LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE INFORMACIÓN

Bloque XML: UD2, Sintaxis XML

Índice

Introducción

- Vamos a seguir la introducción en:
  - http://w3schools.com/xml/xml_whatis.asp
- Ideas básicas:
  - XML : E**X**tensible **M**arkup **L**anguage
  - Creado para organizar, almacenar y transportar información.
  - Las etiquetas XML (tags) no están predefinidas.
    - En HTML sí: solo puede usar las etiquetas definidas en el estándar (o definidas por el navegador). Si utiliza una que no conoce, la ignora.
  - XML permite definir nuestras propias etiquetas y la estructura del documento
  - Un documento XML NO hace nada.
  - XML está diseñado para ser autodescriptivo (si se usa bien, claro..)
- Es una recomendación del W3C

Fichero xml


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

<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Estructura en árbol

- Los docs XML deben contener un elemento raíz (root)
  - Este elemento es el padre de todos los demás.
- El árbol comienza en este nodo raíz.
- Los elementos del árbol pueden tener hijos

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

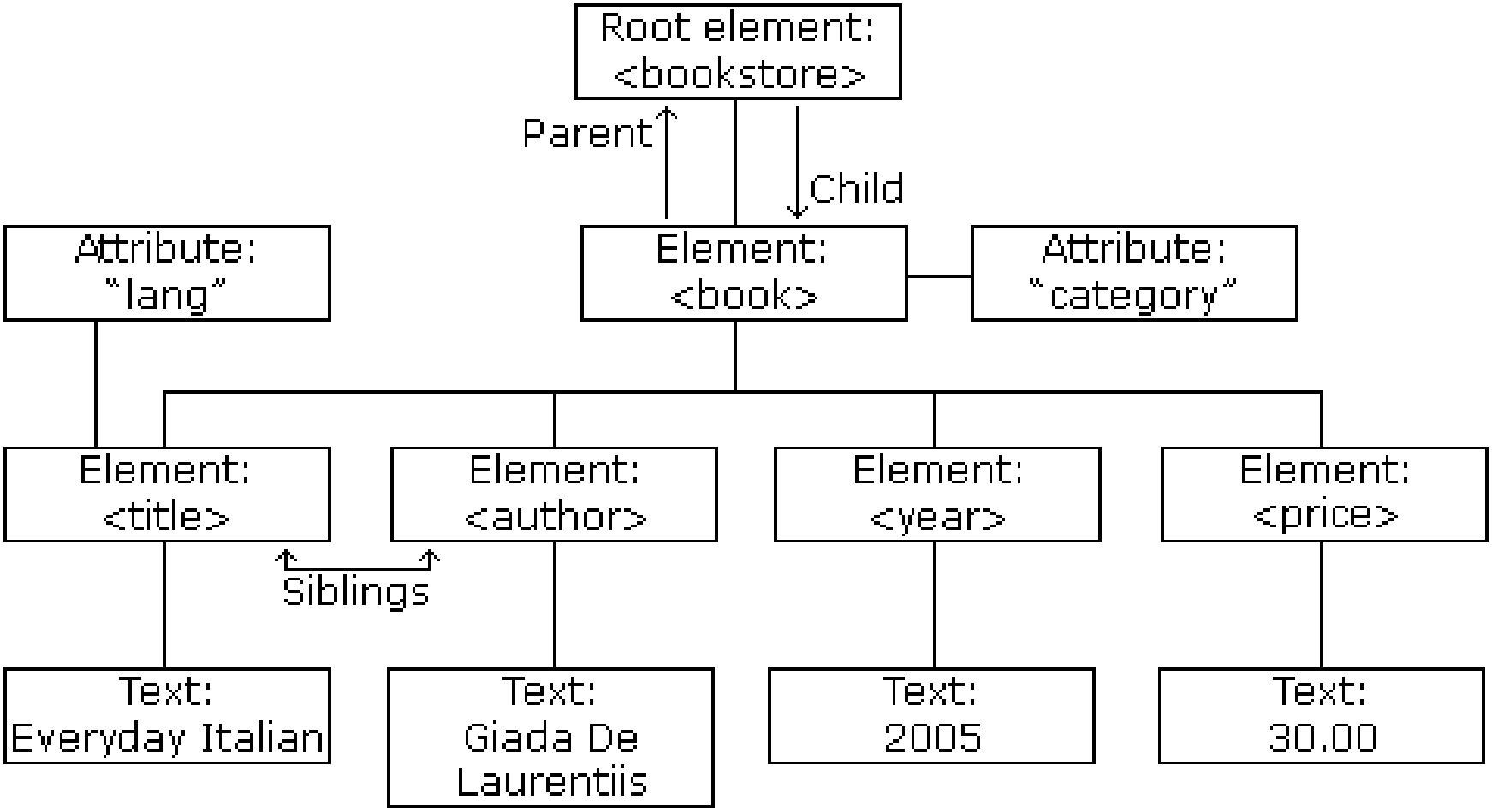
- Los elementos pueden tener
  - contenido textual,
  - atributos
  - otros elementos (se convierten en contenedores).

Ejemplo bookstore

Atributo, dentro etiqueta elemento:
Nombre_atributo="valor_atributo"

```
<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>
</bookstore>
```

Árbol que le corresponde



Reglas XML

- **Todos los elementos deben tener etiqueta de cierre o final**
  - This is another paragraph
- **Las etiquetas distinguen mayúsculas/minúsculas:** (*Case Sensitive*)
  - <Message>This is incorrect</message>
    <message>This is correct</message>
- **Los elementos deben estar correctamente anidados**
  - This text is bold and italic *HTML, correcto, XML incorrecto*
  - This text is bold and italic *XML, correcto*
- **Los docs XML deben tener un elemento raiz**
  - Sólo uno, padre de todos los demás
- **Los atributos deben estar entrecomillados**

  <note date=12/11/2007> *XML incorrecto*

  …

  <note date="12/11/2007"> *XML correcto*

Otras consideraciones

- Referencias de entidades: Concepto + general, alias. 5 predefinidas

< > & ' "	< less than > greater than & ampersand ' apostrophe " quotation mark	Nota: Sólo son ilegales < y &. Es recomendable sustituir >

- Comentarios: similar a HTML

```
<!--   This is a comment   -->
```

- Espacios en blanco:
  - XML no trunca los espacios (recordad que HTML sí).
- Nueva línea: se almacena como LF.
  - En Windows normalmente una "nueva línea" se almacena como dos caracteres, CR y LF. En Unix y Macintosh LF.

XML Naming Rules

- Los nombre de los elementos
  - Pueden contener letras, números y otros caracteres..
  - No pueden comenzar con número o carácter de puntuación
  - Los nombres no pueden contener espacios
  - Cualquier nombre puede ser usado, no hay palabras reservadas.

Buenas prácticas

- Los nombres deben ser significativos.
- Los nombres con un subrayado son aceptables: primer_apellido, ..
- Debería ser en lo posible cortos y simples. <book_title> y no <the_title_of_the_book>.
- Evita los guiones "-". Se puede confundir con la resta..
- Evita los puntos “.” Puede dar lugar a confusión con objetos y propiedades.
- Evita los ":" , reservados para los espacios de nombrado.
- Muchas veces los docs XML se corresponden con una tabla o base de datos. Es buena idea seguir las reglas de nombrado de la base de datos en el doc XML.
- Los caracteres no ingleses son válidos, pero ten cuidado con los problemas que podría generar en SW de terceros.

Atributos

- Proporcionan información adicional sobre el elemento.
- Los valores de los atributos deben estar entrecomillados.
- Puedes usar comilla simple o comilla doble.
- Si el valor del atributo contiene dobles comillas, usa las simples:
  - <gangster name='George "Shotgun" Ziegler'>
  - o usar las entidades carácter predefinidas:
    - <gangster name="George "Shotgun" Ziegler">

¿Atributos o elementos?

- No hay reglas fijas para decidir cuando usar atributos o elementos
- Aunque en HTML se usan mucho, en XML es recomendable no usarlos, o no abusar de ellos, y usar elementos.
- Inconvenientes de los atributos:
  - No pueden contener valores múltiples
  - No pueden contener estructuras en árbol
  - No se pueden "expandir" o extender con facilidad en el futuro
- En general los atributos son difíciles de leer y mantener.
- Usa:
  - Elementos para los datos
  - Atributos para la información que no es relevante para los datos.
  - Es decir, los metadata (datos sobre los datos) deberían ser atributos, el resto elementos.

Comparamos

- Ejemplo con atributo:

```
<note date="10/01/2008">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

- Ejemplo con elemento:

```
<note>
    <date>10/01/2008</date>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

Incluso que date tenga 3 elementos..

+ de atributos

- Siempre asociado a un elemento
- Se define dentro de la etiqueta de inicio de dicho elemento

  <book **category="CHILDREN“** >

  </book>

- Un elemento puede tener uno o más atributos

  <book **categoria="CHILDREN“ localizacion=“almacen1”** >

  - Se recomienda no abusar !!

Validación frente a bien formado

- Bien formado: sintaxis correcta.
  - Las 5 reglas: root, closing tags, case sensitive, nested & quoted atributes
- Válido: contrastado o validado contra una DTD o schema.
  - Necesitamos un documento dónde se especifica la estructura del documento => lo veremos en tema posterior.
- Errores en sintaxis DETIENEN el procesamiento del documento.
- Ejemplo: http://w3schools.com/xml/xml_validator.asp

Bien formado: pruebas

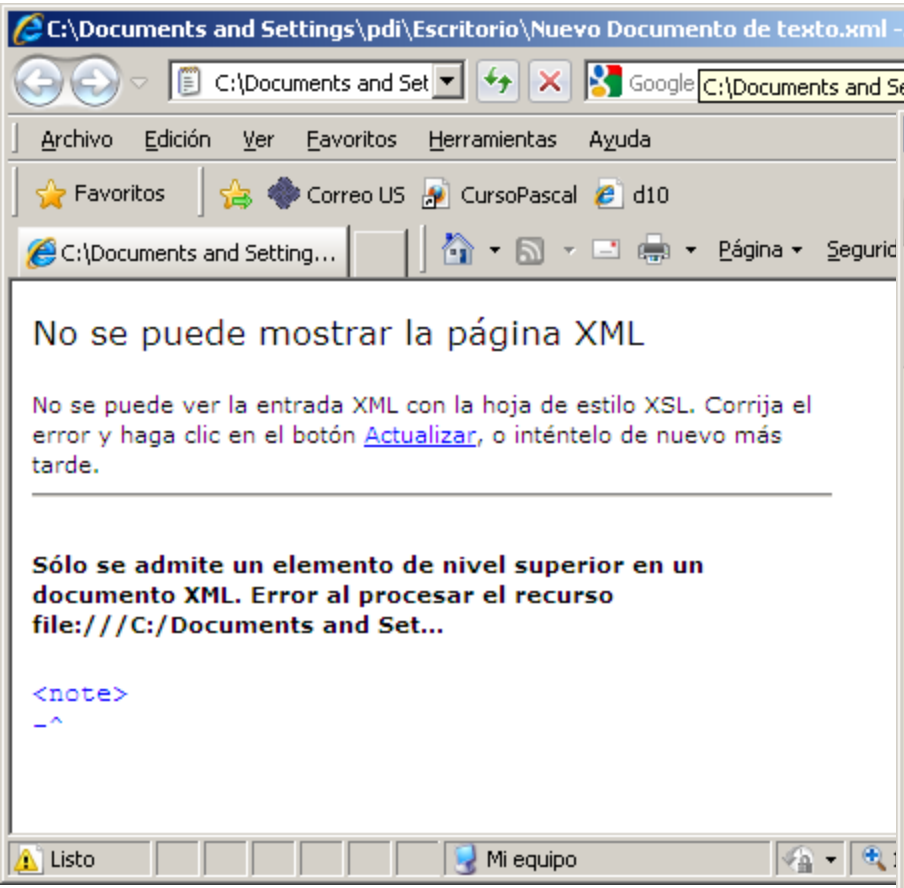
http://w3schools.com/xml/xml_validator.asp

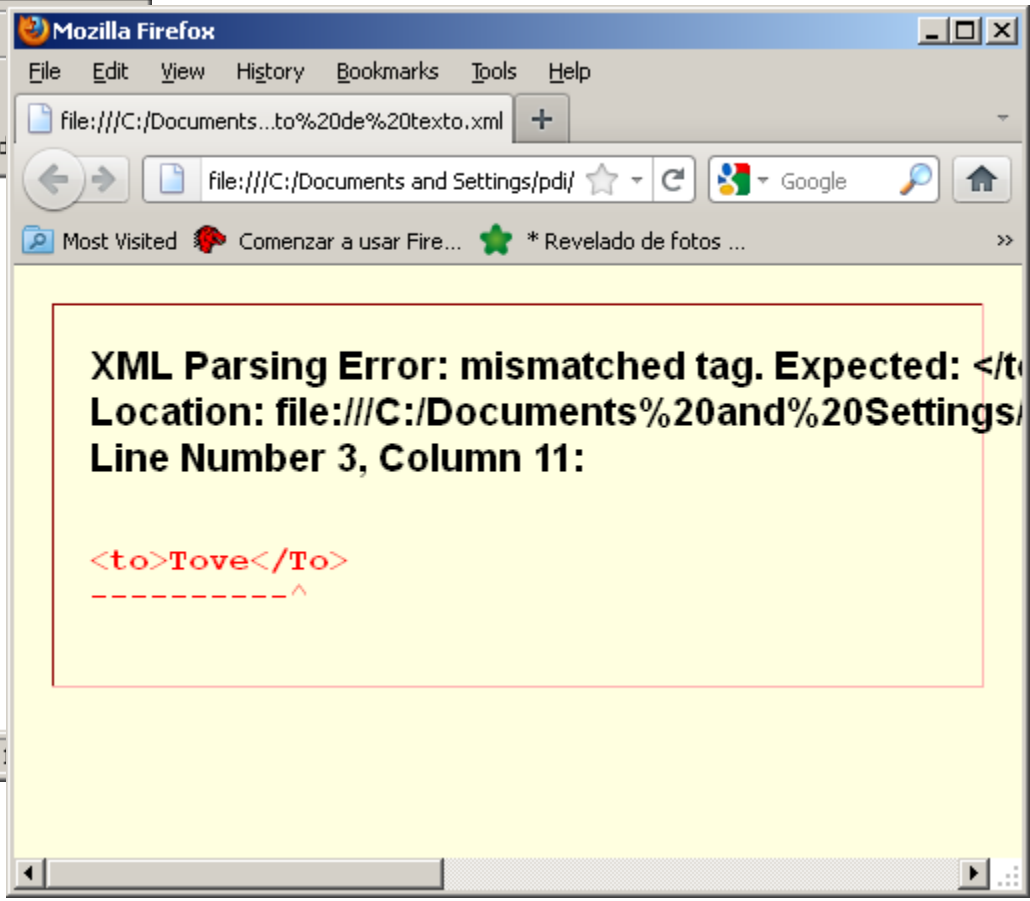
- Añada más de un nodo raíz.
- Olvide cerrar un elemento
- Etiqueta de cierre con mayúsculas/minúsculas
- Anidar mal elementos
- No entrecomillar atributos
- Probar en validador o en local abriendo con navegador => siguiente..

Viendo XML desde navegador

- Con versiones o distintos navegadores => resultados similares..
  - El doc XML se mostrará con los elementos raíz e hijos “coloreados”.
  - Símbolos para contraer/expandir la estructura.
  - Muestran o no la declaración de fichero xml
- Para ver el fuente XML: "View Page Source" or "View Source“.
  - **Note:** en algunos navegadores solo se muestran los elementos de texto
  - Si usamos el inspector de elementos (F12) vemos la transformación por defecto que hace el navegador a HTML y CSS para presentarlo.
- Si hay errores, el navegador muestra el error
  - Ojo, algunos lo ocultan en la consola
- Los docs XML no llevan información sobre presentación
  - los navegadores los muestran “tal cual” (algunos avisan de que no hay estilo asociado)
  - Soluciones para presentación: CSS, XSLT o JavaScript.
  - Ejemplos: http://w3schools.com/xml/xml_view.asp
    - http://w3schools.com/xml/cd_catalog.xml
    - http://w3schools.com/xml/plant_catalog.xml
    - http://w3schools.com/xml/simple.xml
- Importante la extensión del archivo, DEBE SER .xml (probar p.e. con .txt)

Con errores sintácticos





Codificación..

- Concepto ya estudiado en HTML
- La codificación por defecto es UTF-8.
  - Comprobar la codificación de vuestro editor
- Usar en la declaración inicial el atributo encoding

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

```
<?xml version="1.0" ?>
<aviso fecha='hoy' >
    <to>Tove </to>
    <heading> Ñ, ñ, á, é, í, ó, ú </heading>
    <body> El asunto lleva tildes... ¿sale correcto? ¡bravo!</body>
</aviso>
```

Estructura lógica

- Declaración XML, obligatoria

  <?xml version="1.0" **?>**

  Al comienzo del fichero

  - Atributos opcionales
    - Encoding: **encoding="UTF-8"**
    - *DTD externa:* *standalone="yes|no"*
- Doctype: declaración de tipo de documento, la "plantilla"

  <!DOCTYPE note **SYSTEM** "note.dtd">
- Comentarios: mismo formato que HTML

  <!-- filename: prueba.xml **-->**

  Temas posteriores

  - Para legibilidad, o deshabilitar secciones
- Instrucciones de proceso (IP), destinadas aplicación, p.e. presentación

```
<?xml-stylesheet type="text/css"
                 href="cd_catalog.css" ?>
```

Ejercicios

- Crear fichero xml con los datos de un tablón de compra-venta.
  - Cada anuncio tendrá fecha, asunto, texto, precio de venta y contacto.
  - Cada anuncio tendrá un atributo que indique si se ha vendido ya o no, y el precio un atributo que indique la moneda de pago.
- Crear fichero xml con los datos de una clase.
  - La clase tendrá alumnos,
  - estos tendrán nombre, apellidos, fecha de nacimiento y mail.
  - El mail tendrá un atributo de nombre visible, con valores si o no.
  - Comprobar que está bien formado mostrándolo en navegador
  - Añadir notas. Cada elemento nota tendrá distintas asignaturas, y cada una de ellas en elemento para la nota de cada evaluación..

Secciones CDATA

- Estas secciones le indican al parser que ignore el "marcado" de la sección, y pasar esos caracteres como texto a la aplicación.
- Los de limitadores de sección CDATA son
  - Inicio CDATA `<![CDATA[`
  - Final CDATA `]]>`
- Los datos entre estos indicadores se paan directeamente a la aplicación sin pasar por el parser xml
- Ejemplo

```
<desc>
<![CDATA[
  <equipo> BMW & Lotus</equipo>
]]>
</desc>
```

Ejercicio bien formado

- Crear un fichero series.xml. Libertad de decisión.
  - Datos de cada serie: nombre, cadena que la emite, temporadas, nombre de los capítulos etc.
  - Para cada capítulo descripción y enlace de descarga.
  - En el enlace de descarga atributo indicando tipo de descarga: directa, emule, torrent..
- Visualizarlo en los navegadores
- Modificarlo para que no cumpla sintaxis
  - Más de un nodo raíz
  - No cerrar elementos
  - Anidar mal
  - Atributos no entrecomillados..
  - Mayúsculas/minúsculas