

Introducción a XML

Es un lenguaje libre y abierto, pero no es un lenguaje de programación, es un lenguaje de marcas (una forma de añadir información sobre los datos a los propios datos), un “metalenguaje”, es decir, un lenguaje usado para describir otro lenguaje.

SGML es un estándar enorme y complejo, por lo que consumen demasiado. Tanto XML como HTML derivan de él.

XML actúa como estándar para crear lenguajes específicos con sus propias etiquetas, estructuras, etc. Impone que deben estar basadas en caracteres y ser legibles. Además puede ser usado en cualquier SO y por cualquier herramienta. Es un formato público y un estándar abierto.

El diseñador se puede centrar en el uso de etiquetas, contenido y su estructura.

A los ficheros XML se les comprueba la sintaxis y se puede comprobar la estructura (validación).

XML estructura y significado de los datos

HTML forma en la que se presentan los datos.

Sintaxis XML

XML: eXtensible Markup Language, creado para organizar, almacenar y transportar información. Sus etiquetas (tag) no están predefinidas (en HTML sí), por lo tanto permite definir nuestras propias etiquetas y la estructura del documento (Este documento no hace nada)

- Estructura en árbol:

Deben contener un elemento raíz (root), es padre de todos los demás y el árbol empieza por este nodo raíz.

Los elementos del árbol pueden tener hijos, y estos a su vez contenido textual, atributos u otros elementos (contenedores).

- Reglas XML

Todos los elementos deben tener etiqueta de cierre o final.

Las etiquetas distinguen mayúsculas y minúsculas. (Case sensitive)

Los elementos deben estar correctamente anidados.

Los XML deben tener un elemento raíz

Los atributos deben estar entrecomillados

Sustituir “<”(<); “>”(>); “ & ” (&); “ &apos ” (‘); “ " ” (“), nueva línea “LF”. XML no trunca los espacios. Los nombres de los elementos no pueden comenzar por número o carácter de puntuación, ni contener espacios.

- Atributos:

Proporcionan información adicional sobre el elemento, sus valores deben de estar entrecomillados, (valido tanto simple como doble, si el valor contiene doble comilla usa las simples)

Atributos: es recomendable no usarlos en XML y usar elementos.

Inconvenientes:

- No pueden contener valores múltiples
- No pueden contener estructuras en árbol
- No se pueden “expandir o extender con facilidad en el futuro.

En general son difíciles de leer y mantener.

Usa:

- Elementos para los datos
- Atributos para la información que no es relevante
- Los metadata (datos sobre los datos) atributos, el resto elementos

Siempre asociado a un elemento.

Se define dentro de la etiqueta del elemento `<pepe categoria="noob">`

Un elemento puede tener uno o más atributos

- Validación Vs Bien Formado

Bien Formado: Sintaxis correcta, si no detienen el procesamiento del documento.

Válido: contrastado o validado contra una DTD o schema

Si hay algún error se suele mostrar en el navegador aunque algunos lo ocultan en la consola

- Codificación y estructura lógica

La codificación sigue siendo UTF-8, con declaración inicial encoding, opcional.

`<?xml version="1.0" encoding="UTF-8" ?>` La declaración XML es obligatoria.

DTDs

DTD: Document Type Definition, define la estructura de un documento XML, elementos, atributos válidos, etc. Puede ser declarado dentro del XML (inline) o como una referencia externa (o los dos)

Los ficheros XML tienen una descripción de su formato

DTD estandarizada para intercambiar datos.

Las aplicaciones poder verificar los datos recibidos o los propios datos.

- DTDs Externas

DTDs externas: privadas, almacenadas en local o en una página web; de acceso público, en la web

DTDs de acceso público: estándar para uso público.

- DTD Interna:

Solo fichero XML, se declara al inicio del fichero y entre corchetes

- PCDATA Y CDATA

PCDATA: “parse carácter data”, texto que se encuentra entre las etiquetas, será procesado y analizado por el parser. No debería contener &, <, >, etc.

CDATA: “Character data” texto que no será analizado por el parser.

Un elemento puede contener, datos finales(PCDATA), otros elementos, elemento vacío(empty), elemento Any.

- Numero de ocurrencias:

Una (message)

Secuencia obligatoria (,) (nombre,apellido)

Al menos una (+) (aviso+)

Cero o más (*) (alumno*)

Cero o una (?) (prioridad?)

Una u otra, pipe (|) (message|body)

- Valor por Defecto

#REQUIRED: El atributo es obligatorio

DTD: <!ATTLIST alumno numero CDATA **#REQUIRED**>

Valid XML: < alumno numero="12" />

Invalid XML: < alumno />

#IMPLIED: ni es obligatorio ni tenemos valor por defecto.

DTD: <!ATTLIST contacto fax CDATA **#IMPLIED**>

ValidXML: <contacto fax="956345678" />

ValidXML: <contacto />

#FIXED valor: no se permite cambiarlo. Si se incluye otro valor, error.

DTD: <!ATTLIST envioempresa CDATA **#FIXED** "MS">

ValidXML: < envioempresa ="MS" />

InvalidXML: < envioempresa ="Unifert" />

Enumerado: un valor entre un conjunto

DTD: <!ATTLIST pago tipo (**contado | CC**) "**contado**">

XML example: <pago tipo="contado" />

<pago tipo="CC" />

Detrás del enumerado se puede indicar **#REQUIRED**, o **#IMPLIED**

Si no se especifica en el elemento se le asigna ese valor

<!ATTLIST cuadrado lado CDATA "0">

Por defecto será 0 el valor

- Tipos de atributos (I)

CDATA: puede contener cualquier carácter

NMTOKEN: solo puede contener letras, dígitos, punto (.), guion (-), subrayado (_) y dos puntos (:)

NMTOKENS: NMTOKEN + espacios (uno o más espacios, retornos de carro o tabulaciones)

- Tipos de atributos (II)

ID: Solo caracteres validos en NMTOKEN y comenzar por una letra, solo puede haber uno por elemento, y debe de ser único.

- Tipos de atributos (III)

IDREF: tiene que corresponder con el valor de algún atributo ID

IDREFS puede contener varias referencias a los elementos con atributos ID separados por espacios en blanco.

Introducción a XSLT

Transformar XML en otro XML, HTML O XHTML antes de ser presentado por el navegador (cliente (navegador) o en servidor. Con XSLT se pueden añadir o eliminar elementos y atributos.

Una hoja de XSL consiste de uno o más conjuntos de reglas (templates o plantilla).

El elemento **<xsl:stylesheet>**define que es una hoja de estilo XSLT, y añade como atributos la versión y el espacio de nombres.

El elemento **<xsl:template>**define una plantilla:

- `match=“/”` asocia la plantilla con la raíz del documento XML
- El contenido define código HTML

- Diferencia entre XSLT y CSS

CSS se centra en cómo se presentan los datos (parte visual), usa su propia sintaxis.

XSLT cambia la estructura y el tipo de los datos y usa la sintaxis XML.