XML y derivados

# 

[Vista de pájaro 1](#_7cx9g6fu94bc)

[Sobre XML 1](#_wj8gax19a5gk)

[¿Qué es XML? 1](#_gmia0sx0kbuq)

[¿Qué diferencia hay con HTML? 2](#_mtzsxgurrzkt)

[XML separa los datos de la presentación 2](#_2iv2htgwmja5)

[XML separa datos de HTML 3](#_t1p1q0yjyynz)

[La estructura del árbol XML 3](#_qntbmdws0y6y)

[Sintaxis XML 5](#_qo67k9oq6zfj)

[Comentarios en XML 6](#_r59aunb0vd9p)

[¿Qué es un elemento XML? 6](#_d9k4m9h2mqqz)

[Mejores prácticas de nomenclatura 7](#_lszk0igpycw8)

[Uso de las comillas en XML 7](#_hqsv4nrjbl3f)

[Espacios de nombres XML 8](#_fv2q6xx9fwyb)

[Mostrando XML 9](#_lpu4mu44xk9q)

[Visualización de XML con XSLT 10](#_7i260abr47j1)

[Definir hoja XSLT 10](#_exvpitx1rwzv)

[Vincular un XML a un XSLT 10](#_iljf1s4ybjxy)

[Elemento XSLT <xsl:plantilla> 12](#_u40dzvnip6z1)

# 

# Vista de pájaro

* XML (eXtensible Markup Language) → Contiene datos
* XSLT (eXtensible Stylesheet Language Transformations)→ aplica estilo a un documento XML
* XPath → sintaxis para definir partes de un documento XML (se unsa en XSLT)

# Sobre XML

## ¿Qué es XML?

* XML (eXtensible Markup Language) significa lenguaje de marcado extensible.
* XML es un lenguaje de marcado muy parecido a HTML
* XML fue diseñado para almacenar y transportar datos.
* XML fue diseñado para ser legible tanto por humanos como por máquinas.
* XML es una recomendación del W3C
* XML simplifica el intercambio de datos
* XML simplifica el transporte de datos
* XML simplifica los cambios de plataforma
* XML simplifica la disponibilidad de datos

## ¿Qué diferencia hay con HTML?

* XML fue diseñado para transportar datos, centrándose en qué datos son.
* HTML fue diseñado para mostrar datos, centrándose en cómo se ven los datos.
* Las etiquetas XML no están predefinidas como las etiquetas HTML.
* El lenguaje XML no tiene etiquetas predefinidas.
* Las etiquetas del ejemplo anterior (como <to> y <from>) no están definidas en ningún estándar XML. Estas etiquetas son "inventadas" por el autor del documento XML.
* HTML funciona con etiquetas predefinidas como <p>, <h1>, <table>, etc.
* Con XML, el autor debe definir tanto las etiquetas como la estructura del documento.

**Ejemplo:**

<note>

<to>Tove</to>

<from>Jani</from>

<heading>Reminder</heading>

<body>Don't forget me this weekend!</body>

</note>

El XML anterior no HACE nada. XML es solo información envuelta en etiquetas.

## XML separa los datos de la presentación

* XML no contiene ninguna información sobre cómo mostrarse.
* Los mismos datos XML se pueden utilizar en muchos escenarios de presentación diferentes.
* Debido a esto, con XML existe una separación total entre datos y presentación.

## XML separa datos de HTML

* Al mostrar datos en HTML, no debería tener que editar el archivo HTML cuando los datos cambien.
* Con XML, los datos se pueden almacenar en archivos XML separados.
* Con unas pocas líneas de código JavaScript, puede leer un archivo XML y actualizar el contenido de datos de cualquier página HTML.

Ejemplo: XMLNews es un estandar para intercambio de noticias:

<?xml version="1.0" encoding="UTF-8"?>

<nitf>

<head>

<title>Colombia Earthquake</title>

</head>

<body>

<headline>

<hl1>143 Dead in Colombia Earthquake</hl1>

</headline>

<byline>

<bytag>By Jared Kotler, Associated Press Writer</bytag>

</byline>

<dateline>

<location>Bogota, Colombia</location>

<date>Monday January 25 1999 7:28 ET</date>

</dateline>

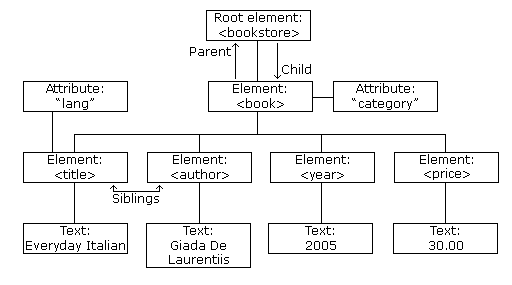
</body>

</nitf>

## La estructura del árbol XML

El documento XML sigue una estructura de árbol comenzando siempre en una “ráiz”.

Ejemplo de librería:



<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

<book category="cooking">

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

<book category="children">

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="web">

<title lang="en">Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

</bookstore>

Un árbol XML comienza en un elemento raíz y se ramifica desde la raíz hasta los elementos secundarios .

Todos los elementos pueden tener subelementos (elementos secundarios):

Los términos padre, hijo y hermano se utilizan para describir las relaciones entre elementos.

Los padres tienen hijos. Los niños tienen padres. Los hermanos son niños del mismo nivel (hermanos y hermanas).

Todos los elementos pueden tener contenido de texto (Harry Potter) y atributos (categoría="cocina").

## Sintaxis XML

Un prólogo define la versión XML y la codificación de caracteres:

<?xml version="1.0" encoding="UTF-8"?>

El prólogo XML no tiene una etiqueta de cierre! Esto no es un error. El prólogo no forma parte del documento XML.

* El prólogo XML es opcional. Si existe, debe aparecer primero en el documento.
* Los documentos XML pueden contener caracteres internacionales, como el noruego øæå o el francés êèé.
* Para evitar errores, debe especificar la codificación utilizada o guardar sus archivos XML como UTF-8.
* UTF-8 es la codificación de caracteres predeterminada para documentos XML.

La siguiente línea es el elemento raíz del documento: <bookstore>

En XML, es ilegal omitir la etiqueta de cierre. Todos los elementos deben tener una etiqueta de cierre

Los elementos XML pueden tener atributos en pares nombre/valor como en HTML.

En XML, los valores de los atributos siempre deben ir entre comillas:

<note date="12/11/2007">

<to>Tove</to>

<from>Jani</from>

</note>

Algunos caracteres tienen un significado especial en XML, como < y > que se interpreta como inicio de etiqueta. Para evitar este error, reemplace el carácter "<" con una referencia de entidad :

<message>salary &lt; 1000</message>

Hay cinco referencias de entidades predefinidas en XML:

&lt; < less than

&gt; > greater than

&amp; & ampersand

&apos; ' apostrophe

&quot; " quotation mark

Los elementos XML deben seguir estas reglas de nomenclatura:

* Los nombres de los elementos distinguen entre mayúsculas y minúsculas.
* Los nombres de los elementos deben comenzar con una letra o un guión bajo.
* Los nombres de los elementos no pueden comenzar con las letras xml (o XML, o Xml, etc.)
* Los nombres de los elementos pueden contener letras, dígitos, guiones, guiones bajos y puntos.
* Los nombres de los elementos no pueden contener espacios.

Se puede utilizar cualquier nombre, no se reservan palabras (excepto xml).

## Comentarios en XML

La sintaxis para escribir comentarios en XML es similar a la de HTML:

<!-- This is a comment -->

No se permiten dos guiones en medio de un comentario:

<!-- This is an invalid -- comment →

## ¿Qué es un elemento XML?

Un elemento XML es todo, desde (incluida) la etiqueta inicial del elemento hasta (incluida) la etiqueta final del elemento.

Un elemento puede contener:

* texto
* atributos
* otros elementos
* o una mezcla de los anteriores

Ejemplo:

<bookstore>

<book category="children">

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="web">

<title>Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

</bookstore>

En el ejemplo anterior:

<título>, <autor>, <año> y <precio> tienen contenido de texto porque contienen texto (como 29,99).

<librería> y <libro> tienen contenidos de elementos , porque contienen elementos.

<libro> tiene un atributo (categoría="niños").

## Mejores prácticas de nomenclatura

Cree nombres descriptivos, como este: <persona>, <nombre>, <apellido>.

Crea nombres cortos y simples, como este: <título\_del\_libro> no así: <el\_título\_del\_libro>.

Evitar "-". Si nombra algo como "nombre", algún software puede pensar que desea restar "nombre" de "nombre".

Evitar ".". Si nombra algo "primero.nombre", algún software puede pensar que "nombre" es una propiedad del objeto "primero".

Evitar ":". Los dos puntos están reservados para espacios de nombres (más adelante).

Las letras que no están en inglés como éòá ​​son perfectamente legales en XML, ¡pero tenga cuidado con los problemas si su software no las admite!

## Uso de las comillas en XML

Los siguientes ejemplos son válido:

<person gender="female">

<person gender='female'>

Si el valor del atributo contiene comillas dobles, puede utilizar comillas simples, como en este ejemplo:

<gangster name='George "Shotgun" Ziegler'>

<gangster name="George &quot;Shotgun&quot; Ziegler">

## Espacios de nombres XML

Los espacios de nombres XML proporcionan un método para evitar conflictos de nombres de elementos.

En XML, los nombres de los elementos los define el desarrollador. Esto suele generar un conflicto al intentar mezclar documentos XML de diferentes aplicaciones XML.

Este XML contiene información de la tabla HTML:

<table>

<tr>

<td>Apples</td>

<td>Bananas</td>

</tr>

</table>

Este XML contiene información sobre una mesa (un mueble):

<table>

<name>African Coffee Table</name>

<width>80</width>

<length>120</length>

</table>

Los conflictos de nombres en XML se pueden evitar fácilmente utilizando un prefijo de nombre.

Este XML contiene información sobre una tabla HTML y un ferniture:

Cuando se utilizan prefijos en XML, se debe definir un espacio de nombres para el prefijo.

El espacio de nombres se puede definir mediante un atributo xmlns en la etiqueta de inicio de un elemento.

La declaración del espacio de nombres tiene la siguiente sintaxis. xmlns: prefijo =" URI ".

<root>

<h:table xmlns:h="http://www.w3.org/TR/html4/">

<h:tr>

<h:td>Apples</h:td>

<h:td>Bananas</h:td>

</h:tr>

</h:table>

<f:table xmlns:f="https://www.w3schools.com/furniture">

<f:name>African Coffee Table</f:name>

<f:width>80</f:width>

<f:length>120</f:length>

</f:table>

</root>

En el ejemplo anterior:

El atributo xmlns en el primer elemento <table> le da al prefijo h: un espacio de nombres calificado.

El atributo xmlns en el segundo elemento <table> le da al prefijo f: un espacio de nombres calificado.

Cuando se define un espacio de nombres para un elemento, todos los elementos secundarios con el mismo prefijo se asocian con el mismo espacio de nombres.

Los espacios de nombres también se pueden declarar en el elemento raíz XML:

<root xmlns:h="http://www.w3.org/TR/html4/"

xmlns:f="<https://www.w3schools.com/furniture>">

# Mostrando XML

No espere que los archivos XML se muestren como páginas HTML.

La mayoría de los navegadores mostrarán un documento XML con elementos codificados por colores.

A menudo, se puede hacer clic en un signo más (+) o menos (-) a la izquierda de los elementos para expandir o contraer la estructura del elemento.

Los documentos XML no contienen información sobre cómo mostrar los datos.

Dado que las etiquetas XML las "inventa" el autor del documento XML, los navegadores no saben si una etiqueta como <table> describe una tabla HTML o una mesa de comedor.

Sin ninguna información sobre cómo mostrar los datos, los navegadores pueden simplemente mostrar el documento XML tal como está.

## Visualización de XML con XSLT

SLT (eXtensible Stylesheet Language Transformations) es el lenguaje de hojas de estilo recomendado para XML.

XSLT es mucho más sofisticado que CSS. Con XSLT puede agregar/eliminar elementos y atributos hacia o desde el archivo de salida. También puedes reorganizar y ordenar elementos, realizar pruebas y tomar decisiones sobre qué elementos ocultar y mostrar, y mucho más.

XSLT utiliza XPath para buscar información en un documento XML.

### Definir hoja XSLT

<xsl:stylesheet version="1.0"

xmlns:xsl="<http://www.w3.org/1999/XSL/Transform>">

o

<xsl:transform version="1.0"

xmlns:xsl="<http://www.w3.org/1999/XSL/Transform>">

Para obtener acceso a los elementos, atributos y características XSLT debemos declarar el espacio de nombres XSLT en la parte superior del documento.

xmlns:xsl="http://www.w3.org/1999/XSL/Transform" apunta al espacio de nombres oficial W3C XSLT. Si utiliza este espacio de nombres, también debe incluir el atributo versión="1.0".

### Vincular un XML a un XSLT

Tenemos el XML:

<?xml version="1.0" encoding="UTF-8"?>

<catalog>

<cd>

<title>Empire Burlesque</title>

<artist>Bob Dylan</artist>

<country>USA</country>

<company>Columbia</company>

<price>10.90</price>

<year>1985</year>

</cd>

.

.

</catalog>

Creamos la hoja de estilo XSL

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h2>My CD Collection</h2>

<table border="1">

<tr bgcolor="#9acd32">

<th>Title</th>

<th>Artist</th>

</tr>

<xsl:for-each select="catalog/cd">

<tr>

<td><xsl:value-of select="title"/></td>

<td><xsl:value-of select="artist"/></td>

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

Ahora agreguemos la referencia XSL a XML:

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>

<catalog>

<cd>

<title>Empire Burlesque</title>

<artist>Bob Dylan</artist>

<country>USA</country>

<company>Columbia</company>

<price>10.90</price>

<year>1985</year>

</cd>

.

.

</catalog>

## Elemento XSLT <xsl:plantilla>

Una hoja de estilos XSL consta de uno o más conjuntos de reglas que se denominan plantillas.

Una plantilla contiene reglas que se aplicarán cuando coincida un nodo específico.

El elemento <xsl:template> se utiliza para crear plantillas.

El atributo de coincidencia se utiliza para asociar una plantilla con un elemento XML. El atributo de coincidencia también se puede utilizar para definir una plantilla para todo el documento XML. El valor del atributo de coincidencia es una expresión XPath (es decir, match="/" define todo el documento).

Explicando el código anterior

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

….

Dado que una hoja de estilos XSL es un documento XML, siempre comienza con la declaración XML: <?xml version="1.0" encoding="UTF-8"?> .

El siguiente elemento, <xsl:stylesheet> , define que este documento es un documento de hoja de estilos XSLT (junto con el número de versión y los atributos del espacio de nombres XSLT).

El elemento <xsl:template> define una plantilla. El atributo match="/" asocia la plantilla con la raíz del documento fuente XML.

El contenido dentro del elemento <xsl:template> define algo de HTML para escribir en la salida.

Las dos últimas líneas definen el final de la plantilla y el final de la hoja de estilo.

El resultado de este ejemplo fue un poco decepcionante porque no se copiaron datos del documento XML al resultado. En el próximo capítulo aprenderá cómo utilizar el elemento <xsl:value-of> para seleccionar valores de los elementos XML.

## El elemento <xsl:value-of>

El elemento <xsl:value-of> se puede utilizar para extraer el valor de un elemento XML y agregarlo al flujo de salida de la transformación:

En el ejemplo:

<xsl:for-each select="catalog/cd">

<tr>

<td><xsl:value-of select="title"/></td>

<td><xsl:value-of select="artist"/></td>

</tr>

</xsl:for-each>