

24 CSS: Cajas de Línea (Inline) VS Cajas de Bloque (Block)

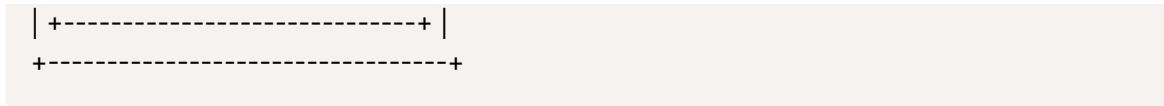
Cada elemento HTML tiene un valor por defecto para la propiedad `display` que determina su comportamiento fundamental en el flujo del documento y cómo interactúa con el Modelo de Caja (width, height, margin, padding). Los dos valores de `display` más fundamentales y comunes son `block` y `inline`.

Elementos de Nivel de Bloque (`display: block`)

Son los elementos que forman la estructura principal de una página web.

- **Comportamiento en el Flujo:**
 - Siempre comienzan en una **nueva línea**.
 - Apilan verticalmente uno debajo del otro por defecto.
- **Dimensiones (`width` y `height`):**
 - Por defecto (`width: auto`), ocupan **todo el ancho disponible** de su bloque contenedor.
 - **Respetan** las propiedades `width` y `height` que les establezcas explícitamente. Puedes controlar sus dimensiones.
- **Márgenes y Rellenos (`margin` y `padding`):**
 - Respetan **todos** los márgenes y rellenos: `margin-top` , `margin-right` , `margin-bottom` , `margin-left` , `padding-top` , `padding-right` , `padding-bottom` , `padding-left` .
 - Los márgenes verticales (`margin-top` , `margin-bottom`) empujan a otros elementos de bloque hacia arriba o hacia abajo y pueden colapsar entre sí.
- **Ejemplos Comunes (por defecto `display: block`):**
 - `<div>`
 - `<p>`
 - `<h1>` , `<h2>` , `<h3>` , `<h4>` , `<h5>` , `<h6>`
 - `` , `` , ``
 - `<form>`
 - `<header>` , `<footer>` , `<nav>` , `<section>` , `<article>` , `<aside>` (elementos semánticos de HTML5)
 - `<table>`
- **Visualización:**

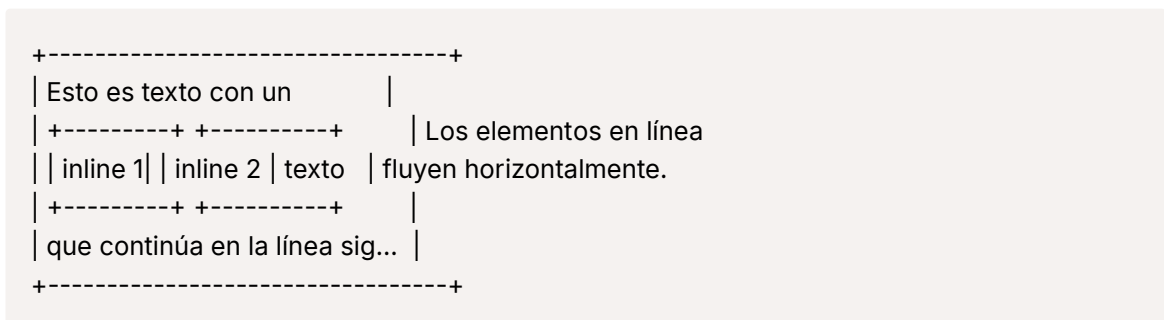
```
+-----+ ← Contenedor
| +-----+ |
| | Elemento de Bloque 1 | | Ocupa todo el ancho
| +-----+ |
| +-----+ |
| | Elemento de Bloque 2 | | Empieza en nueva línea
```



Elementos de Nivel de Línea (`display: inline`)

Son los elementos que normalmente fluyen dentro de un bloque de texto, sin causar saltos de línea.

- **Comportamiento en el Flujo:**
 - **No** comienzan en una nueva línea.
 - Se colocan uno al lado del otro horizontalmente, dentro de su bloque contenedor, formando líneas. Si no caben en una línea, continúan en la siguiente.
- **Dimensiones (`width` y `height`):**
 - Su ancho (`width`) se ajusta automáticamente al **tamaño de su contenido**.
 - **Ignoran** cualquier valor explícito que les asignes a `width` y `height` . No puedes establecer sus dimensiones directamente.
- **Márgenes y Rellenos (`margin` y `padding`):**
 - **Respetan** los márgenes y rellenos **horizontales**: `margin-left` , `margin-right` , `padding-left` , `padding-right` . Estos sí crean espacio horizontalmente.
 - ¡**importante!** Aunque `margin-top` , `margin-bottom` , `padding-top` , y `padding-bottom` **pueden aplicarse visualmente** (por ejemplo, el fondo se extenderá al padding vertical), **generalmente NO afectan el espacio vertical entre las líneas de texto ni empujan a otros elementos de bloque o en línea verticalmente**. Es decir, no alteran significativamente el layout vertical.
- **Ejemplos Comunes (por defecto `display: inline`):**
 - ``
 - `<a>` (enlace)
 - `` , ``
 - `
` (salto de línea, caso especial)
 - `<label>`
 - `<q>` (cita corta)
 - `<code>`
- **Visualización:**



3. Caso Especial: Elementos Reemplazados y `inline-block`

Existe una categoría intermedia importante:

- **Elementos Reemplazados (Replaced Elements):** Son elementos cuyo contenido no está definido por el propio documento HTML/CSS, sino por un recurso externo. El ejemplo más claro es ``. Otros incluyen `<input>`, `<textarea>`, `<select>`, `<button>`, `<video>`.
 - Estos elementos, aunque a menudo tienen un `display` por defecto de `inline` (como ``, `input`), se comportan de manera diferente: **Sí respetan `width` y `height`**, y los márgenes/paddings verticales sí suelen tener efecto en el layout vertical. Su comportamiento se asemeja mucho a `inline-block`.
- **`display: inline-block`**: Este valor combina características de ambos mundos:
 - **Flujo:** El elemento fluye horizontalmente con otros elementos en línea (no empieza en una nueva línea).
 - **Modelo de Caja: Respetar `width`, `height`**, y **todos** los márgenes y rellenos (verticales y horizontales), como un elemento de bloque.
 - **Uso:** Muy útil para colocar elementos (como botones, iconos con texto, pequeñas tarjetas) uno al lado del otro, pero pudiendo controlar su tamaño y espaciado vertical de forma precisa. Muchos elementos de formulario (`button`, `input`, `select`, `textarea`) se comportan como `inline-block` por defecto o pueden ser estilizados fácilmente para comportarse así.

Tabla Comparativa Rápida (Comportamiento por Defecto)

Característica	<code>display: block</code>	<code>display: inline</code>	<code>display: inline-block</code> (o similar)
Flujo	Empieza en nueva línea	Fluye con el texto	Fluye con el texto
<code>width</code> por defecto	Ocupa ancho completo del contenedor	Se ajusta al contenido	Se ajusta al contenido
¿Respetar <code>width</code>?	Sí	No	Sí
¿Respetar <code>height</code>?	Sí	No	Sí
¿Respetar <code>margin</code> Vert?	Sí	No (no afecta layout vertical)	Sí
¿Respetar <code>padding</code> Vert?	Sí	No (no afecta layout vertical)	Sí
¿Respetar Horiz M/P?	Sí	Sí	Sí

Entender la diferencia entre elementos de bloque y en línea es fundamental para predecir cómo se comportarán los elementos en tu página. Los elementos de bloque estructuran la página verticalmente y ocupan todo el ancho, mientras que los elementos en línea fluyen horizontalmente dentro del texto y tienen limitaciones en cómo aplican dimensiones y márgenes/paddings verticales. Los elementos `inline-block` (y los reemplazados) ofrecen un útil punto intermedio. Recuerda que estos son los comportamientos *por defecto*, y la propiedad `display` (que veremos a continuación) te permite cambiar explícitamente el tipo de caja de cualquier elemento.