

Tema 4

CSS en XML

Tema 4

CSS en XML

Tema 4

1.-

CSS en XML

- Para asociar un archivo CSS (Cascading Style Sheets, Hojas de estilo en cascada) a un documento XML, se utiliza la declaración de procesamiento llamada xml-stylesheet.
- Esta declaración se coloca en la parte superior del documento XML, después de la declaración XML

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <?xml-stylesheet type="text/css" href="estilos.css"?>
3. <root>
4.   <!-- Contenido del documento XML -->
5. </root>
```

CSS en XML

- La declaración `xml-stylesheet` para indicar que se debe aplicar un archivo CSS al documento XML.
- La declaración tiene dos atributos:
 - **type**: Especifica el tipo de archivo de estilo, que en este caso es «text/css» para un archivo CSS.
 - **href**: Indica la ubicación del archivo CSS que se debe aplicar al documento XML. En este ejemplo, el archivo CSS se encuentra en el mismo directorio que el documento XML y se llama «estilos.css».

CSS en XML

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <?xml-stylesheet type="text/css" href="estilos.css"?>
3. <root>
4.   <element attribute="value">Text Content</element>
5. </root>
```

```
1. /* Estilos CSS */
2. root {
3.   font-family: Arial, sans-serif;
4.   color: red;
5. }
6. element {
7.   background-color: #f0f0f0;
8.   padding: 10px;
9.   border: 1px solid #ccc;
10. }
```

Ventajas

- Separación de preocupaciones:
 - CSS permite separar el estilo visual de la estructura y contenido del documento XML. Esto facilita la mantenibilidad del código, ya que los cambios en la apariencia no afectarán la estructura ni el contenido del documento.
- Consistencia visual:
 - Con CSS, se pueden definir estilos consistentes en todo el documento XML. Esto asegura que todos los elementos se vean de la misma manera, lo que mejora la experiencia del usuario y brinda una apariencia coherente.
- Flexibilidad y personalización:
 - CSS ofrece una amplia gama de propiedades y selectores que permiten personalizar la apariencia de los elementos XML. Esto brinda flexibilidad para adaptar el diseño a diferentes dispositivos, tamaños de pantalla o preferencias de los usuarios.

Ventajas

- Mantenibilidad y reutilización:
 - Al utilizar CSS, es posible definir estilos una vez y aplicarlos a múltiples elementos en el documento XML. Esto promueve la reutilización de estilos y simplifica la tarea de mantenimiento, ya que los cambios en el diseño se pueden realizar de forma centralizada en la hoja de estilos CSS.
- Mejora de la accesibilidad:
 - CSS ofrece características para mejorar la accesibilidad del documento XML, como el ajuste del tamaño del texto, el contraste de color y la navegación mediante el uso de estilos visuales claros y consistentes.



Display Propiedad más importante para establecer el modo de visualización

- Los modos de visualización más utilizados son:
 - **none** (ninguno) permite ocultar elementos.
 - **block (bloque)** es el modo de visualización de los elementos de tipo bloque, como párrafos (<p>), encabezados (<h1>, <h2>, ...), etc. El elemento ocupa toda la ventana en horizontal y ocupa el espacio vertical necesario para alojar el contenido del elemento.
 - **inline (elemento en línea)** es el modo de visualización de etiquetas como , , , <a>, etc. El elemento sólo ocupa el espacio necesario para alojar el contenido del elemento. En HTML los elementos en línea tienen que estar contenidos dentro de elementos de bloque.
 - **list-item (lista)** es el modo de visualización de elementos de listas como , <dd>, <dt>.
 - **table (tabla), table-row (fila) y table-cell (celda)** se combinan para mostrar elementos en forma de tabla.

- Ejemplo biblioteca.xml
 - estilosBiblioteca.css
 - estilosBiblioteca1.css
 - Cuando el único elemento de bloque es libro
 - estilosBiblioteca2.css
 - Título, autor y fecha de publicación como bloque
 - estilosBiblioteca3.css
 - Mostrando el documento como una única lista no ordenada
 - estilosBiblioteca4.css
 - Mostrando el documento como una tabla
 - estilosBiblitoecca5.css
 - Mostrando los elementos como listas ordenadas (ERROR)
 - Problema: uso de contadores en elementos anidados

Solución a los problemas de contadores en elementos anidados

- Uso de pseudoclases y pseudoelementos
 - Los pseudo-elementos **::before** y **::after** permiten añadir contenido a un elemento desde la hoja de estilo, al principio o al final del elemento.
 - El contenido se genera mediante la propiedad content y puede incluir texto:
- Uso de contadores
 - counter, propiedad que permite añadir un contador a un elemento
 - counter-increment aumenta el contador
 - counter-reset pone a cero el contador
- estilosBiblitoea6.css