Espacio entre elementos inline-block en HTML y CSS

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Nov - Dic
- iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice

OBJETIVO

- Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.
- Utilizar marcos y tablas para presentar la información de manera ordenada.
- Identificar nuevos elementos y etiquetas en HTML5.
- Reconocer las posibilidades de modificar etiquetas HTML.
- Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.

GLOSARIO

Formularios. Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

Fuentes seguras. Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

Guías de estilo. Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

HTML. Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

GLOSARIO

HTML5. Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

Legibilidad. Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

Marcos. Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

Tipografía. Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa ”modo o estilo en que está impreso un texto” o ”clase de tipos de imprenta”.

INTRODUCCIÓN

Cuando trabajamos con elementos en línea-bloque `display: inline-block;` en HTML y CSS, es posible que te encuentres con el **problema del espacio no deseado** entre estos elementos.

A continuación, se explora el problema y presentan algunas soluciones con ejemplos prácticos. Sin embargo, cabe destacar que lo ideal sería optar por utilizar alternativas como **Flexbox** o **CSS Grid**, técnicas que veremos en próximas unidades.

Descripción del problema con el espacio entre elementos en línea-bloque

Los elementos en línea-bloque, como divs con la propiedad `display: inline-block;`, pueden presentar un espacio adicional entre ellos. Este espacio puede causar que los elementos se visualicen en líneas separadas en lugar de estar uno al lado del otro.

Por ejemplo:

```
<div class="container">
  <!-- Contenido del primer contenedor -->
</div>
<div class="container">
  <!-- Contenido del segundo contenedor -->
</div>
<div class="container">
  <!-- Contenido del tercer contenedor -->
</div>
```

Solución con comentarios en el código HTML (No recomendada)

Una solución es utilizar comentarios HTML para eliminar el espacio que se incluye por defecto entre los elementos. Esto se logra insertando comentarios entre los elementos div. **Esta solución complica el código, por lo que es mejor optar por otras alternativas.**

Código HTML:

```
<div class="container">1</div><!--
--><div class="container">2</div><!--
--><div class="container">3</div>
<p>Quita los comentarios para
visualizar el problema</p>
```

Código CSS:

```
.container {
  display: inline-block;
  width: 33.33%;
  box-sizing: border-box;
  border: 1px solid black;
  font-size: 16px;
}
```

Solución con comentarios en el código HTML (No recomendada)

- Salida con problemática

1	2
3	

Quita los comentarios para visualizar el problema

- Salida sin problemática

1	2	3

Quita los comentarios para visualizar el problema

Solución modificando el tamaño de fuente en el CSS (No recomendada)

Otra alternativa es ajustar el tamaño de fuente de `html` y `body` a cero para eliminar el espacio en blanco, y luego restaurar el tamaño de fuente deseado para el contenido dentro de los contenedores.

Código HTML:

```html
<div class="container">1</div>
<div class="container">2</div>
<div class="container">3</div>
<p>Elimina el font-size:0 del CSS
para visualizar el problema</p>

/* font-size:0 establece el tamaño de
fuente a cero para eliminar el espacio
entre elementos inline-block */
```

Código CSS:

```css
html, body {
    font-size: 0;
    }
.container {
  display: inline-block;
  width: 33.33%;
  box-sizing: border-box;
  border: 1px solid black;
  font-size: 16px;
}
```

Solución modificando el tamaño de fuente en el CSS (No recomendada)

- Salida con problemática

1	2
3	

Elimina el font-size:0 del CSS para visualizar el problema

- Salida sin problemática

1	2	3

Solución al problema usando flotantes

Otra alternativa es utilizar flotantes para la alineación de los elementos.

Código HTML:

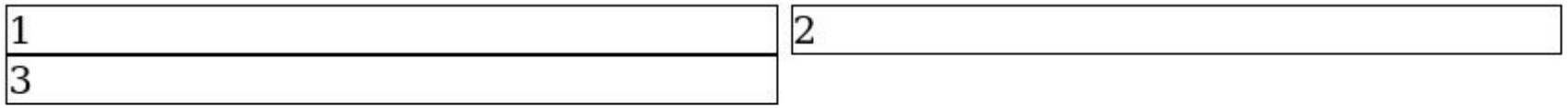
```html
<div class="container">1</div>
<div class="container">2</div>
<div class="container">3</div>
```

Código CSS:

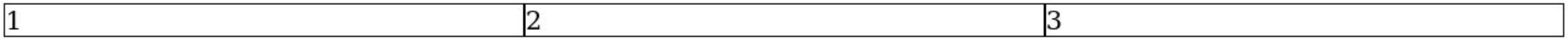
```css
.container {
  float: left;
  width: 33.33%;
  box-sizing: border-box;
  border: 1px solid black;
  font-size: 16px;
}
```

Solución al problema usando flotantes

- Salida con problemática



- Salida sin problemática



Solución al problema usando *Flexbox*

Otra alternativa es utilizar Flexbox para la alineación de los elementos.

Código HTML:

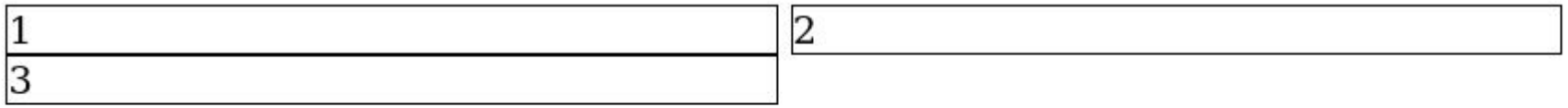
```html
<div class="contenedor-principal">
  <div class="container">1</div>
  <div class="container">2</div>
  <div class="container">3</div>
</div>
```

Código CSS:

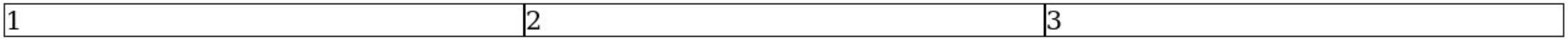
```css
.contenedor-principal {
  display: flex;
}
.container {
  width: 33.33%;
  box-sizing: border-box;
  border: 1px solid black;
  font-size: 16px;
}
```

Solución al problema usando *Flexbox*

- Salida con problemática



- Salida sin problemática



Solución al problema usando *CSS Grid*

Otra alternativa es utilizar CSS Grid o Grid Layout para la alineación de los elementos.

Código HTML:

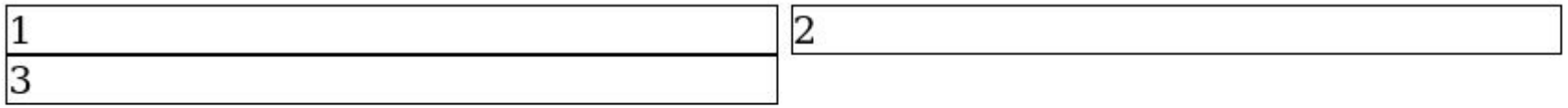
```html
<div class="contenedor-principal">
  <div class="container">1</div>
  <div class="container">2</div>
  <div class="container">3</div>
</div>
```

Código CSS:

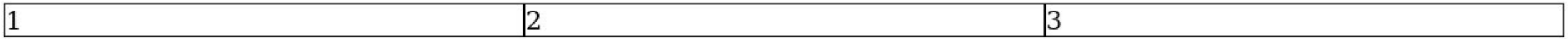
```css
.contenedor-principal {
  display: flex;
}

.container {
  flex-grow: 1;
  box-sizing: border-box;
  border: 1px solid black;
}
```

Solución al problema usando *Flexbox*

- Salida con problemática



- Salida sin problemática



¿Mejor solución al espacio adicional generado en los elementos con *display: inline-block*?

Cuando nos enfrentamos al espacio adicional generado por el uso de `display: inline-block;` en elementos HTML, es crucial adoptar enfoques que mantengan la estética del diseño sin comprometer la estructura del código.

De forma general, lo ideal será descartar el uso de comentarios o el ajuste del tamaño de fuente y optar por alternativas como el uso de Flexbox o CSS Grid.

Asegúrate de que la solución elegida sea compatible con los navegadores que necesitas soportar. En proyectos modernos, la compatibilidad con versiones antiguas puede ser menos crítica.

Como ves, seleccionar la mejor forma de abordar el espacio adicional con `display: inline-block;` depende del contexto del proyecto y de las necesidades específicas de compatibilidad y mantenimiento del código.