiernan la aplicación de esos diferentes layouts.

Las media queries son una herramienta poderosa en CSS que te permite introducir puntos de quiebre (breakpoints). Estos puntos de quiebre actúan como condiciones que, cuando se cumplen (por ejemplo, cuando la ventana del navegador alcanza un cierto ancho), permiten que se apliquen estilos CSS diferentes. De esta manera, las media queries "gobiernan" qué layout o conjunto de estilos se aplica en función de las características del dispositivo, como el tamaño de la pantalla, la resolución, la orientación (vertical u horizontal), etc.

Conclusión

Terminamos aquí este módulo en el que hemos explorado el modelo de caja, el posicionamiento en CSS y cómo estos fundamentos interactúan con el diseño responsivo para crear páginas web que no solo son visualmente atractivas, sino también funcionales en una variedad de dispositivos y tamaños de pantalla.

Gracias por acompañarme en este viaje. Espero que este módulo te haya proporcionado una base sólida y la inspiración para llevar tus habilidades de diseño web al siguiente nivel.

Nos vemos en el siguiente módulo.

Recursos adicionales

A continuación, encontrarás una selección cuidadosa de enlaces a documentación oficial, tutoriales y herramientas de práctica que te ayudarán a profundizar en tu comprensión del Modelo de caja y el posicionamiento en CSS:

Enlaces a documentación oficial y tutoriales

- MDN Web Docs (Mozilla Developer Network): Un recurso imprescindible para cualquier desarrollador web. Ofrece documentación detallada y ejemplos sobre el Modelo de Caja, posicionamiento y mucho más. (https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/The_box_model)
- CSS-Tricks: Este sitio es una mina de oro de información y consejos sobre CSS. Para el Modelo de Caja y posicionamiento, no te pierdas sus guías y almanaques. (<https://css-tricks.com/the-css-box-model/>)
- W3Schools: Ofrece tutoriales claros y prácticos sobre muchos aspectos de CSS, incluido el Modelo de Caja y el posicionamiento. Es ideal para principiantes. https://www.w3schools.com/css/css_boxmodel.asp

Herramientas en línea para practicar CSS

- CodePen: Una plataforma social de desarrollo que permite escribir código en el navegador y ver los resultados en tiempo real. Es excelente para experimentar con CSS y compartir tus creaciones. (<https://codepen.io/>)
- CSS Zen Garden: Un proyecto clásico que demuestra lo que se puede lograr con diseño CSS. Puedes descargar los archivos HTML y CSS y experimentar con el diseño. (<https://csszengarden.com/>)