

Guía CSS

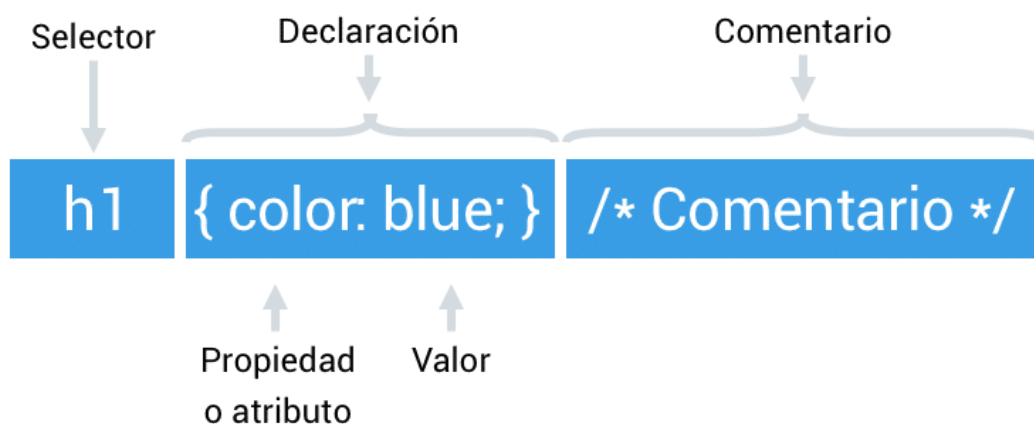
Temas

- ¿Qué es CSS?
- ¿Cómo aplicar estilos CSS?
- Tipos de selectores en CSS
- Pseudo-clases y Pseudo-elementos
- Colores y fondo en CSS, atributos y propiedades
- Propiedades de texto en CSS
- Propiedades de las fuentes en CSS
- Propiedades de las listas en CSS
- Propiedades de las tablas
- Unidades de medida en CSS
- Modelo de cajas: márgenes, relleno y bordes
- Propiedades width, height, max-width, min-width, max-height y min-height
- Posición y comportamiento de contenedores en CSS

1. ¿Qué es CSS (Cascading Style Sheets)?

CSS (*Cascading Style Sheets*, Hojas de Estilo en Cascada en español) es un lenguaje de estilo utilizado para definir la presentación y el diseño de documentos HTML. Con CSS, se puede controlar la apariencia de los elementos de una página web, como el color de fondo, la tipografía, el espaciado, los bordes y la disposición de los elementos en pantalla. Veamos detalladamente qué es CSS y cuál es su sintaxis.

1.2. Sintaxis CSS



1. **Selector** (*selector*): indica sobre qué elemento se aplican los estilos CSS.
2. **Propiedad o atributo** (*property or attribute*): indica qué característica se va a cambiar.
3. **Valor** (*value*): indica el valor de la propiedad que se desea modificar.
4. **Comentario** (*comment*): los comentarios se escriben entre el carácter de apertura `/*` y el carácter de cierre `*/`.

2. Estilos CSS en un documento HTML

Hay tres formas de aplicar estilos CSS en un documento HTML: en línea, incrustado en la cabecera y mediante hojas de estilo externas

2.1. CSS en línea

Los estilos en línea son declaraciones CSS que se integran en las etiquetas HTML mediante el atributo `style`. Este método tan solo afecta al elemento en el que se integra el código. El CSS en línea es **complicado de entender y mantener ya que mezcla los estilos CSS con el código HTML**.

Ejemplo:

```
<p style="color:green;">Párrafo de color verde.</p>
```

2.2. CSS incrustado en la cabecera

Otra manera muy simple de añadir estilo con CSS es utilizando la etiqueta `<style>` en la cabecera `<head>` del fichero HTML del sitio. La **desventaja** de este método es que a la hora de realizar **cualquier cambio, se debe realizar en múltiples páginas diferentes y el código estará repetido.**

Ejemplo:

```
<html>
<head>
  <title>CSS incrustado en la cabecera</title>
  <style>p{color:green;}</style>
</head>
<body>
  <p>Párrafo de color verde.</p>
</body>
</html>
```

2.3. CSS en hojas de estilo externas

Mediante hojas de estilo externas se consigue separar el archivo de estilos del fichero HTML. El archivo de estilos cuenta con la extensión `.css` y se referencia desde HTML mediante el elemento `<link>`. Este es el **método más eficiente y más sencillo de mantener ya que el código CSS se encuentra separado del fichero HTML.**

Ejemplo:

inicio.html

```
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <p>Párrafo de color verde.</p>
</body>
</html>
```

styles.css

```
p {color:green;}
```

3. Tipos de selectores en CSS

Los selectores nos ayudan a indicar el elemento sobre el que se van a aplicar los estilos. Los selectores pueden apuntar a elementos específicos, clases, identificadores o incluso atributos de un elemento.

Existen muchos tipos de selectores y algunos de los más destacados son los que se detallan a continuación.

3.1. Selector universal

Sintaxis: `* { atributo:valor; }`

Ejemplo: `* { color: grey; }/*` El estilo se aplicará a todos los elementos de la página`*/`

El **selector universal (*)** es útil cuando deseas aplicar un estilo a todos los elementos en una página web sin excepción. Es una forma rápida y sencilla de establecer reglas generales que afectarán a todos los elementos en el documento. Por ejemplo, si deseas eliminar todos los márgenes y rellenos predeterminados de los elementos en tu página para empezar desde cero, puedes usar el selector universal para restablecerlos.

3.2. Selector etiqueta

Sintaxis: `etiqueta { atributo:valor }`

Ejemplo: `p {color: green;} /*` El estilo se aplicará a todos los elementos `<p>.*/*`

3.3. Selector clase

Sintaxis: `.clase { atributo:valor }`

Ejemplo: `.blend{color: red;} /*` El estilo se aplicará a cualquier elemento que tenga la clase `.blend` `*/`

Un elemento puede tener múltiples clases: `<div class="blend otro-estilo"></div>`
En este ejemplo, el elemento `<div>` tendrá los estilos definidos para ambas clases `.blend` y `.otro-estilo`. Sin embargo, si hay propiedades CSS que se repiten en ambas clases, se aplicarán los estilos de la última clase declarada. Es decir, si tanto `.blend` como `.otro-estilo` definen la propiedad `color`, prevalecerá el color definido en `.otro-estilo` porque es la última clase definida. Esto proporciona un mecanismo para controlar la prioridad de los estilos cuando se aplican múltiples clases a un elemento.

3.4. Selector identificador

El **selector identificador** utiliza el atributo id para seleccionar un elemento. Solo puede haber un elemento con un id dado en un documento.

Sintaxis: #id { atributo:valor }

Ejemplo: #cent {color: blue;} /* El estilo se aplicará al elemento que tenga el id #cent */

3.5. Selector descendiente

Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre.

Sintaxis: selector1 selector2 selectorN {atributo: valor;} /* El estilo se aplica sobre el selector N */

Ejemplo: div p { color: black;} /* El estilo se aplica a todos los párrafos que se encuentren dentro de una etiqueta div */

3.6. Combinación de selectores

La combinación de selectores nos permite dar un estilo a todos los selectores indicados.

Sintaxis: selector1, selector2, selector3{atributo: valor;} /* El estilo se aplica sobre los selectores indicados */

Ejemplo: div, p { color: orange;} /* El estilo se aplica a todos los divs y párrafos */

3.7. Selector de hijos

Se usa para seleccionar un elemento que es *hijo* de otro elemento.

Sintaxis: selector1 > selector2 {atributo: valor;} /* El estilo se aplica sobre el selector 2 */

Ejemplo: div > p { color: white;} /* El estilo se aplica a todos los párrafos que sean hijos de un div */

Diferencia entre selector de hijos y selector descendiente. Los **hijos** son los elementos que se encuentran un nivel directamente debajo del mismo elemento section. Los **descendientes** son todos los elementos que se encuentran dentro del elemento section sean los mismos **hijos** o los **hijos** de los **hijos**.

3.8. Selector adyacente

Se usa para seleccionar elementos que son *hermanos*, es decir, su elemento padre es el mismo y están seguidos en el código HTML.

Sintaxis: selector1 + selector2{ atributo: valor; } /* El estilo se aplica al selector 2 */

Ejemplo: `div + p { color: black; }` /* El estilo se aplica a todos los párrafos que sean hermanos de un div */

4. Pseudo-clases y pseudo-elementos en CSS

Gracias a las pseudo-clases y los pseudo-elementos de CSS podemos realizar una selección más específica de los elementos a los que queremos aplicar un cierto estilo sin necesidad de crear una clase concreta.

4.1. Pseudo-clases para selección de hijos o hermanos

Las pseudo-clases más importantes para la selección de hijos o hermanos son las que se muestran en la siguiente tabla.

Pseudo-clase	Descripción
:first-child	Primer hijo
:last-child	Último hijo
:first-of-type	Primer hermano de su tipo
:last-of-type	Último hermano de su tipo
:only-child	Hijos únicos
:only-of-type	Únicos hermanos de su tipo
:empty	Elementos que no tienen hijos
:nth-child(n)	Enésimo elemento hijo
:nth-last-child(n)	Enésimo elemento hijo contando desde el último
:nth-of-type(n)	Enésimo hermano de su tipo
:nth-last-of-type(n)	Enésimo hermano de su tipo comenzando desde el último

Ejemplo pseudoclases first-child, last-child y only-child:

Vamos a utilizar las propiedades first-child, last-child y only-child para seleccionar los párrafos que sean el primer y el último hijo de un contenedor o que sea hijo único.

```
p:first-child {color: red;} /*Todos los párrafos que son primer hijo*/  
p:last-child {color: blue;} /*Todos los párrafos que son el último hijo*/  
p:only-child {color: green;} /*Todos los párrafos que son hijos únicos*/
```

Ejemplo pseudo-clase nth-child:

Veamos cómo funciona la pseudo-clase nth-child mediante un ejemplo en el que aprovechamos la jerarquía de los elementos para referenciar una etiqueta. En el siguiente código tenemos cuatro elementos <p> que son hermanos entre sí e hijos del elemento <div>.

```
<div>  
  <p>Primer párrafo</p>  
  <p>Segundo párrafo</p>  
  <p>Tercer párrafo</p>  
  <p>Cuarto párrafo</p>  
</div>
```

Utilizando la pseudo-clase **nth-child()** podemos seleccionar un hijo específico. El número que indiquemos entre paréntesis será la posición del hijo.

En el siguiente código, seleccionaremos todos los párrafos que sean el primer, segundo y cuarto hijo de un contenedor (div, span, etc.).

```
p:nth-child(1){ color: red; }  
p:nth-child(2){ color: green; }  
p:nth-child(4){ color: blue; }
```

Ejemplo odd (impar) y even (par):

Los valores *odd* y *even* son muy útiles para seleccionar elementos pares e impares. El valor *even* representa elementos en posición par y el valor *odd* representa los elementos en posición impar. En el siguiente ejemplo en el que se ha creado una tabla se muestran en verde las filas impares y en rojo las filas pares.

```
tr:nth-child(odd){ color: green;} /* Filas impares */  
tr:nth-child(even){ color: red;} /* Filas pares */
```


4.2. Pseudo-clases para los estados de un elemento

Podemos utilizar diferentes pseudo-clases para definir las propiedades de ciertos elementos con diferentes estados. Uno de los usos más comunes es en los enlaces `<a>` aunque se pueden aplicar en otros elementos.

Pseudo-clase	Descripción
<code>:link</code>	No visitado por el usuario
<code>:visited</code>	Visitado por el usuario
<code>:hover</code>	Modifica el estilo cuando un elemento apuntador pasa por encima
<code>:active</code>	Se activa cuando el usuario pulsa el elemento
<code>:focus</code>	Se activa cuando tiene el foco sobre el elemento

Aplica las siguientes propiedades sobre un enlace y observa las diferencias entre los distintos valores.

```
a:hover { text-decoration: underline;}  
a:active { background-color: yellow;}  
a:link { color: green;}  
a:visited { color: brown;}
```

4.3. Pseudo-elementos

A diferencia de las pseudo-clases, los pseudo-elementos no describen un estado sino que permiten añadir estilos a una parte específica del documento. Por ejemplo, el pseudo-elemento `:first-line` selecciona la primera línea del elemento indicado por el selector.

Pseudo-elemento	Descripción
::first-line	Primera línea de texto de un elemento
::first-letter	Primera letra de la primera línea de texto de un elemento
::before	Añade contenido al principio del documento
::after	Añade contenido al final del documento
::selection	Coge la porción del texto que se está seleccionando por el usuario

Aplica las siguientes propiedades sobre diferentes elementos y observa las diferencias entre los distintos resultados.

```

p::first-line { color: green; }
h2::first-letter { color: blue; }
p::after { content: " - Aloja more don"; color: red; }
p::before { content: "*. "; }
p::selection { color: brown; }

```

5. Colores y fondo en CSS

5.1. Valores de los colores

Los valores de los colores se pueden definir de varias formas:

- **Nombre predefinido:** red, yellow, blue, green...
- **Código hexadecimal:** #RRGGBB o #RRGGBBAA
- **RGB:** Red, Green, Blue
- **RGBA:** Red, Green, Blue, Alpha
- **HSL:** Hue, Saturation, Lightness
- **HSLA:** Hue, Saturation, Lightness, Alpha

5.2. Propiedades más utilizadas

Algunas de las propiedades CSS relacionadas con el color y el fondo más utilizadas son las siguientes:

Propiedad	Descripción	Valores
<code>color</code>	Color del texto	RGB HSL HEX nombre del color RGBA HSLA
<code>background-color</code>	Color de fondo	RGB HSL HEX nombre del color RGBA HSLA transparent
<code>background-image</code>	Imagen de fondo	url(«...») none
<code>background-repeat</code>	Repetición de la imagen de fondo	repeat repeat-x repeat-y no-repeat
<code>background-attachment</code>	Desplazamiento de la imagen de fondo	scroll fixed
<code>background-position</code>	Posición de la imagen de fondo	percentage length left center right
<code>background-size</code>	Tamaño de la imagen de fondo	auto cover contain valor
<code>Opacity</code>	Transparencia de un elemento	[0 – 1] (0 → totalmente transparente)

5.2.1. color – Color del texto

La propiedad color se utiliza para definir el color del texto en un elemento HTML.

Ejemplo:

```
p { color: #FF0000; /* Color rojo en formato HEX */ }
```

5.2.2. background-color – Color de fondo

La propiedad background-color se utiliza para establecer el color de fondo de un elemento HTML, como un div o una sección.

Ejemplo:

```
div { background-color: rgb(0, 128, 255); /* Color de fondo azul en formato RGB */ }
```

5.2.3. background-image – Imagen de fondo

La propiedad background-image se utiliza para establecer una imagen como fondo de un elemento HTML. Puedes proporcionar la ruta de la imagen usando el formato url("..."). El valor none se utiliza para eliminar cualquier imagen de fondo.

Ejemplo:

```
section { background-image: url("imagen-fondo.jpg"); }
```

5.2.4. background-repeat – Repetición de la imagen de fondo

La propiedad background-repeat controla la forma en que la imagen de fondo se repite en el elemento. Puedes elegir entre repeat (repetir en ambas direcciones), repeat-x (repetir solo horizontalmente), repeat-y (repetir solo verticalmente) y no-repeat (no repetir la imagen).

Ejemplo:

```
header { background-image: url("patron-fondo.png"); background-repeat: repeat-x; /* Repetir solo horizontalmente */ }
```

5.2.5. background-attachment – Desplazamiento de la imagen de fondo

La propiedad background-attachment controla si la imagen de fondo se desplaza con el contenido de la página (scroll) o permanece fija en su posición mientras se desplaza el contenido (fixed).

Ejemplo:

```
body { background-image: url("fondo-estrellado.jpg"); background-attachment: fixed; /* Fondo fijo en la ventana */ }
```

5.2.6. background-position – Posición de la imagen de fondo

La propiedad background-position se utiliza para definir la posición inicial de la imagen de fondo dentro del elemento.

Ejemplos:

Posición en el Centro Superior: Este ejemplo coloca la imagen de fondo en el centro superior del elemento.

```
div { background-image: url("imagen-fondo.jpg"); background-position: center top; }
```

Posición en la Esquina Superior Derecha: En este caso, la imagen de fondo se coloca en la esquina superior derecha del elemento.

```
div { background-image: url("imagen-fondo.jpg"); background-position: right top; }
```

Posición en la Esquina Inferior Izquierda: Aquí, la imagen de fondo se encuentra en la esquina inferior izquierda del elemento.

```
div { background-image: url("imagen-fondo.jpg"); background-position: left bottom; }
```

Posición Personalizada en Píxeles: Puedes especificar coordenadas en píxeles para posicionar la imagen de fondo de manera precisa.

```
div { background-image: url("imagen-fondo.jpg"); background-position: 50px 30px; /* 50px desde la izquierda, 30px desde arriba */ }
```

Posición en el Centro del Contenedor: Para centrar la imagen de fondo tanto vertical como horizontalmente en el contenedor.

```
Div { background-image: url("imagen-fondo.jpg"); background-position: center center; }
```

Posición en Porcentajes: Puedes utilizar porcentajes para posicionar la imagen de fondo en relación con el tamaño del elemento.

```
Div { background-image: url("imagen-fondo.jpg"); background-position: 25% 75%;  
/* 25% desde la izquierda, 75% desde arriba */ }
```

5.2.7. background-size – Tamaño de la imagen de fondo

La propiedad background-size se utiliza para controlar el tamaño de la imagen de fondo en relación con el elemento que la contiene. La sintaxis de la propiedad background-size es la siguiente:

```
background-size: valor;
```

Donde «valor» puede ser uno de los siguientes:

auto: Este es el valor predeterminado. La imagen de fondo conserva su tamaño original y se repite si es necesario para llenar el contenedor.

cover: La imagen de fondo se escala para que cubra completamente el contenedor, recortando cualquier exceso que no quepa. Esto puede dar como resultado que la imagen se estire o recorte.

contain: La imagen de fondo se escala para que quepa completamente dentro del contenedor sin recortarla. Puede haber espacio vacío alrededor de la imagen si el contenedor tiene una relación de aspecto diferente a la imagen.

valor personalizado: Puedes especificar el tamaño de la imagen de fondo utilizando valores de longitud (píxeles, em, porcentaje) o usando auto para una de las dimensiones (ancho o alto) y un valor específico para la otra dimensión.

Ejemplos:

```
/* La imagen se ajusta para cubrir completamente el contenedor */
```

```
background-size: cover;
```

```
/* La imagen se ajusta para caber completamente dentro del contenedor */
```

```
background-size: contain;
```

```
/* La imagen tiene un ancho fijo de 200px y se ajusta automáticamente en altura */  
background-size: 200px auto;
```

```
/* La imagen tiene un ancho fijo del 50% del contenedor y una altura fija de 100px  
*/
```

```
background-size: 50% 100px;
```

La propiedad `background-size` es especialmente útil cuando se trabaja con imágenes de fondo en elementos como encabezados, secciones, divs, etc., y se desea controlar cómo se ajusta la imagen de fondo en relación con el tamaño del contenedor.

5.2.8. `opacity` – Opacidad del Elemento

La propiedad `opacity` se utiliza para controlar la transparencia de un elemento. Un valor de 1 representa opacidad completa (sin transparencia), mientras que un valor de 0 hace que el elemento sea completamente transparente. Los valores entre 0 y 1 permiten un grado variable de transparencia.

Ejemplo:

```
.caja-transparente { background-color: #000; /* Fondo negro */ opacity: 0.7; /*  
70% de opacidad */ }
```

6. Propiedades de texto en CSS

Las propiedades de texto son las que nos permiten controlar el texto como bloque, es decir, afectan al interlineado, a la separación entre palabras, al tabulado, etc.

Propiedad	Descripción	Valores
text-indent	Desplazamiento de la primera línea del texto	longitud porcentaje
text-align	Alineamiento del texto	left right center justify
text-decoration	Efectos de subrayado y tachado	none underline overline line-through
letter-spacing	Espacio entre caracteres	normal longitud
word-spacing	Espacio entre palabras	normal longitud
text-transform	Transformación a mayúsculas / minúsculas	capitalize uppercase lowercase none
line-height	Tamaño del espacio entre líneas (interlineado)	longitud porcentaje
vertical-align	Alineación vertical	top middle bottom baseline sub super valor

7. Propiedades de las fuentes en CSS

Las propiedades CSS de las fuentes son las que permiten controlar el tamaño, el tipo, el grosor o el estilo de las letras, entre otras cosas. En la siguiente tabla se muestran las propiedades de las fuentes más destacadas.

Propiedad	Descripción	Valores
font-family	Familias de fuentes	nombre-familia *
font-style	Estilo de la fuente	normal italic oblique
font-variant	Convierte a mayúsculas manteniendo todas las letras en un tamaño inferior a la primera	normal small-caps
font-weight	Anchura de los caracteres. Normal = 400, Negrita = 700	normal bold bolder lighter 100 200 300 400 500 600 700 800 900
font-size	Tamaño de la fuente	xx-small x-small small medium large x-large xx-large larger smaller longitud porcentaje

Ejemplo:

```
.a { font-family: "Times New Roman", Times, serif; }  
.b { font-family: Arial, Helvetica, sans-serif; }
```

8. Propiedades de las listas

Propiedad	Descripción	Valores
list-style-type	Estilo aplicable a los marcadores visuales o viñetas de las listas	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none
list-style-image	Imagen aplicable a las viñetas de las listas	url() none
list-style-position	Posición de las viñetas dentro de la lista	inside outside
list-style	Permite establecer varios estilos de la lista en una sola propiedad	list-style-type list-style-position list-style-image

9. Propiedades de las tablas

Las propiedades CSS de las tablas son las que nos permiten controlar los estilos de los títulos de la tabla, el tamaño de las celdas, las filas y las columnas o espaciado entre los bordes.

Propiedad	Descripción	Valores
caption-side	Posición del título respecto la tabla	top bottom
table-layout	Formato de las celdas, filas y columnas	auto fixed
border-collapse	Selección del modelo de los bordes	collapse separate
border-spacing	Espaciado entre los bordes de celdas adyacentes	longitud
empty-cells	Visibilidad de los bordes de celdas sin contenido	show hide

9.1. Caption-side

La propiedad caption-side se utiliza para controlar la ubicación del título (caption) en una tabla HTML. Cuenta con los siguientes valores:

1. top (valor predeterminado): Esto coloca el título en la parte superior de la tabla, encima del contenido de la tabla.
2. bottom: Esto coloca el título en la parte inferior de la tabla, debajo del contenido de la tabla.

9.2. Table-layout

La propiedad table-layout se utiliza para controlar cómo se distribuye el espacio y se ajusta el ancho de las columnas en una tabla HTML. Esta propiedad determina si el navegador debe calcular automáticamente los tamaños de las columnas basándose en el contenido de las celdas o si debe utilizar un diseño fijo en el que se especifica explícitamente el ancho de las columnas. Cuenta con los siguientes valores:

1. auto (valor predeterminado): Cuando table-layout se establece en «auto», el navegador calculará automáticamente el ancho de las columnas en función del contenido de las celdas. Esto significa que las columnas se ajustarán automáticamente según el contenido.

2. **fixed:** Cuando table-layout se establece en «fixed», el ancho de las columnas se fija y se especifica explícitamente mediante CSS. Las columnas tendrán un ancho uniforme o el ancho que se especifique para cada columna, independientemente del contenido de las celdas. Esto puede ser útil cuando se desea un diseño de tabla con columnas de ancho uniforme.

9.3. Border-collapse

La propiedad border-collapse se utiliza en tablas HTML para controlar cómo se comportan los bordes de las celdas adyacentes en una tabla. Esta propiedad tiene dos valores:

1. **separate** (valor predeterminado): Los bordes de las celdas de la tabla se separan entre sí. Cada celda tiene sus propios bordes individuales, y no se combinan ni comparten con las celdas adyacentes. Esto significa que hay un espacio entre las celdas donde se encuentran sus bordes.
2. **collapse:** Los bordes de las celdas de la tabla se combinan y se presentan como un solo borde común entre las celdas adyacentes. En este caso, no hay espacio entre las celdas, y los bordes de las celdas se fusionan, creando un aspecto más compacto.

9.4. Border-spacing

La propiedad border-spacing se utiliza en tablas HTML para controlar la separación entre las celdas de la tabla cuando la propiedad border-collapse se establece en «**separate**» (que es su valor predeterminado). border-spacing permite definir el espacio horizontal y vertical entre las celdas adyacentes de una tabla, creando un espacio uniforme alrededor de cada celda.

La propiedad border-spacing se establece utilizando dos valores, separados por un espacio:

- El primer valor define la separación horizontal entre las celdas.
- El segundo valor define la separación vertical entre las celdas.

Ambos valores se pueden especificar en unidades de longitud (píxeles, em, rem, etc.) o como porcentaje del ancho del contenedor de la tabla.

9.5. Empty-cells

La propiedad empty-cells se utiliza para controlar la visualización de celdas vacías en una tabla HTML. Cuando aplicas esta propiedad a una tabla, puedes especificar si deseas mostrar o ocultar las celdas que no contienen contenido o están vacías.

La propiedad empty-cells admite dos valores:

1. **show** (valor predeterminado): Cuando se establece empty-cells: show, se muestra el contorno de las celdas vacías en la tabla. Esto significa que las celdas sin contenido visible, como espacios en blanco o celdas

completamente vacías, mostrarán los bordes y el fondo de la celda, incluso si no tienen contenido.

2. `hide`: Al establecer `empty-cells: hide`, se ocultan las celdas vacías en la tabla. Las celdas que no contienen contenido visible no mostrarán bordes ni fondo, lo que hace que las celdas vacías sean invisibles.

10. Unidades de medida en CSS

En CSS, hay varias unidades de medida que puedes utilizar para especificar tamaños y dimensiones de elementos en tu página web.

El tamaño de los elementos de una web se puede expresar en unidades **absolutas** y **relativas**.

10.1. Unidades absolutas

Las **unidades absolutas** mantienen su aspecto y se visualizan siempre igual independientemente de las características del dispositivo.

px	Píxeles
in	Pulgadas (1 pulgada = 2.54 cm)
cm	Centímetros
mm	Milímetros
pt	Puntos (1 pt = 1/72 pulgadas)
pc	Picas (1 pica = 12 puntos)

10.2. Unidades relativas

Las **unidades relativas** se ajustan a cada tipo de dispositivo ya que dependen de la resolución de cada pantalla.

em	Relativo al tamaño de la fuente del elemento (2 em significa 2 veces el tamaño de la fuente actual)
%	Porcentaje (relativo al elemento padre)
vh y vw	Medidas relativas de acuerdo al viewport 1vh = 1% de la altura del viewport 100vh = altura del viewport
fr	Flexible Grid Units (fr) Se utiliza en Grid Layout y representa una fracción del espacio disponible en un contenedor

10.3. Usos de unidades

Normalmente es **recomendable usar unidades relativas** en la medida de lo posible, ya que mejora la accesibilidad de la página web y permite que los documentos se adapten fácilmente a cualquier medio. Por tanto, para la creación de una página web, el uso de medidas absolutas queda descartado.

Centímetros (cm), milímetros (mm), pulgadas (in) y puntos (pt)

Unidades de medida físicas para impresión y otros usos específicos.

Unidad em

Es especialmente útil para establecer tamaños proporcionales al tamaño de fuente de un antecesor. Aunque no hay un criterio definido, el organismo W3C, recomienda el uso de la unidad em para indicar el tamaño del texto.

2 em = 2 x tamaño del padre

2em (Sin padre) = 2 x 16px(tamaño por defecto).

El uso de esta unidad puede ser problemático debido a su efecto compuesto de em anidados.

Existe también la unidad **REM (rem)** que es similar al em, pero se basa en el tamaño de fuente del elemento raíz (generalmente el tamaño de fuente del elemento HTML).

Píxeles (px)

Es la unidad más utilizada y representa un punto en la pantalla. Se usa para **tamaños fijos** y proporciona control preciso sobre el diseño. Aunque se ha mencionado que normalmente es recomendable utilizar unidades relativas en la medida de lo posible, el píxel sigue siendo una opción muy empleada en el diseño web, especialmente cuando se requiere un diseño más estático o se necesita un control exacto sobre el tamaño de los elementos.

Porcentaje (%)

Representa una proporción del tamaño del elemento padre. Es útil para hacer diseños fluidos y responsivos teniendo en cuenta la **relación de los elementos con su contenedor padre**.

Viewport Width (vw) y Viewport Height (vh)

Representan un porcentaje del ancho y alto de la ventana del navegador, respectivamente. Son útiles para crear **diseños responsive basados en el tamaño de la pantalla**.

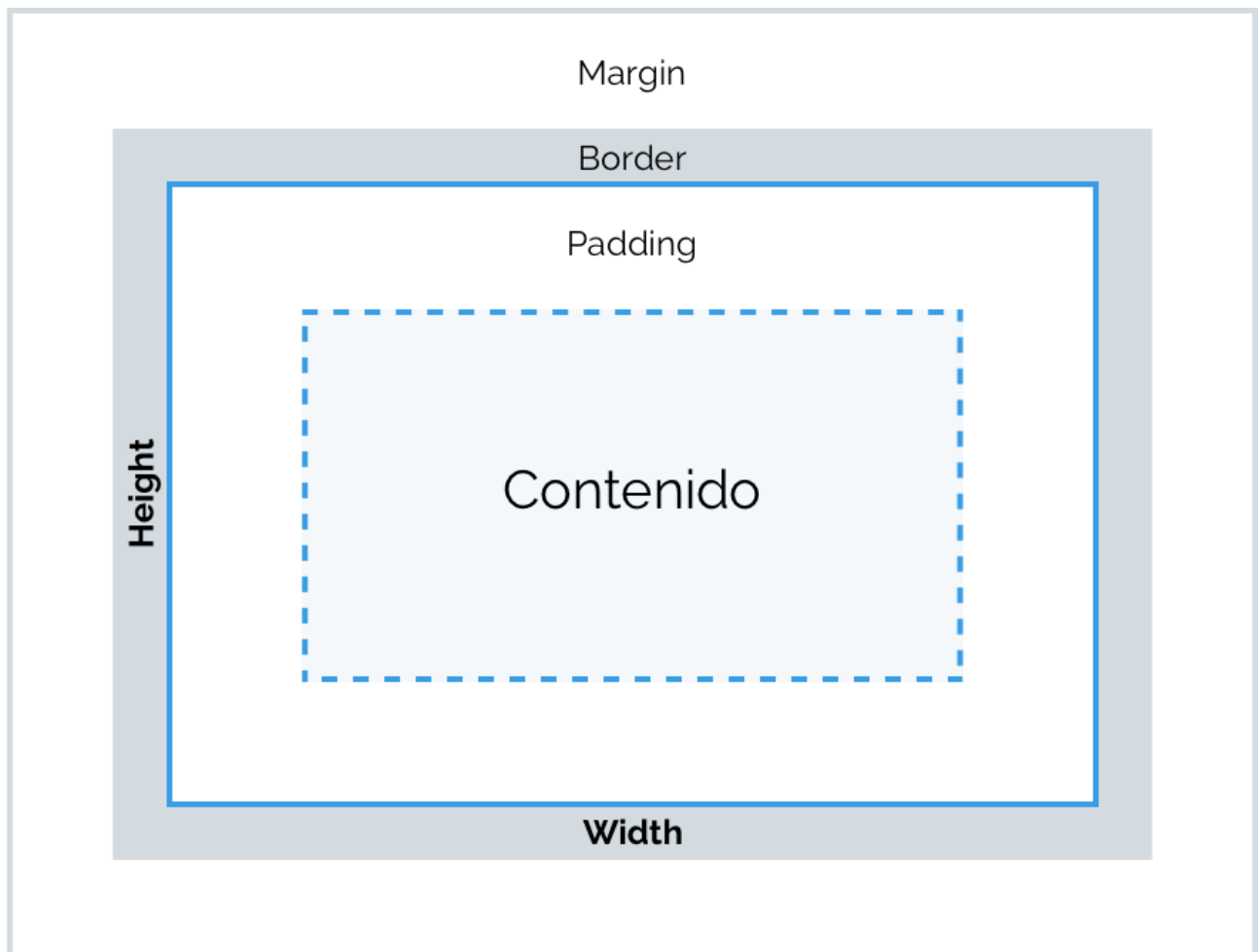
Flexible Grid Units (fr)

Se utiliza en Grid Layout y representa una fracción del espacio disponible en un contenedor. Es útil para distribuir el espacio disponible entre elementos flexibles.

11. Modelo de cajas: márgenes, relleno y bordes

Cualquier elemento incluido en un documento HTML dispone de una estructura tipo caja que se puede modificar usando las propiedades CSS. Las propiedades más importantes de las cajas o contenedores son las siguientes: margin (margen externo), border (borde) y padding (margen interno).

Las propiedades operan en el siguiente orden: **superior, derecha, inferior e izquierda**.



11.1. Padding o relleno

El **padding** es el **margen interno** de un elemento, también se le llama relleno y es la cantidad de espacio entre el borde y el contenido del elemento. Veamos una tabla con las propiedades para dar un margen interno a un elemento.

Propiedad	Descripción	Valores
padding-top padding-right padding-bottom padding-left	Tamaño del relleno superior, derecho, inferior e izquierdo	longitud porcentaje
padding	Tamaño del relleno	longitud porcentaje {1,4}

Valores del padding (también aplicables a la propiedad margin):

- **Un valor:** se aplica el estilo a los **4 lados**.
- **Dos valores:** el primer valor se aplica a **arriba y abajo**, el segundo valor se aplica a **derecha e izquierda**.
- **Tres valores:** el primer valor se aplica a **arriba**, el segundo valor a **derecha e izquierda** y el tercer valor se aplica a **abajo** del elemento.
- **Cuatro valores:** el primer valor se aplica a **arriba**, el segundo valor se aplica a la **derecha**, el tercer valor se aplica a **abajo** y el cuarto valor se aplica a la **izquierda**.

11.2. Margin

El **margin** es el **margen externo** de un elemento, fuera de cualquier borde definido. Veamos las distintas propiedades para dar estilo a los márgenes de un elemento.

Propiedad	Descripción	Valores
margin-top margin-right margin-bottom margin-left	Tamaño del margen superior, derecho, inferior e izquierdo	Longitud porcentaje auto
margin	Ancho de los márgenes	Longitud porcentaje auto {1,4}

11.3. Bordes

La propiedad ***border*** en CSS permite especificar el estilo, el ancho y el color de los bordes de un elemento. Puedes usar diferentes valores para crear distintos tipos de bordes, como líneas lisas, de puntos, redondeados, etc. Veamos las propiedades para dar estilo a los bordes de un elemento.

Propiedad	Descripción	Valores
border-top-width border-right-width border-bottom-width border-left-width	Anchura del borde superior, derecho, inferior o izquierdo	thin medium thick longitud
border-width	Anchura del borde (reducida)	thin medium thick longitud {1,4}
border-top-color border-right-color border-bottom-color border-left-color	Color del borde superior, derecho, inferior e izquierdo	color transparent
border-color	Color del borde (reducida)	color transparent {1,4}
border-top-style border-right-style border-bottom-style border-left-style	Estilo del borde superior, derecho, inferior e izquierdo	none hidden dotted dashed solid double groove ridge inset outset
border-style	Estilo del borde (reducida)	none hidden dotted dashed solid double groove ridge inset outset {1,4}
border-top border-right border-bottom border-left	Ancho, estilo y color para el borde superior, derecho, inferior e izquierdo	border-top-width border-top-style border-top-color
border	Ancho, estilo y color para los bordes (reducida)	border-width border-style border-color
border-radius	Curvatura del borde	longitud porcentaje {1,4}

***Las propiedades del borde no tendrán efecto hasta que se defina la propiedad **border-style**.*

12.Propiedades width, height, max-width, min-width, max-height y min-height

Nombre propiedad	Descripción	Valores
width	Establece el ancho del área de contenido de un elemento	Unidad de longitud (px, em, %, etc.), auto, initial, inherit
height	Establece el alto del área de contenido de un elemento	Unidad de longitud (px, em, %, etc.), auto, initial, inherit
max-width	Establece el ancho máximo que puede tener un elemento	Unidad de longitud (px, em, %, etc.), none, initial, inherit
min-width	Establece el ancho mínimo que debe tener un elemento	Unidad de longitud (px, em, %, etc.), 0, initial, inherit
max-height	Establecer el alto máximo que debe tener un elemento	Unidad de longitud (px, em, %, etc.), 0, initial, inherit
min-height	Establecer el alto mínimo que debe tener un elemento	Unidad de longitud (px, em, %, etc.), 0, initial, inherit

Las propiedades CSS **width** y **height** se usan para establecer el ancho y el alto de un elemento. Por defecto, estas propiedades se refieren al área de contenido del elemento, es decir, el espacio que ocupa el texto, las imágenes u otros elementos dentro del elemento. El área de contenido no incluye el padding (relleno), el border (borde) ni el margin (margen) del elemento.

Sin embargo, podemos cambiar este comportamiento con la propiedad CSS **box-sizing**, que nos permite definir qué partes del elemento se incluyen en el cálculo del ancho y el alto

Las propiedades CSS **max-width** y **max-height** se usan para establecer el ancho y el alto máximo de un elemento. Estas propiedades nos permiten limitar el tamaño de un elemento para que no supere cierto valor, incluso si le damos un ancho o un alto mayor con las propiedades **width** o **height**.

Esto puede ser muy útil para crear diseños responsivos que se adapten a diferentes tamaños de pantalla. Por ejemplo, podemos usar **max-width** para evitar que una imagen se salga del contenedor o que se distorsione al cambiar la resolución.

También podemos usar **max-height** para controlar la altura de un elemento que tenga contenido variable, como un menú desplegable o una caja de texto.

Las propiedades CSS **min-width** y **min-height** se utilizan para establecer la anchura y la altura mínima de un elemento, respectivamente. Estas propiedades impiden que el elemento se reduzca a un ancho o alto menor que el valor especificado.

Las propiedades min-width y min-height se pueden utilizar para garantizar que un elemento tenga un tamaño mínimo, independientemente del tamaño del contenido que contenga. Por ejemplo, se pueden utilizar para garantizar que un elemento siempre tenga un tamaño mínimo de visualización, incluso en pantallas pequeñas.

13.Posición y comportamiento de contenedores en CSS

En el proceso de creación de una web es imprescindible organizar elementos como imágenes, textos o tablas. Para ello, necesitaremos conocer los elementos de ordenación y las propiedades que nos ayudan a organizar todos los componentes. Las propiedades más importantes se definen en la siguiente tabla y se explican a continuación.

Propiedad	Descripción	Valores
display	Comportamiento del contenedor	inline block inline-block none
position	Esquema de posicionamiento	static relative absolute fixed sticky
top right bottom left	Desplazamiento de la caja respecto al borde superior, derecho, inferior o izquierdo	longitud porcentaje auto
float	Posicionamiento flotante	left right none
clear	Control de cajas adyacentes a las float	none left right both
z-index	Nivel de la capa	auto número entero
box-sizing	Control de bordes y relleno en el comportamiento del contenedor	content-box border-box
visibility	Muestra u oculta un elemento ocupando el espacio	visible hidden

13.1. Display

Valores: **none** | **inline** | **block** | **inline-block**

- **none:** los elementos se ocultan y no se muestra el espacio reservado.
- **inline:** los elementos se muestran en la misma línea (respetando el flujo) y no se aceptan las propiedades width, height ni márgenes verticales.
- **block:** los elementos se muestran en líneas independientes y se aceptan las propiedades width, height y márgenes verticales.

- **inline-block:** su comportamiento es una mezcla entre los dos anteriores, los elementos se muestran en la misma línea (respetando el flujo) y se aceptan las propiedades width, height y márgenes verticales.

13.2. Position

Valores: **static** | **relative** | **absolute** | **fixed**

- **static:** los elementos se posicionan de acuerdo al flujo normal de la página. Es la **posición natural de los elementos**. No son afectados por las propiedades top, bottom, left y right.
- **relative:** los elementos se posicionan de forma relativa a su posición normal.
- **fixed:** los elementos se posicionan de forma relativa a la ventana del navegador. Su posición permanece fija aunque se desplace la ventana.
- **absolute:** los elementos se posicionan de forma relativa al primer elemento padre que tenga una posición distinta a static.
- **sticky:** los elementos son posicionados de forma relativa hasta que su bloque contenedor alcanza un límite establecido.

13.3. Float

Valores: **left** | **right** | **none**

Cuando a un elemento HTML se le aplica un estilo con la propiedad float, el elemento sale del flujo normal y aparece posicionado a la izquierda o a la derecha de su contenedor, donde el resto de elementos de la página se posicionarán alrededor.

13.4. Clear

Valores: **none** | **left** | **right** | **both**

La propiedad clear establece si un elemento debe estar al lado de los elementos flotantes que lo preceden o si debe situarse bajo de ellos. Se suele utilizar para restaurar el flujo normal del documento y así los elementos dejan de flotar hacia la izquierda, la derecha o ambos lados.

13.5. Z-index

Valores: **auto** | **número entero**

Mediante el atributo z-index podemos organizar cada uno de los elementos del contenido de una página web. Como se puede apreciar en la imagen, cuando varios elementos se superponen, **los elementos con mayor valor z-index se sitúan por encima de los que tienen menor valor**.

13.6. Box-sizing

Por defecto en el modelo de cajas de CSS, el ancho y alto asignado a un elemento es aplicado solo al contenido de la caja del elemento. Si el elemento tiene algún borde (border) o relleno (padding), este es entonces añadido al ancho y alto del tamaño de la caja o contenedor. Esto significa que cuando se define el ancho y alto, se tiene que ajustar el valor para permitir cualquier borde o relleno que se pueda añadir.

Valores: **content-box** | **border-box**

- **content-box** es el comportamiento CSS por defecto para el tamaño de la caja (box-sizing). Si se define el ancho de un elemento en 100 píxeles, la caja del contenido del elemento tendrá 100 píxeles de ancho, y el ancho de cualquier borde o relleno será añadido al ancho final desplegado.
- **border-box** toma en cuenta cualquier valor que se especifique de borde o de relleno para el ancho o alto de un elemento. Es decir, si se define un elemento con un ancho de 100 píxeles. Esos 100 píxeles incluirán cualquier borde o relleno que se añada, y la caja de contenido se encogerá para absorber ese ancho extra. Esta propiedad es especialmente útil para redimensionar cualquier elemento.

13.7. Visibility

La propiedad visibility indica si un elemento es visible o permanece oculto (ocupando el mismo espacio).

Valores: **visible** | **hidden**