

DISPLAY

El atributo CSS `display` es una de las propiedades más importantes y versátiles del CSS. Define cómo un elemento HTML se debe mostrar en la página web. A continuación te explico los valores más comunes del atributo `display`, junto con ejemplos prácticos:

Valores Comunes de `display`

1. `block`
2. `inline`
3. `inline-block`
4. `flex`
5. `grid`
6. `none`

1. `display: block`

Los elementos de tipo bloque (block) ocupan todo el ancho disponible de su contenedor, comenzando en una nueva línea. Ejemplos de elementos de bloque son `<div>`, `<p>`, `<h1>`, etc.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Display Block</title>
  <style>
    .block {
      display: block;
      width: 100%;
      background-color: lightblue;
      margin-bottom: 10px;
    }
  </style>
</head>
<body>
  <div class="block">Soy un elemento de bloque</div>
  <div class="block">Ocupo todo el ancho disponible</div>
</body>
</html>
```

2. display: inline

Los elementos de tipo en línea (inline) no comienzan en una nueva línea y solo ocupan el ancho necesario. Ejemplos de elementos en línea son ``, `<a>`, ``, etc.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Display Inline</title>
  <style>
    .inline {
      display: inline;
      background-color: lightcoral;
      padding: 5px;
    }
  </style>
</head>
<body>
  <span class="inline">Soy un</span>
  <span class="inline">elemento en línea</span>
</body>
</html>
```

3. display: inline-block

Los elementos **inline-block** son similares a los elementos en línea, pero se comportan como bloques, es decir, respetan las propiedades de altura y anchura y permiten márgenes y rellenos.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Display Inline-Block</title>
  <style>
    .inline-block {
      display: inline-block;
      width: 100px;
      height: 100px;
      background-color: lightgreen;
      margin: 5px;
    }
  </style>
</head>
<body>
  <div class="inline-block">Elemento 1</div>
  <div class="inline-block">Elemento 2</div>
  <div class="inline-block">Elemento 3</div>
</body>
</html>
```

4. display: flex

La propiedad **flex** se usa para diseñar contenedores flexibles y sus elementos hijos. Es una manera eficiente de distribuir el espacio dentro de un contenedor y alinear los elementos.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Display Flex</title>
  <style>
    .flex-container {
      display: flex;
      justify-content: space-around;
      background-color: lightgray;
      padding: 10px;
    }
    .flex-item {
      background-color: lightblue;
      padding: 20px;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">Elemento 1</div>
    <div class="flex-item">Elemento 2</div>
    <div class="flex-item">Elemento 3</div>
  </div>
</body>
</html>
```

5. display: grid

La propiedad **grid** permite crear una cuadrícula para organizar los elementos en filas y columnas. Es una herramienta poderosa para diseños complejos.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Display Grid</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: auto auto auto;
      gap: 10px;
      background-color: lightgray;
      padding: 10px;
    }
    .grid-item {
      background-color: lightpink;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">Elemento 1</div>
    <div class="grid-item">Elemento 2</div>
    <div class="grid-item">Elemento 3</div>
    <div class="grid-item">Elemento 4</div>
    <div class="grid-item">Elemento 5</div>
    <div class="grid-item">Elemento 6</div>
  </div>
</body>
</html>
```

6. display: none

La propiedad **none** oculta el elemento, lo elimina del flujo del documento, y no ocupa espacio.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Display None</title>
  <style>
    .hidden {
      display: none;
    }
  </style>
</head>
<body>
  <p>Este párrafo es visible.</p>
  <p class="hidden">Este párrafo está oculto y no ocupa espacio.</p>
</body>
</html>
```