TALLER DTD Y XML SCHEMA

1. Especifique los tipos de datos soportados por DTD y XML SHEMA?

R=

* 1. **Schema XML**

Los tipos soportados por el Schema XML se pueden clasificar en los siguientes:

1. Tipos Simples Predefinidos.
2. ***xsd:decimal***

Un elemento con tipo simple decimal puede ser cualquier número positivo o negativo.

1. ***xsd:integer***

Un elemento con tipo simple integer puede ser cualquier número entero positivo, negativo o cero.

1. ***xsd:positiveInteger***

Un elemento con el tipo simple positiveInteger puede ser cualquier número entero positivo, pero no puede ser negativo ni cero.

1. ***xsd:negativeInteger***

Un elemento con el tipo simple negativeInteger puede ser cualquier número entero negativo pero ni positivo ni cero.

1. ***xsd:nonPositiveInteger***

Un elemento con el tipo simple nonPositiveInteger puede ser cualquier número negativo o cero, pero no positivo.

1. ***xsd:nonNegativeInteger***

Un elemento con el tipo simple nonNegativeInteger puede ser cualquier número positivo o cero, pero no negativo.

1. ***xsd:float***

Un elemento con el tipo simple float puede ser un número de coma flotante de 32 bits, como 3.5 ó 4.5e+08. También puede ser infinito positivo (INF), infinito negativo (-INF), y “no es un número” (NaN).

1. ***xsd:double***

Un elemento con el tipo simple double puede ser un número de coma flotante de 64 bits, con el mismo formato que float.

1. Tipos simples basados en fechas y horas.
2. ***xsd:date***

Un elemento date debe tener el formato AAAA – mm – dd .

1. ***xsd:time***

Un elemento time debe tener el formato hh:mm:ss.sss.

1. ***xsd:timeInstant***

Este tipo permite combinar data y time en una sola cadena.

1. ***xsd:timeDuration***

Este tipo es un poco raro. El formato es: PnYnMnDnTHnMnSn. “P” es igual al periodo y es de uso obligatorio. Las otras letras estarán presentes solo si se van a usar, excepto la “T” que también debe aparecer.

1. ***xsd:month***

Este tipo especifica un mes y año. El formato es AAAA – MM.

1. ***xsd:year***

Este tipo especifica un año. Su formato es AAAA.

1. ***xsd:century***

Este tipo especifica un siglo; se marca como los dos primeros dígitos de los cuatro de un año, incrementados en uno. No hace falta recordarlo pero a veces la inercia nos confunde.

1. ***xsd:recurringDate***

Este tipo especifica un mes y un día, pero no el año. El formato es el siguiente: -- MM – DD. Observe el guión doble al principio.

1. ***xsd:recurringDay***

Este tipo especifica un día en el mes. El formato es: --- DD.

1. Tipos simples variados:

1. ***xsd:boolean***

Este tipo especifica “verdadero” (true) o “falso” (false); también puede usarse el 0 y el 1.

1. ***xsd:language***

Este tipo especifica un idioma, en el formato de dos caracteres establecido por la norma ISO639. Por ejemplo, el inglés es “EN”, y el español es “ES”.

1. ***xsd:uri-reference***

Aunque se escriba así, este tipo no especifica – todavía – un URI (Universal Resource Identifier, identificador universal de recursos), sino un URL (Uniform Resource Locator, localizador uniforme de recursos).

1. ***xsd:NMTOKEN***

Este tipo obliga al texto a ser un nombre XML válido

* 1. **Tipos de datos DTD**

Los tipos de soportados por el DTD se especifican en la siguiente tabla

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Sintaxis** |
| CDATA | El valor está formado por caracteres. | **<!ELEMENT nombre CDATA 'valor' >** |
| (en1|en2|..) | El valor debe ser uno de los que forma la lista. | **<!ELEMENT nombre (en1 | en2 | ...) 'en2' >** |
| ID | El valor es un ID único. | **<!ELEMENT nombre ID 'valor' >** |
| IDREF | El valor es un ID de otro elemento. | **<!ELEMENT nombre IDREF 'valor' >** |
| IDREFS | El valor es una lista de ID de otros elementos. | **<!ELEMENT nombre IDREFS 'valor' >** |
| NMTOKEN | El valor es un nombre XML válido. | **<!ELEMENT nombre NMTOKEN 'valor' >** |
| NMTOKENS | El valor es un nombre XML de una lista válida. | **<!ELEMENT nombre NMTOKENS 'valor' >** |
| ENTITY | El valor es una entidad que hace referencia a otras entidades del XML. | **<!ENTITY nombre 'valor' >** |
| ENTITIES | El valor es una entidad tomada de una lista de entidades. | **<!ENTITIES nombre 'valor' >** |
| NOTATION | El valor es el nombre de una notación. | **<!ELEMENT nombre NOTATION 'valor' >** |
| XML | El valor es un valor xml predefinido. | **<!ELEMENT nombre 'valor' >** |

1. Es posible definir otro tipo de dato en DTD y XML SHEMA

R=

* 1. **Schema XML**

En el Schema XML, se pueden crear tipos de datos personalizados cuando no queremos amoldarnos a ninguno de los anteriores, donde tenemos unas reglas exactas. Para ello lo que hacemos es coger un tipo simple predefinido como base y, a partir de él, hacemos las restricciones oportunas para modelarlo a nuestro problema.

Para realizar la creación de estos nuevos tipos de datos se utiliza el elemento *restriction*. El formato de este elemento es: *<xsd:restriction base=“base”>*, donde base es el tipo simple sobre el que queremos imponer la restricción. A continuación se especifican las características particulares del tipo (utilizando las propiedades que veremos a continuación). Para finalizar la restricción utilizamos la correspondiente etiqueta de cierre </xsd:restriction>.

Ejemplo:

*<xsd:restriction base=“xsd:string”>*

*<xsd:pattern value=“(\d{3})\s\d{3} - \d{4}” />*

*</xsd:restriction>*

1. Especifique las restricciones y reglas para la definición de una estructura en DTD y XML SHEMA

R=

* 1. **DTD – Definición de tipo de Documento**

Un archivo DTD representa una gramática que describe qué etiquetas y atributos son válidos dentro de un documento XML, y en qué contexto son válidos. Un DTD se compone de una serie de declaraciones. Cada declaración va encerrada entre los caracteres <! y >, y se clasifican en cuatro tipos de declaraciones:

1. **Declaración de elemento**

Una declaración de elemento se emplea para definir un elemento nuevo (definición de etiquetas) y especificar su contenido válido. La sintaxis de esta declaración es:

*<!ELEMENT nombre contenido >*

Restricciones:

* El nombre de un elemento debe comenzar con una letra, el carácter de subrayado ‘\_’ o dos puntos ‘:’, seguido de uno de esos caracteres o también dígitos u otros caracteres de puntuación (‘.’ y ‘-‘). No hay límites a la longitud del nombre.
* El contenido de un elemento puede ser:
* Nada. Se dice que es un elemento vacío. Se indica poniendo en el contenido la palabra clave EMPTY. Por ejemplo: *<!ELEMENT imagen EMPTY>*
* Sólo otros elementos hijo. Si puede contener cualquier otro elemento existente en el DTD, se indica con la palabra clave ANY, aunque lo normal es indicar un conjunto de elementos con un modelo de grupo. Por ejemplo: *<!ELEMENT p ANY>*
* Sólo texto.
* Una mezcla de elementos hijo y texto.
* Un elemento que puede contener elementos hijo, texto, o ambas cosas, puede ser que alguna vez no tenga ningún contenido. En ese caso, es correcto emplear tanto la etiqueta inicial y final sin contenido (<titulo></titulo>) o una etiqueta vacía (<titulo/>).
* Un modelo de grupo va encerrado entre paréntesis y al menos debe contener el nombre de otro elemento. Se pueden controlar dos aspectos del modelo de grupo: la secuencia de aparición de los elementos y la secuencia de cantidad (cuánto se repite cada uno de ellos).
* La secuencia de aparición se controla con los conectores **‘,’ -** conector de secuencia y **‘|’** - conector de elección.

La secuencia de cantidad permite indicar cuantas veces puede aparecer un elemento en cada sitio. Para ello se emplean los indicadores de cantidad:

? - El elemento puede o no puede aparecer (es opcional) 0 ó 1

+ - El elemento es requerido, pero puede repetirse 1 a N

\* - El elemento puede no aparecer o aparecer varias veces (es opcional pero también se puede repetir) 0 a N.

1. **Declaración de atributo**

Los atributos de los elementos se declaran separadamente del elemento, en la declaración de atributos. Normalmente, todos los atributos asociados con un elemento se declaran todos juntos en una única declaración; cuando haya más de una declaración, se combinan todas ellas en una sola. Cuando existan varias declaraciones del mismo atributo, la primera de las definiciones tiene precedencia sobre las demás.

Restricciones

* Una declaración de atributos se identifica por la palabra reservada ATTLIST, seguida del nombre del elemento al que hace referencia, más una serie de parejas o tríos que indican el nombre del atributo, el tipo de atributo y el valor por defecto (opcional):

*<!ATTLIST elemento atributo1 contenido1 defecto1 atributo2 contenido2 defecto2 ...>*

* El nombre del atributo sigue las mismas reglas que los nombres de los elementos. El tipo de atributo restringe el rango de valores que puede contener. Los tipos de atributos existentes son:
* CDATA

Una cadena de texto (no impone ninguna restricción).

* NMTOKEN

Un token (palabra), con las mismas limitaciones que los nombres de los elementos y los atributos, excepto que no existen restricciones con el primer carácter (puede comenzar por un dígito).

* NMTOKENS

Una serie de tokens separados por espacios en blanco.

* ENTITY

El valor del atributo es una referencia a una entidad.

* ENTITIES

Una serie de referencias a entidades separadas por espacios en blanco.

* ID

Sirve para identificar a un elemento. Todos los atributos ID de un documento han de contener valores diferentes de modo que sea posible identificar cada elemento por su ID. Cada elemento sólo puede tener un atributo de este tipo.

* IDREF

Empleado para facilitar los enlaces a hipertexto: sirve para crear referencias otros elementos a partir de su ID.

* IDREFS

Una serie de referencias a valores ID separadas por espacios en blanco.

* NOTATION

Indica que tipo de datos embebidos pueden aparecer dentro de un elemento.

* Grupo de tokens

Restringe el valor a un conjunto finito de valores (lista de enumerados). Por ejemplo, la definición (left | center | right) especifica que el valor debe ser uno de los indicados:

<table align=”center”>

* Los atributos obligatorios se denominan atributos requeridos y se indican con la palabra reservada #REQUIRED.
* Los atributos opcionales se denominan atributos implícitos y se indican con la palabra reservada #IMPLIED. Los atributos también pueden tener un valor por defecto: cuando no se indica un valor, se emplea el valor por defecto definido. No se puede definir un valor por defecto con los atributos requeridos o implícitos.
* Por último, si un valor por defecto va precedido de la palabra reservada #FIXED, entonces el valor por defecto es el único valor que puede tomar el atributo .

1. **Declaración de entidad**

Las declaraciones de entidad permiten asociar un nombre con un fragmento de texto, de modo que cuando se haga referencia al nombre, se sustituirá por el fragmento de texto. Las ventajas que nos ofrece esta característica son:

Restricciones:

* Una referencia a una entidad no puede estar modificada directamente por un indicador de cantidad (?, +, \*). Si se quiere modificar, tiene que ir encerrado entre paréntesis.
* Al principio y al final del texto que se sustituye se añade un espacio en blanco para asegurar que la entidad contiene únicamente tokens completos.

1. **Declaración de notación**

Las declaraciones de notación son utilizadas para especificar el tipo de los ficheros binarios referenciados mediante las referencias a entidades externas no textuales. Esta información se pasa directamente sin analizar y es por tanto la aplicación la que debe decidir qué hacer con esta información.

Una declaración típica de este tipo es:

*<!NOTATION GIF87A SYSTEM “GIF”>*

* 1. **Schema XML**

1. Especifique librerías para utilizar DTD y XML SHEMA en (java, C#, Python, JavaScript PHP, C++)

R=

En la página <http://www.w3.org/XML/Schema> , encontraremos una serie de librerías que permiten la implementación del DTD y el Schema XML, sobre los diferentes lenguajes. Ejemplo:

1. Liquid XML 2008

Liquid XML Studio 2008 Released - A Freeware XML Development Environment 2007-10-23

Code Generation, C++, C#, .Net, Java, VB .Net & VB6 (trial)

Download available from http://www.liquid-technologies.com/Product XmlStudio.aspx

2. NetBeans Schema support

Netbeans Enterprise Pack Download: http://www.netbeans.org/products/

Referencia bibliográfica

<http://www.tic2.org/WebTecnica/Programacion/DTD/DTDTiposAtributos/DTDTiposAtributos.htm>

<http://www.tic2.org/Librerias/Librerias.htm>

<http://www.hipertexto.info/documentos/dtds.htm>

<http://www.w3.org/XML/Schema>

<http://bibing.us.es/proyectos/abreproy/11387/fichero/VOLUMEN+1%252FCapitulo4.pdf>