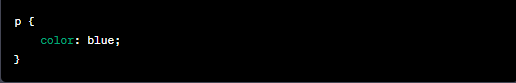
Selectores CSS

Los selectores son una parte fundamental del lenguaje CSS (Cascading Style Sheets) que se utiliza para **seleccionar elementos HTML y aplicarles estilos.** Los selectores en CSS son patrones que especifican qué elementos HTML serán afectados por las reglas de estilo.

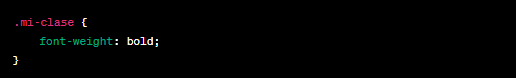
Cómo se aplican a elementos HTML:

❶ Selector de tipo: Selecciona todos los elementos de una etiqueta específica. Por ejemplo, el selector **“p”** selecciona todos los elementos de párrafo en el documento HTML:



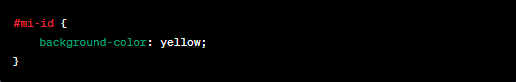
En este caso, todas las etiquetas **<p>** en el documento tendrán texto de color azul.

❷ Selector de clase: Selecciona elementos que tienen un atributo **class** específico. Por ejemplo, el selector **“.mi-clase”** selecciona todos los elementos con la clase **“mi-clase”**:



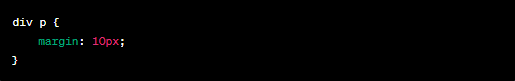
En este caso, todos los elementos con **class=“mi-clase”** tendrán un texto en negrita.

❸ Selector ID: Selecciona un elemento específico con un atributo **“id”**. Por ejemplo, el selector **“#mi-id”** selecciona un elemento con el **ID “mi-id****”**:



En este caso, el elemento con **id=“mi-id”** tendrá un fondo amarillo.

❹ Selector de descendencia: Selecciona elementos que son descendientes de otros elementos. Por ejemplo, el selector **“div p”** selecciona todos los elementos **“<p>”** que están dentro de un elemento **“<div>”**:

****

En este caso, todos los elementos **“<p>”** dentro de elementos **“<div>”** tendrán un margen de 10 píxeles.

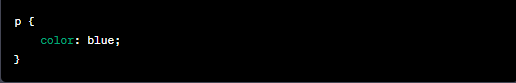
***Estos son solo ejemplos básicos de selectores en CSS. Los selectores pueden volverse mucho más complejos y sofisticados para aplicar estilos de manera precisa a elementos específicos en tu página web. La combinación de selectores y reglas de estilo CSS te permite personalizar la presentación de tu contenido HTML de manera efectiva.***

Propiedades CSS

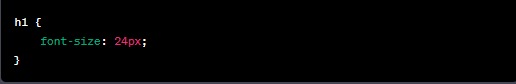
las **"propiedades"** son atributos que se utilizan para definir cómo se debe presentar un elemento HTML o XML en una página web. Cada propiedad CSS controla un aspecto específico del estilo de un elemento, como su color, tamaño, posición, márgenes, bordes, entre otros.

**Aquí hay algunos ejemplos de propiedades CSS y cómo se utilizan:**

❶ color: Controla el color del texto dentro de un elemento. Puedes establecer un color usando palabras clave (como **“red”** para rojo) o códigos de color (como **“#FF0000”** para rojo).



En este caso, todas las etiquetas **“<p>”** en el documento tendrán texto de color azul.

❷ font-size: Define el tamaño de la fuente del texto dentro de un elemento. 

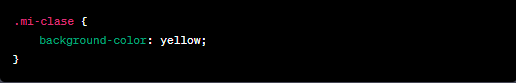
En este caso, todos los encabezados **“<h1>”** tendrán un tamaño de fuente de 24 píxeles.

❸ margin: Controla el espacio alrededor del elemento. Puedes establecer márgenes para la parte superior, derecha, inferior e izquierda del elemento.



Esto aplicará un margen de 10 píxeles a todos los elementos **“<div>”**.

❹ background-color: Define el color de fondo de un elemento.



Todos los elementos con la clase **“mi-clase”** tendrán un fondo amarillo.

Lista de propiedades más usadas:

**color:** Define el color del texto dentro de un elemento.

**font-family:** Establece la fuente o tipo de letra utilizada para el texto dentro de un elemento.

**font-size:** Controla el tamaño de la fuente del texto dentro de un elemento.

**font-weight:** Determina el grosor o el peso de la fuente, como "normal," "bold," o valores numéricos.

**font-style:** Define el estilo de la fuente, como "normal," "italic," o "oblique."

**text-decoration:** Controla la decoración del texto, como subrayado, tachado o ninguno.

**text-align:** Alinea el texto dentro de un elemento en relación con el elemento en sí, como "left," "center," "right," o "justify."

**line-height:** Establece la altura de línea, que es la distancia entre líneas de texto.

**letter-spacing:** Controla el espacio entre caracteres en el texto.

**word-spacing:** Define el espacio entre palabras en el texto.

**text-transform:** Cambia el caso del texto, como "uppercase," "lowercase," o "capitalize."

**background-color:** Define el color de fondo de un elemento.

**background-image:** Establece una imagen de fondo para un elemento.

**background-position:** Controla la posición de una imagen de fondo.

**background-repeat:** Define si una imagen de fondo se repite o no y cómo lo hace.

**margin:** Controla el espacio alrededor del elemento.

**padding:** Define el espacio dentro del elemento, entre el contenido y el borde.

**border:** Controla el borde alrededor de un elemento, incluyendo su estilo, grosor y color.

**width:** Establece el ancho de un elemento.

**height:** Define la altura de un elemento.

**display:** Controla cómo se muestra el elemento, como "block," "inline," o "none."

**position:** Define el tipo de posicionamiento del elemento, como "relative," "absolute," o "fixed."

**float:** Permite que un elemento flote a la izquierda o a la derecha de su contenedor.

**clear:** Controla cómo los elementos deben comportarse en relación con los elementos flotantes.

**overflow:** Define cómo se maneja el contenido que desborda el tamaño de un elemento.

**visibility:** Controla si un elemento es visible o no en la página.

**opacity:** Establece la opacidad (transparencia) de un elemento.

**box-shadow:** Agrega sombras alrededor de un elemento.

**text-shadow:** Agrega sombras al texto dentro de un elemento.

**z-index:** Controla el apilamiento de elementos en capas en la página.

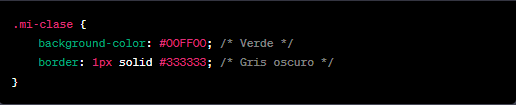
***Estas son algunas de las propiedades CSS más utilizadas y sus descripciones básicas. CSS ofrece una amplia gama de propiedades para controlar la apariencia y el diseño de los elementos HTML en una página web. La elección de las propiedades adecuadas depende de tus necesidades específicas de diseño.***

Valores CSS

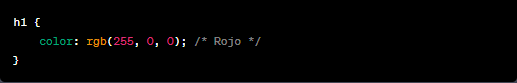
los **“valores”** se utilizan junto con las propiedades para definir cómo se debe aplicar un estilo específico a un elemento HTML. Los valores determinan cómo se verán o se comportarán los elementos cuando se apliquen las reglas de estilo.

❶ Valores de color: Se utilizan para definir el color de elementos HTML, como texto, fondos, bordes y más. Puedes especificar colores de varias maneras, ya sea mediante **palabras clave**, códigos de color **hexadecimales**, valores **RGB** (Red, Green, Blue), valores **RGBA** (con una componente alfa para transparencia) o **nombres de color**.

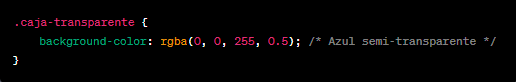
* Palabras clave: Algunos colores comunes se pueden especificar mediante palabras clave simples:
  + **red:** Rojo.
  + **blue:** Azul.
  + **green:** Verde.
  + **black:** Negro.
  + **white:** Blanco.
  + **yellow:** Amarillo.
  + **orange:** Naranja.
  + **purple:** Morado.
  + **gray** o **grey:** Gris.
* Código de color hexadecimal: Los códigos de color hexadecimales utilizan valores hexadecimales para representar el color en el formato **"#RRGGBB"**, donde **RR** es la **intensidad de rojo**, **GG** es la **intensidad de verde** y **BB** es la **intensidad de azul**. Cada componente se representa con dos dígitos hexadecimales (de 00 a FF). ***Ejemplo:***



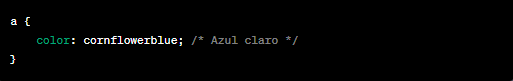
* Valores RGB: Puedes definir colores mediante valores **RGB**, que especifican la intensidad de cada componente de color (**rojo, verde y azul**) en una escala de **0 a 255**. ***Ejemplo:***



* Valores RGBA: Los valores **RGBA** son similares a los valores RGB, pero **incluyen una componente alfa (transparencia) adicional**, que varía de **0** (completamente transparente) **a 1** (completamente opaco). ***Ejemplo:***



* Nombres de color: Además de las palabras clave comunes, **CSS incluye una lista de nombres de color predefinidos** que pueden utilizarse para representar colores específicos. ***Ejemplo:***



❷ Valores de tamaño: Se utilizan para **especificar dimensiones**, **tamaños** y **espaciado** en elementos HTML. Estos valores pueden aplicarse a propiedades como font-size, width, height, margin, padding, y muchas otras.

* Píxeles (px): Los píxeles son una unidad de medida fija y comúnmente utilizada en CSS. **Un píxel es la unidad de medida más pequeña en una pantalla.**
* Porcentajes (%): Los porcentajes se utilizan para definir **tamaños relativos en relación con el tamaño del elemento padre.**
* Em (em): El **"em"** es una unidad de **medida relativa que se basa en el tamaño de fuente del elemento padre.** **Un valor de 1em equivale al tamaño de fuente actual del elemento.**
* Rem (rem): Similar a **"em"**, pero **se basa en el tamaño de fuente del elemento raíz** (generalmente el **elemento <html>**). Esto hace que sea más predecible y fácil de manejar.
* Centímetros (cm), milímetros (mm), pulgadas (in): Estas **unidades de medida son absolutas** y se utilizan principalmente para dimensiones de impresión y otros contextos fuera de la web.
* Puntos (pt) y Picas (pc): Estas **unidades** también son **absolutas** y se utilizan principalmente para impresión y maquetación de documentos. **Un punto es igual a 1/72 de pulgada y una pica es igual a 12 puntos.**

Estos son algunos ejemplos de unidades de medida de tamaño en CSS. **La elección de la unidad depende del contexto y las necesidades de diseño de tu proyecto web**. Las unidades relativas como **em** y **rem** suelen ser más flexibles y escalables, mientras que las unidades absolutas como **px** pueden proporcionar un control preciso sobre el diseño.

❸ Valores de posición: Se utilizan para controlar cómo se posiciona un elemento en relación con su contexto o con otros elementos en la página web.

* static: Este **es el valor predeterminado** para la propiedad position y significa que el elemento se coloca de acuerdo con el flujo normal del documento. No se ve afectado por las propiedades top, right, bottom o left, ni por las propiedades z-index.
* relative: Este valor **permite ajustar la posición de un elemento** con respecto a su posición normal en el flujo del documento utilizando las propiedades top, right, bottom o left. **El elemento sigue ocupando espacio en su posición original.**
* absolute: Cuando se establece **position: absolute;**, el **elemento se posiciona con respecto a su elemento padre más cercano** **que tenga una propiedad position distinta de static.** Si no hay un padre con posición diferente de static, el elemento se posiciona con respecto al elemento raíz del documento (generalmente el **<body>**). Los valores top, right, bottom y left se utilizan para determinar la posición exacta del elemento.
* fixed: Este valor coloca **el elemento en una posición fija en la ventana del navegador**, **independientemente del desplazamiento (scroll) de la página**. Los valores top, right, bottom y left se utilizan para especificar la posición exacta.
* sticky: Los elementos con **position: sticky;** se comportan inicialmente como **elementos relative**. Sin embargo, **cuando el usuario desplaza la página y el elemento alcanza cierta posición** (generalmente definida con top, right, bottom o left), **el elemento se convierte en fijo en esa posición.**

Estos son los valores de posición más comunes en CSS. La elección del valor adecuado **depende de cómo desees que se comporte y se posicione un elemento en tu diseño web.** Cada uno tiene su uso particular y puede ser útil en diferentes situaciones de diseño.

❹ Valores de alineación (text-align): Se utilizan para controlar la alineación de elementos dentro de su contenedor o para alinear el contenido dentro de un elemento en relación con su propio espacio.

* left (izquierda): Alinea un elemento o su contenido hacia la izquierda dentro de su contenedor. En el caso de elementos de bloque, esto significa que el elemento estará en la parte izquierda del contenedor.
* center (centro): Alinea un elemento o su contenido en el centro horizontalmente dentro de su contenedor.
* right (derecha): Alinea un elemento o su contenido hacia la derecha dentro de su contenedor. En el caso de elementos de bloque, esto significa que el elemento estará en la parte derecha del contenedor.
* justify (justificar): Alinea el texto dentro de un elemento de manera que los márgenes izquierdo y derecho estén perfectamente alineados, creando un efecto de justificación. Esto generalmente se usa para el texto en bloques de texto más largos.
* start (inicio): Alinea el texto o el contenido de un elemento según la dirección de escritura del idioma. Por defecto, esto es lo mismo que "left" en idiomas de escritura de izquierda a derecha, pero en idiomas de escritura de derecha a izquierda, se alinearán hacia la derecha.
* end (fin): Al igual que **"start,"** alinea el texto o el contenido de un elemento según la dirección de escritura del idioma. Por defecto, esto es lo mismo que "right" en idiomas de escritura de derecha a izquierda, pero en idiomas de escritura de izquierda a derecha, se alinearán hacia la izquierda.

**Estos son los valores de alineación más comunes utilizados con la propiedad text-align** en CSS para controlar la alineación horizontal del texto dentro de elementos HTML, como **párrafos**, **encabezados** y **divs**. La elección del valor de alineación depende del diseño que desees lograr en tu página web.

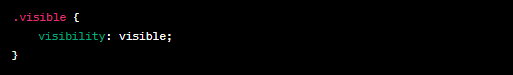
❺ Valores de borde: Se utilizan para controlar el estilo, el grosor y el color de los bordes de un elemento HTML. Se pueden aplicar a elementos como **divs**, **imágenes**, **botones** y otros elementos HTML para personalizar su apariencia.

* Estilo de borde (border-style): Controla cómo se ve el borde del elemento. Puedes elegir entre varios estilos, como **"solid"** (sólido), **"dotted"** (punteado), **"dashed"** (rayado), **"double"** (doble línea) y otros.
* Grosor de borde (border-width): Define el grosor del borde en píxeles u otras unidades de medida.
* Color de borde (border-color): Establece el color del borde utilizando palabras clave de color o códigos de color.
* Borde individual: Puedes especificar los valores de estilo, grosor y color para los bordes individuales de un elemento, como "**border-top**," "**border-right**," "**border-bottom**" y "**border-left**".
* Borde redondeado: Permite crear esquinas redondeadas en un elemento mediante propiedades como **"border-radius".** Puedes establecer radios de esquina diferentes para cada esquina del elemento.
* Borde degradado: Puedes aplicar degradados a los bordes de un elemento utilizando propiedades como "**border-image**" y "**border-image-source**."Esto permite que los bordes tengan un aspecto degradado o de imagen.

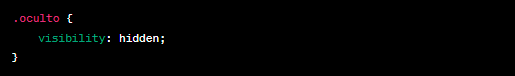
***Estos son algunos ejemplos de cómo se pueden utilizar los valores de borde en CSS para personalizar la apariencia de los bordes de los elementos HTML. Puedes combinar estos valores para crear una variedad de estilos de borde diferentes y lograr el diseño que desees en tu página web.***

❻ Valores de visibilidad: La propiedad de **"visibilidad"** se utiliza para controlar si un elemento HTML es **visible o invisible** en la página web. Los valores que puedes aplicar a la propiedad de visibilidad son:

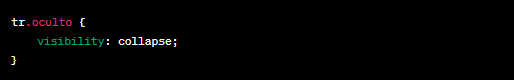
* visible: Este es el valor predeterminado. Indica que el elemento es visible y se muestra en la página.



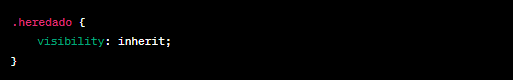
* hidden: Este valor hace que el elemento sea invisible, pero aún ocupa espacio en el diseño de la página. Es decir, **el elemento se "oculta" pero su espacio se reserva.**



* collapse: Este valor se usa comúnmente con tablas HTML para **ocultar una fila o columna de una tabla**. Similar a "hidden," pero especialmente diseñado para tablas.



* inherit: Este **valor hereda la propiedad de visibilidad de su elemento padre.** Si el elemento padre está visible, el elemento hijo también lo estará.



El valor más comúnmente utilizado es "visible" para asegurarse de que un elemento sea visible en la página. "hidden" es útil cuando deseas ocultar un elemento sin cambiar el diseño de la página, ya que el espacio se mantiene ocupado. "collapse" es específico para tablas HTML y se utiliza para ocultar filas o columnas en una tabla. "inherit" se utiliza para que el elemento herede la propiedad de visibilidad de su elemento padre.

***Recuerda que cambiar la propiedad de visibilidad a "hidden" o "collapse" no elimina completamente el elemento de la estructura del documento, solo lo oculta visualmente. Puedes cambiar la propiedad de visibilidad en respuesta a eventos o interacciones para mostrar u ocultar dinámicamente elementos en tu página web.***

❼ Valores de opacidad: En CSS, la propiedad de opacidad se utiliza para controlar la transparencia de un elemento HTML. Los valores de opacidad pueden variar de 0 a 1, donde:

* 0: El elemento es completamente **transparente** y no se verá en la página.
* 1: El elemento es completamente **opaco** y se verá normalmente sin transparencia.

***Puedes utilizar cualquier valor decimal entre 0 y 1 para definir el nivel de opacidad que deseas aplicar a un elemento.***

❽ Valores de posición de fondo: Los valores de posición de fondo en CSS se utilizan para controlar la posición de una imagen de fondo dentro de un elemento HTML. Estos valores definen dónde se colocará la imagen de fondo en relación con el elemento. Aquí tienes algunos de los valores de posición de fondo más comunes:

* center: La imagen de fondo se centra tanto horizontal como verticalmente dentro del elemento.
* left: La imagen de fondo se alinea a la izquierda dentro del elemento.
* right: La imagen de fondo se alinea a la derecha dentro del elemento.
* top: La imagen de fondo se alinea en la parte superior del elemento.
* bottom: La imagen de fondo se alinea en la parte inferior del elemento.
* left top: La imagen de fondo se alinea en la esquina superior izquierda del elemento.
* right top: La imagen de fondo se alinea en la esquina superior derecha del elemento.
* center bottom: La imagen de fondo se centra horizontalmente y se alinea en la parte inferior del elemento.
* 50% 50%: También puedes especificar la posición en píxeles o porcentajes. "50% 50%" centraría la imagen tanto horizontal como verticalmente en el elemento.

***Estos son ejemplos de valores de posición de fondo en CSS que puedes utilizar para controlar dónde se coloca una imagen de fondo en relación con el elemento contenedor. Puedes combinar valores para lograr posiciones precisas y efectos de diseño personalizados.***