

Lenguajes de Marcas y Sistemas de Gestión de Información

Bloque XML: UD3, Validación con DTDs

Recordamos..

2

- XML y su sintaxis.
- El prólogo del doc. XML:
 - ▣ aparecía el DOCTYPE
- Uso de editores
 - ▣ que comprueban sintaxis,
 - ▣ pero además también que el documento es VÁLIDO
- Ejemplo tema anterior:
 - ▣ tablon_con_dtd.xml y tablon.dtd

Document Type Definition (DTD)

3

- Define la estructura de un documento xml: lista de elementos, atributos válidos, etc.
- Puede ser declarado *dentro del documento XML (inline)* o **como una referencia externa** (o los dos)
- ¿Por qué usar una DTD?
 - ▣ Los ficheros XML tienen una descripción de su formato
 - ▣ Grupos distintos pueden acordar usar una DTD estandarizada para intercambiar datos
 - ▣ Las aplicaciones que se desarrollen y usen una DTD estándar pueden verificar que los datos recibidos sean válidos
 - ▣ O usarla para verificar los propios datos..

DTDs externas

4

□ DTDs Privadas

`<!DOCTYPE tablón SYSTEM "tablón.dtd" >`

□ Pueden estar

- Almacenadas en local

- Localizadas en una página web

 - `<!DOCTYPE tablón SYSTEM "http://www.ae.es/tablón.dtd" >`

 - Pública, pero la “empresa” tiene control, la puede modificar.

□ DTDs de acceso público

- Cuando la DTD se considera un estándar, para uso público

- `<!DOCTYPE tablón PUBLIC fpi URL>`

 - Fpi: identificador público formal

XMLCopyEditor valida DTD en URL (NotePad++ NO!)

`<!DOCTYPE tablón SYSTEM "http://localhost/tablón03.dtd">`

Resumimos

5

- Un documento válido
 - ▣ Es bien formado
 - ▣ incluye una referencia a una DTD o schema
 - ▣ y cumple lo especificado en esa DTD o schema.
- El fichero xml tiene
 - ▣ Prólogo
 - El DOCTYPE debe estar incluida en el prólogo
 - Declara el tipo de documento, si es interno o externo, y dónde o como localizarlo para comprobar que el doc xml es válido.
 - ▣ Instancia de datos: contiene el elemento raíz.

DTD Externa

6

- Fichero XML: referencia a DTD externa en DOCTYPE:

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

```
<!DOCTYPE note SYSTEM "note.dtd">
```

```
<note>
```

```
  <to>Tove</to>
```

```
  <from>Jani</from>
```

```
  <heading>Reminder</heading>
```

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

```
</note>
```

- Fichero DTD: fichero "note.dtd" con la DTD sería:

```
<!ELEMENT note (to, from, heading, body)>
```

```
<!ELEMENT to (#PCDATA)>
```

```
<!ELEMENT from (#PCDATA)>
```

```
<!ELEMENT heading (#PCDATA)>
```

```
<!ELEMENT body (#PCDATA)>
```

DTD interna (inline)

7

- Sólo fichero XML:

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

Analizando “por encima”

8

- **<!DOCTYPE note SYSTEM "note.dtd">**
 - ▣ Indica que el elemento raíz es note
 - ▣ Y que la DTD está en el fichero note.dtd
- **<!ELEMENT note (to, from, heading, body) >**
 - ▣ Indica que note tiene 4 elementos hijos que deben aparecer en ese orden.
- **<!ELEMENT to (#PCDATA) >**
 - ▣ elemento to del tipo “#PCDATA”, no tiene hijos
- luego veremos atributos..

Formato de una DTD

9

- Para una DTD, un documento XML compuesto de:
 - ▣ Elementos
 - ▣ Atributos (y sus tipos)
 - ▣ Entidades (definidas por el usuario)
 - ▣ PCDATA
 - ▣ CDATA

PCDATA y CDATA

10

- PCDATA : “parsed character data”.
 - ▣ Es el texto que se encuentra entre las etiquetas de inicio y cierre de un elemento XML.
 - ▣ PCDATA es texto que será procesado y analizado por el parser, en busca de entidades o etiquetas..
 - ▣ no debería contener &, <, > ,etc, que deberían ser representados por las entidades & amp; & lt; and & gt;
- CDATA: character data.
 - ▣ CDATA es texto que no será analizado por el parser.

Un elemento puede contener

11

- Datos finales: PCDATA
 - ▣ `<!ELEMENT from (#PCDATA)>`
- Otros elementos (elemento contenedor):
 - ▣ Ejemplo: `<!ELEMENT note (to, from, heading, body)>`
 - Hijos que deben aparecer, en ese mismo orden
 - ▣ Faltan definiciones de cada uno de los elementos hijos
- Elemento vacío: no tiene texto, pero si puede tener atributos
 - ▣ `<!ELEMENT vacio EMPTY>`
- Elemento ANY: no se comprueba nada
 - ▣ `<!ELEMENT batiburrillo ANY>`

- Evitar si es posible, REDISEÑAR..

- Ejemplo:

```
<!ELEMENT estadoInv (#PCDATA | orderMsg )*>
```

- ▣ El elemento estadoInventario puede contener el número de elementos, o un elemento del tipo order con el estado del pedido.

- Debe existir un espacio en blanco antes y despues de la barra vertical.

```
<estadoInv>
```

```
10
```

```
  <orderMsg> pedido impresora </orderMsg>
```

```
</estadoInv>
```

DTD: Número de ocurrencias

13

- Una:
 - ▣ `<!ELEMENT note (message)>`
- , especifica una secuencia obligatoria de elementos
 - ▣ `<!ELEMENT alumno (nombre, apellidos, nota) >`
- + Al menos una:
 - ▣ `<!ELEMENT tablon (aviso+) >`
- * Cero o más:
 - ▣ `<!ELEMENT clase (alumno*)>`
- ? Cero o una, elemento opcional
 - ▣ `<!ELEMENT aviso (from, to, msg, priority?) >`
- | , pipe, o una u otra:
 - ▣ `<!ELEMENT note (to, from, (message | body))>`

Ejercicio (sin atributos)

14

- Hacer DTD para ejemplo tablon compraventa
 - ▣ Incluir DOCTYPE en fichero xml
 - ▣ Probar si es válido con editor (XMLcopyEditor o Notepad++)
 - ▣ Comprobar que navegadores NO validan
 - ▣ Modificar el fichero xml para que falte o sobre algún elemento.
 - Comprobar que NO es válido, pero
 - Si debe estar “bien formado”
- Hacer DTDs para otros ejemplos: clase.xml

+ sobre ejercicio tablón

15

- Puede haber cero anuncios..
- Texto (descripción) opcional, pero si aparece, sólo una vez
 - ▣ Probar que valida si no aparece, pero no si aparece dos veces..
- Contacto obligatorio, pero puede aparecer más de uno.
 - ▣ Comprobar que no valida sin el contacto, pero que permite más de uno
- Añadir elemento opcional: fecha límite o número de días que está activo el anuncio..

+ejercicios: la clase

16

- ❑ Quitar atributos
- ❑ Usar versión con notas -> Evaluación -> (id, nota)
- ❑ La clase puede estar vacía.
- ❑ Nombre y apellidos obligatorios y únicos. Fecha opcional. Mail uno o más. Notas puede no haber. Evaluaciones cero o más.

DTD: atributos

17

Pueden aparecer en la definición
varios atributos asociados al mismo
elemento

□ Sintaxis:

```
<!ATTLIST nombreElemento nombreAtributo  
          tipoAtributo valorpordefecto >
```

□ Ejemplo

▣ DTD contiene...

```
<!ATTLIST pago tipo CDATA "contado">
```

▣ XML contiene ...

```
<pago tipo="contado"> lavadora</pago>
```

```
<pago tipo="tarjeta"> ordenador</pago>
```

Valor por defecto: posibilidades

18

- “valor” // Valor por defecto
- #REQUIRED
- #IMPLIED
- #FIXED “valor”
- O enumerado:
 - ▣ (valor1 | valor2 ..)

Vamos a verlos uno a uno..

Atributos pueden ser declarados (I)

19

- **#REQUIRED:** El atributo es obligatorio.

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

Valid XML: `< alumno numero="1 2" />`

Invalid XML: `< alumno />`

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

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

Valid XML: `<contacto fax="956345678" />`

Valid XML: `<contacto />`

Atributos pueden ser declarados (II)

20

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

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

Valid XML: < envio empresa ="MS" />

Invalid XML: < envio empresa ="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, ○ #IMPLIED

Valor por defecto del atributo

21

- Si no se especifica en el elemento se le asigna ese valor
 - ▣ ver luego presentación en navegadores, no lo usan..

DTD:

```
<!ELEMENT cuadrado EMPTY>
```

```
<!ATTLIST cuadrado lado CDATA "0">
```

Valid XML: <cuadrado lado ="100" />

Valid XML: <cuadrado/>

Ejercicio sobre tablon.xml

22

- Crear DTD para que el atributo :
 - ▣ vendido del elemento anuncio sea obligatorio
 - ▣ moneda_pago de precio sea opcional
 - ▣ lang (idioma) del anuncio sea fijo, de valor spa
 - ▣ visible del elemento contacto tenga los valores si o no.
- Resolver paso a paso: probarlos de uno en uno
 - ▣ Comprobar la validez de ficheros xml con/sin atributos.
 - ▣ Comprobar la validez o no de ficheros que cumplan o no dichas restricciones
- Presentar los resultados en Firefox/IE .. ¿diferencias?
- Modificar DTDs de ejercicios anteriores para incluir los atributos en la DTD

Tipos de atributos (I)

23

□ CDATA

- ▣ puede contener cualquier carácter si éste se atiene a las reglas de formación.

□ NMTOKEN

- ▣ sólo puede contener letras, dígitos, punto [.], guión [-], subrayado [_] y dos puntos [:] .

□ NMTOKENS pueden contener los mismos caracteres que NMTOKEN más espacios en blanco. Un espacio en blanco consiste en uno o más espacios, retornos de carro o tabuladores.

Tipos de atributos (I): Ejemplo

24

- Ni el espacio ni la # están permitida si el atributo es NMTOKEN

```
<!ATTLIST attributes aaa CDATA #IMPLIED  
bbb NMTOKEN #REQUIRED  
ccc NMTOKENS #REQUIRED>
```

```
<attributes bbb="AB C" aaa="#d ."   
ccc="C D" />
```

Correcto,
NMTOKENS

Incorrecto, espacio

correcto, # Y
ESPACIO

Tipos de atributos (II)

25

□ ID

- El valor de un atributo de tipo ID puede contener sólo caracteres válidos en NMTOKEN y debe comenzar con una letra.
- Ningún tipo de elemento puede tener especificado más de un atributo de tipo ID. El valor de un atributo ID debe ser único entre todos los valores de atributos ID.
 - El atributo ID no puede comenzar con un número ni contener caracteres no permitidos en NMTOKEN:
 - El atributo ID debe tener un valor único, incluso entre varios atributos del tipo ID.

Ejercicio con tablon.dtd

26

- Modificar para que el atributo moneda_de_pago sea del tipo NMTOKEN.
 - ▣ Comprobar que con espacios, o comenzado con #, o /, da error
- Modificar para añadir al elemento anuncio un atributo obligatorio de nombre codigo y de tipo ID
 - ▣ Comprobar que sigue sintaxis NMTOKEN
 - ▣ Comprobar que no se puede tener dos anuncios con el mismo código

Tipos de atributos (III)

27

□ **IDREF**

- ▣ El valor de un atributo IDREF tiene que corresponder con el valor de algún atributo ID del documento.

□ **IDREFS**

- ▣ El valor del atributo IDREFS puede contener varias referencias a elementos con atributos ID separados por espacios en blanco.
- Y algunos otros que ni nombramos

Recuerda: atributos:

28

- + de un atributo

```
<!ATTLIST anuncio vendido CDATA #REQUIRED  
                codigo ID #REQUIRED >
```

- Enumerados: poner valor por defecto, IMPLIED, REQUIRED

```
<!ATTLIST contacto visible ( si | no ) #REQUIRED >
```

```
<!ATTLIST contacto visible ( si | no ) #IMPLIED>
```

```
<!ATTLIST contacto visible ( si | no ) "no">
```

Entidades

29

- Variables para usar abreviaturas o atajos
 - DTD:
`<!ENTITY writer "Donald Duck.">`
 - XML:
`<author>&writer;</author>`
- Se dice externa si la referencia es una URL/URI
 - DTD Example:
`<!ENTITY writer SYSTEM "http://www.w3schools.com/entities.dtd">`
 - XML example:
`<author>&writer;</author>`
- Ventaja: se puede compartir en muchas docs xml, pero sólo hay que actualizar en una localización.
- Comprobar..
 - XMLcopyEditor la valida correctamente, pero los navegadores no la visualizan o incluso dan error.

Ejercicio

30

- Generar una dtd para un sistema de mensajes o avisos. Cada mensaje:
 - Remitente
 - Uno o más destinatarios
 - Texto de la nota
 - Atributos:
 - cada mensaje lleva un atributo que será urgente o normal. Por defecto normal. Obligatorio.
 - Fecha de la nota. Del tipo NMTOKEN. Ponerlo primero opcional, luego obligatorio..
 - Atributo del asunto: idioma, por defecto es. Valores posible: es, en y fr
 - Atributo id de la nota, del tipo ID.
- Crear un documento xml, sms.xml con varios mensajes y probar
 - Si está bien formado
 - Añadir la referencia a la DTD y ver si es válido.
- Ver diferencia al presentar con Firefox Internet Explorer con complemento (ver valores por defecto). Ver código fuente.
- Modificar sms.xml para que no cumpla la DTD y comprobar.

Ejercicio clase

31

- Hacer DTD para validar el ejemplo de clase.xml
- Añadir al fichero clase.xml
 - ▣ Atributo repetidor al alumno. Obligatorio, con valores si y no.
 - ▣ Atributo opcional idioma en alumno.
 - ▣ Atributo visible al mail del alumno. Tipo NMTOKEN
 - ▣ Atributo código a la asignatura. Tipo ID. Obligatorio.
- Añadir antes de los alumnos los profesores. Con datos nombre y apellidos.
 - ▣ Anadir a los profesores un atributo asignaturas para indicar si es tutor o no.

Resumen

32

- Necesidad de comprobar la estructura, atributos, etc.
- Varias formas de hacerlo, una de ellas con DTD
- Debemos saber y comprender cómo
 - ▣ Definir elementos de los distintos tipos
 - ▣ Ocurrencias de los elementos
 - ▣ Atributos
 - REQUIRED, IMPLIED, etc.
 - Tipos de los atributos
- Ejercicios más completos: ahora podemos restringir formato, etc..