

Programación Web Desde Cero

Posicionamiento en CSS – Parte 1

¡Hola! 🖐️

Hasta ahora vimos cómo utilizar diferentes selectores de CSS para afectar a los elementos HTML y darle diseños como **background-color, color, font-size, etc.**

Pero, todavía nos falta una parte muy importante de CSS que es el posicionamiento y comportamiento de los elementos y cómo los ordenamos para que la página luzca como la del ejemplo que les proponemos 🙌 <https://www.minicarbono.com/>

Para ello vamos a ver propiedades como **display** y **position** que nos van a ayudar con este paso.

¡Comencemos! 🚀

Propiedad de Display

Cada elemento HTML tiene un valor de visualización predeterminado según el tipo de elemento que sea. El valor de visualización predeterminado para la mayoría de los elementos es **block** o **inline**.

- Los elementos de **bloque (block)** ocupan todo el ancho de la pantalla
- Los elementos **en línea (inline)** sólo ocupan el espacio necesario.

Otras propiedades de display que existen son:

- **none:** Oculta completamente el elemento, es decir, no se muestra en la página web. Es muy útil para hacer que un elemento desaparezca

temporalmente o para ocultarlo en diferentes condiciones de visualización (esto se logra manipulando el CSS con javascript).

- **inline-block:** Combina las características de los elementos inline y los elementos block. Los elementos inline-block se muestran como elementos en línea, pero permiten establecer ancho y altura como si fueran elementos de bloque.
- **inherit:** Hace que el elemento herede el valor de la propiedad display de su elemento padre.
- **flex:** Convierte un elemento en un contenedor flexible, lo que permite controlar el tamaño y la posición de sus elementos hijos.
- **grid:** Convierte un elemento en un contenedor de cuadrícula, lo que permite colocar elementos hijos en filas y columnas.

Propiedad de Position (posición)

Algunos valores comunes para la propiedad de position (“posicionamiento”) incluyen:

- **static:** Este es el valor predeterminado de la propiedad de "position". Cuando se establece el valor de "static", no se aplica ningún posicionamiento especial al elemento y se colocará en el lugar que le corresponda en el flujo normal del documento.
- **relative:** El elemento se posiciona en relación con su posición normal, es decir, puede moverse hacia arriba, abajo, izquierda o derecha desde su posición inicial.
- **absolute:** El elemento se posiciona en relación a su “ancestro posicionado” más cercano, es decir, al “elemento padre” que tenga un valor de posición distinto a "static". Si no hay ningún “ancestro posicionado”, el elemento se posiciona en relación al bloque contenedor inicial del documento (generalmente al “body”). Cuando se establece el valor de "absolute" en un elemento, se elimina del flujo normal del documento, lo que significa que no afecta a la posición de otros elementos en la página.

- **fixed:** El elemento se posiciona en relación con la ventana del navegador, lo que significa que permanece en la misma posición incluso cuando se desplaza la página.

Cuando se establece el valor de "fixed" en un elemento, también se elimina del flujo normal del documento. El elemento se coloca en la posición especificada con las propiedades "top", "right", "bottom" y "left". Si no se especifica ninguna de estas propiedades, el elemento se coloca en la esquina superior izquierda de su "ancestro posicionado" más cercano.

- **sticky:** El elemento se comporta como un elemento posicionado relativamente hasta que alcanza una posición específica en la página, donde se vuelve fijo.
- **inherit:** Hereda el comportamiento del elemento padre para dicha propiedad.

Floats (flotado)

En CSS, los **floats** son una propiedad que se utiliza para controlar el posicionamiento de elementos dentro de un contenedor. La propiedad float **se utiliza comúnmente para crear diseños de páginas web donde los elementos flotan a la izquierda o a la derecha del contenedor principal, permitiendo que otros elementos fluyan alrededor de ellos**. Sin embargo, es importante tener en cuenta que el uso excesivo de floats puede llevar a problemas de diseño y maquetación complicados.

1. Valores de la propiedad float:

- **none** (valor predeterminado): El elemento no se desplaza y se comporta como si no tuviera una propiedad float aplicada.
- **left:** El elemento flota hacia la izquierda del contenedor y otros elementos en línea fluyen alrededor de él por la derecha.
- **right:** El elemento flota hacia la derecha del contenedor y otros elementos en línea fluyen alrededor de él por la izquierda.

2. Flotar elementos:

Al aplicar **float: left;** o **float: right;** a un elemento, este se moverá a la izquierda o a la derecha del contenedor respectivamente. Los elementos que sigan a este elemento flotante fluirán alrededor de él. Es importante tener en cuenta que los elementos flotantes se eliminan del flujo normal del documento, lo que puede afectar el diseño si no se maneja correctamente.

3. Clearing Floats:

Debido a que los elementos flotantes son eliminados del flujo normal, **a menudo es necesario aplicar un estilo para "limpiar" el efecto del float. Esto se hace utilizando la propiedad clear.** Por ejemplo, si tienes varios elementos flotantes dentro de un contenedor y deseas que un nuevo elemento no flote junto a ellos, puedes aplicar **clear: both;** o **clear: left;** o **clear: right;** al nuevo elemento.

4. Problemas con floats:

Aunque los floats fueron ampliamente utilizados en el pasado para crear diseños de páginas web, presentan algunos problemas, como:

- **Problemas de altura irregular:** Los contenedores que solo contienen elementos flotantes pueden no expandirse automáticamente para abarcar su contenido, lo que puede afectar la alineación de otros elementos.
- **Problemas de desbordamiento:** Si los elementos flotantes no están correctamente limpiados, pueden provocar problemas de desbordamiento en los contenedores padre.
- **Diseño poco flexible:** El diseño basado en floats puede volverse complicado y difícil de mantener en diseños más complejos.

En los últimos años, se han desarrollado **técnicas y propiedades CSS más modernas**, como **Flexbox y Grid**, que ofrecen un mejor control sobre el diseño y la maquetación de las páginas web sin los problemas asociados a los floats. Estas técnicas son más recomendadas para diseños modernos y flexibles.

Line-height (interlineado)

La propiedad line-height en CSS se utiliza para controlar el espacio vertical entre líneas de texto dentro de un elemento. Básicamente, regula la altura de cada línea de texto, lo que afecta la distancia entre la línea base de una línea y la línea base de la siguiente línea. Puedes establecer el valor de line-height de varias maneras, incluyendo unidades de medida absolutas, unidades relativas y valores numéricos.

Aquí hay algunas características clave de la propiedad line-height:

- **Valor numérico:** Puedes establecer la propiedad usando un número decimal o entero, como 1.5 o 2. El valor numérico se multiplica por el tamaño de fuente actual para determinar el espacio entre líneas. Por ejemplo, si el tamaño de fuente es 16px y line-height es 1.5, entonces el espacio entre líneas será 24px ($16 * 1.5$).
- **Unidades absolutas:** Puedes usar unidades de medida absolutas, como px, pt o em, para establecer el espacio entre líneas con un valor fijo.
- **Unidades relativas:** También puedes utilizar unidades relativas, como rem, % o em, que se basan en el tamaño de fuente del elemento padre. Esto permite que el espacio entre líneas se adapte proporcionalmente a los cambios de tamaño de fuente en elementos secundarios.
- **Sin unidades:** También puedes usar valores sin unidades, como normal, que deja que el navegador establezca el espaciado por defecto, o inherit para heredar el valor de line-height del elemento padre.
- **Múltiples valores:** Es posible definir dos valores, separados por un espacio, para establecer diferentes alturas de línea para diferentes situaciones. Por ejemplo, line-height: 1.2 1.5; establecerá un espaciado de línea de 1.2 para el texto normal y 1.5 para el texto destacado.

El uso adecuado de line-height puede mejorar la legibilidad y el aspecto general de un diseño web. Un valor demasiado bajo puede hacer que el texto se vea apretado y difícil de leer, mientras que un valor demasiado alto puede resultar en un espaciado excesivo entre líneas. Experimentar con diferentes valores y observar cómo afectan

el diseño y la legibilidad es importante para lograr el aspecto deseado en tu página web.

Resumen

¡Felicidades!

Esperamos que esta información te haya sido de utilidad y que te lleves para repasar:

- Cuáles son las propiedades de Display
- Cuáles son las propiedades de Position
- Cuáles son las propiedades de Float