



XML

CSS

JSON

Agenda

- **Introducción a XML**
- **Sintaxis de XML**
- **Ventajas/desventajas de XML**
- **Casos de Uso de XML**
- **Paralelo con HTML (HyperText Markup Language)**
- **CSS**

XML (Extensible Markup Language)

Introducción a XML:

XML es un lenguaje de marcado que se utiliza para almacenar y transportar datos de manera estructurada.

Se utiliza comúnmente para intercambiar datos entre diferentes sistemas y aplicaciones.

<https://www.tutorialspoint.com/xml/index.htm>

Sintaxis de XML:

Un documento XML consta de elementos, atributos y texto.

Los elementos se definen mediante etiquetas que encierran datos.

Los atributos proporcionan información adicional sobre los elementos.

La sintaxis básica de XML es similar a HTML, pero XML permite definir etiquetas y estructuras personalizadas.



XML

Ejemplo XML:

```
1 <bookstore>
2   <book category="fiction">
3     <title lang="en">Harry Potter</title>
4     <author>J.K. Rowling</author>
5     <year>2005</year>
6     <price>29.99</price>
7   </book>
8   <book category="non-fiction">
9     <title lang="en">The Power of Now</title>
0     <author>Eckhart Tolle</author>
1     <year>1997</year>
2     <price>15.00</price>
3   </book>
4 </bookstore>
5
```

Ventajas de XML:

Flexibilidad: Permite definir estructuras de datos personalizadas.

Interoperabilidad: Puede ser utilizado en diferentes plataformas y sistemas.

Legibilidad: La sintaxis de XML es fácil de entender y leer para humanos.

Desventajas de XML:

Verboso: XML puede ser más verboso en comparación con otros formatos de intercambio de datos como JSON.

Overhead: Puede haber un mayor consumo de recursos debido a etiquetas adicionales y estructuras de datos.

Casos de Uso de XML:

- **Intercambio de Datos:** Utilizado para intercambiar datos estructurados entre sistemas.
- **Configuración de Aplicaciones:** Se utiliza para definir configuraciones y opciones de aplicación.
- **Almacenamiento de Datos:** Puede ser utilizado como formato de almacenamiento de datos en bases de datos y archivos.

Paralelo con HTML (HyperText Markup Language)

1. Similaridades con HTML:

Tanto XML como HTML son lenguajes de marcado.

Ambos utilizan etiquetas para definir la estructura y el contenido de los documentos.

2. Diferencias con HTML:

Propósito:

HTML se utiliza para crear contenido web y define la estructura de una página

Mientras que XML se utiliza para almacenar y transportar datos.

Flexibilidad:

XML es más flexible y permite definir estructuras personalizadas

Mientras que HTML tiene una estructura predefinida para crear páginas web.

Semántica:

HTML se centra en la presentación y el diseño de páginas web

Mientras que XML se centra en la estructura y organización de datos.

Sintaxis XML

1. Elementos y etiquetas: Los datos en XML se organizan en elementos, que están delimitados por etiquetas. Por ejemplo:

```
<persona>
  <nombre>Juan</nombre>
  <edad>30</edad>
</persona>
```

2. Atributos: Los elementos pueden tener atributos que proporcionan información adicional. Por ejemplo:

```
<persona id="1">
  <nombre>Juan</nombre>
  <edad>30</edad>
</persona>
```

3. Texto: Los elementos pueden contener texto además de otros elementos. Por ejemplo:

```
<persona>
  <nombre>Juan</nombre>
  <edad>30</edad>
  <ciudad>Buenos Aires</ciudad>
</persona>
```

Sintaxis XML: ejemplos

1. Lista de tareas:

```
<tareass>  
  <tarea id="1" estado="pendiente">Comprar leche</tarea>  
  <tarea id="2" estado="completada">Pagar la factura</tarea>  
  <tarea id="3" estado="pendiente">Hacer ejercicio</tarea>  
</tareass>
```

2. Libro:

```
<libro>  
  <titulo>El señor de los anillos</titulo>  
  <autor>J.R.R. Tolkien</autor>  
  <genero>Fantasía</genero>  
  <precio>25.99</precio>  
</libro>
```

Ejercicios Prácticos

Crea un archivo XML para almacenar información sobre tus películas favoritas.

Diseña un documento XML para representar información sobre estudiantes de una clase, incluyendo su nombre, edad y calificaciones.

Crea un archivo XML que describa la estructura de un menú de un restaurante, incluyendo platos principales, bebidas y postres.

CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para controlar el aspecto y el diseño de los documentos HTML. Permite definir el color, la fuente, el tamaño y la disposición de los elementos en una página web.



CSS (Cascading Style Sheets)

¿Cómo Funciona CSS?

CSS funciona mediante la asociación de reglas de estilo con elementos HTML. Cada regla de estilo consiste en un selector y un conjunto de propiedades y valores. El selector especifica a qué elementos HTML se aplicarán los estilos, mientras que las propiedades y valores definen cómo se verán esos elementos.

```
h1 {  
  color: blue;  
  font-size: 24px;  
}
```

CSS (Cascading Style Sheets)

Incorporación de CSS en HTML

Hay tres formas principales de incorporar CSS en un documento HTML:

1. Estilos en Línea: Se aplican directamente a un elemento utilizando el atributo **style**.

```
<h1 style="color: blue; font-size: 24px;">Título</h1>
```

2. Etiqueta <style>: Se coloca dentro del elemento <head> y contiene reglas de estilo CSS.

```
<head>
  <style>
    h1 {
      color: blue;   font-size: 24px;
    }
  </style>
</head>
```

3. Archivo CSS Externo: Se enlaza a través de la etiqueta <link> y contiene todas las reglas de estilo.

```
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
```

Introducción a JSON

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero y fácil de leer.

Se utiliza comúnmente para transmitir datos entre un servidor y un cliente web, pero también puede ser utilizado para almacenar datos estructurados localmente.

{.json}

Introducción a JSON

¿Por qué usar JSON?

JSON es ideal para intercambiar datos entre diferentes plataformas y lenguajes de programación, debido a su naturaleza universal y legible por humanos.

<https://www.hostinger.com/tutorials/what-is-json>

<https://www.tutorialspoint.com/json/index.htm>

{.json}

Estructura de JSON

Objetos: Colección de pares clave-valor encerrados entre llaves {}. Las claves son cadenas de texto y los valores pueden ser cualquier tipo de dato válido en JSON (números, cadenas, booleanos, objetos o arreglos).

```
{  
  "nombre": "Juan",  
  "edad": 30,  
  "casado": false,  
  "dirección": {  
    "calle": "123 Calle Principal",  
    "ciudad": "Ciudad",  
    "país": "País"  
  },  
  "intereses": ["fútbol", "viajes", "lectura"]  
}
```

Estructura de JSON

Arreglos: Colección ordenada de valores encerrados entre corchetes []. Los valores pueden ser cualquier tipo de dato válido en JSON.

```
["manzana", "banana", "naranja"]
```