

## Elementos en bloque, elementos en línea y display

### Elementos en bloque (block)

- Ocupan todo el ancho disponible
- Que ocupen todo el ancho disponible no significa que ocupen toda la pantalla, por el contrario ocupan el ancho total o disponible del elemento padre.
- Siempre comienzan en una nueva línea
- Se usan para estructurar la página

#### Ejemplos comunes:

<div>: contenedor genérico

<p>: párrafo

<h1>, <h2>, ..., <h6>: encabezados

<ul>, <ol>: listas

<li>: ítems de lista

<section>, <article>, <aside>, <header>, <footer>: elementos semánticos

<form>, <table>, <hr>, <nav>

### Elementos en línea (inline)

- Solo ocupan el ancho necesario para su contenido
- No comienzan en una nueva línea
- Se usan dentro de los elementos de bloque

#### Ejemplos comunes:

<span>: contenedor en línea genérico

<a>: enlaces

<strong>, <b>: texto en negrita

<em>, <i>: texto en cursiva

<img>: imágenes

<input>, <label>, <select>, <textarea> (algunos se comportan como inline o inline-block)

### Bonus: inline-block

Se comportan como en línea, pero pueden tener ancho, alto, márgenes como los de bloque.

Ejemplo: <button> o elementos con display: inline-block; en CSS.

### Flexbox (display: flex)

- Sistema de distribución en una dimensión (fila o columna)
- Se usa en un contenedor padre, y afecta a sus hijos

#### ¿Para qué sirve?

- Para alinear elementos de forma responsiva
- Distribuir espacios fácilmente
- Centrar cosas vertical u horizontalmente

### Grid (display: grid)

- Sistema de distribución en dos dimensiones (filas y columnas)
- Muy útil para layouts complejos

### ¿Para qué sirve?

- Para diseñar rejillas tipo tabla
- Ideal para galerías, dashboards, estructuras más completas

### **display: none**

- Elimina completamente el elemento de la página: no se ve y no ocupa espacio.
- Es como si no existiera en el HTML para el navegador.

### ¿Para qué sirve?

- Para ocultar algo totalmente (por ejemplo, un menú que se muestra al hacer clic).
- Muy usado en JS para mostrar/ocultar dinámicamente.

### **visibility: hidden**

- El elemento sigue ocupando espacio, pero no se ve.
- Es invisible, pero el navegador lo reserva en el layout.

### ¿Para qué sirve?

- Cuando quieres ocultar algo, pero mantener el lugar reservado.
- Útil para interfaces donde no querés que se muevan los elementos al ocultar algo.