

1. EJERCICIO CON DISPLAY: BLOCK

Descripción: Los elementos con display: block ocupan todo el ancho disponible y comienzan en una nueva línea.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio display: block</title>
  <style>
    .block {
      display: block;
      background-color: lightblue;
      margin: 10px 0;
      padding: 10px;
      width: 100%;
    }
  </style>
</head>
<body>
  <h2>Ejemplo de display: block</h2>
  <div class="block">Elemento 1</div>
  <div class="block">Elemento 2</div>
  <div class="block">Elemento 3</div>
</body>
</html>
```

2. EJERCICIO CON DISPLAY: INLINE

Descripción: Los elementos con display: inline no inician en una nueva línea y solo ocupan el espacio necesario.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio display: inline</title>
  <style>
    .inline {
      display: inline;
      background-color: lightgreen;
      padding: 5px;
    }
  </style>
</head>
<body>
  <h2>Ejemplo de display: inline</h2>
  <span class="inline">Elemento 1</span>
  <span class="inline">Elemento 2</span>
  <span class="inline">Elemento 3</span>
</body>
</html>
```

3. EJERCICIO CON DISPLAY: INLINE-BLOCK

Descripción: Los elementos con display: inline-block se comportan como elementos en línea pero permiten ajustar su tamaño y margen.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio display: inline-block</title>
  <style>
    .inline-block {
      display: inline-block;
      background-color: lightcoral;
      padding: 10px;
      margin: 5px;
      width: 100px;
      text-align: center;
    }
  </style>
</head>
<body>
  <h2>Ejemplo de display: inline-block</h2>
  <div class="inline-block">1</div>
  <div class="inline-block">2</div>
  <div class="inline-block">3</div>
</body>
</html>
```

4. EJERCICIO CON DISPLAY: FLEX

Descripción: Los elementos con display: flex permiten organizar contenido en filas o columnas con alineación dinámica.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio display: flex</title>
  <style>
    .flex-container {
      display: flex;
      justify-content: space-around;
      align-items: center;
      background-color: lightyellow;
      padding: 20px;
    }
    .flex-item {
      background-color: lightblue;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <h2>Ejemplo de display: flex</h2>
  <div class="flex-container">
    <div class="flex-item">A</div>
    <div class="flex-item">B</div>
    <div class="flex-item">C</div>
  </div>
</body>
</html>
```

5. EJERCICIO CON DISPLAY: GRID

Descripción: Los elementos con display: grid permiten organizarse en una cuadrícula.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio display: grid</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      gap: 10px;
      background-color: lightgray;
      padding: 10px;
    }
    .grid-item {
      background-color: lightpink;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <h2>Ejemplo de display: grid</h2>
  <div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
  </div>
</body>
</html>
```

6. EJERCICIO CON DISPLAY: NONE

Descripción: Los elementos con display: none no se muestran en la página y no ocupan espacio.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio display: none</title>
  <style>
    .hidden {
      display: none;
    }
  </style>
</head>
<body>
  <h2>Ejemplo de display: none</h2>
  <div class="hidden">No se mostrará este contenido.</div>
  <p>Este contenido es visible.</p>
</body>
</html>
```

EJEMPLO DE USO BÁSICO:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo display: none</title>
  <style>
    .oculto {
      display: none;
    }
  </style>
</head>
<body>
  <h2>Ejemplo de display: none</h2>
  <p>Este texto es visible.</p>
  <div class="oculto">Este contenido está oculto.</div>
  <button onclick="mostrar()">Mostrar contenido oculto</button>

  <script>
```

```
function mostrar() {  
    document.querySelector('.oculto').style.display = 'block';  
}  
</script>  
</body>  
</html>
```

DIFERENCIA ENTRE DISPLAY: NONE Y VISIBILITY: HIDDEN

- **display: none:** Oculta el elemento completamente. No se renderiza en el flujo del diseño y no ocupa espacio.
- **visibility: hidden:** Oculta el elemento, pero este sigue ocupando espacio en el diseño.

Ejemplo comparativo:

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>display: none vs visibility: hidden</title>  
    <style>  
        .none {  
            display: none;  
        }  
        .hidden {  
            visibility: hidden;  
        }  
    </style>  
</head>  
<body>  
    <h2>Comparación</h2>  
    <p class="none">Este texto no se ve y no ocupa espacio.</p>  
    <p class="hidden">Este texto no se ve pero sí ocupa espacio.</p>  
    <p>Este es un texto visible.</p>  
</body>  
</html>
```

Observa la diferencia: el texto con `visibility: hidden` crea un "hueco" en el diseño, mientras que el texto con `display: none` desaparece completamente del flujo.