

A. Lenguajes para la visualización de información (continuación) :

- Semejanzas y diferencias entre HTML Y XHTML

En primer lugar, la mayor similitud que les caracteriza es su procedencia. Los lenguajes de marcas proceden de SGML del que deriva directamente HTML y XML, y a partir de este último deriva XML.

Un punto a favor de HTML es que se puede visualizar un documento en el navegador aunque este contenga algunos fallos, mientras que en XHTML ha de cumplir todas las reglas para que se muestre. Para continuar en XHTML no puede haber texto suelto, sin embargo en HTML sí. XHTML es algo más estricto con las mayúsculas y minúsculas ya que los nombres de elementos o atributos han de escribirse en mayúsculas y las referencias a entidades en minúsculas estrictamente algo que con HTML no pasa. En los documentos XHTML encontraremos siempre un único elemento raíz <html> sin embargo los documentos HTML no es necesario. En XHTML los elementos que tienen etiqueta de apertura deben tener etiqueta de cierre mientras que HTML no los necesita. En conclusión podemos observar que XHTML es un lenguaje más estricto y que HTML deja algo mas de libertad.

- Sintaxis de las etiquetas XHTML

En XHTML se deben utilizar solamente las etiquetas y los atributos que sirven para definir y saber que es cada cosa y no las que sirven para definir el aspecto de cada cosa. Etiquetas – minúsculas, <etiqueta /> tenemos en cuenta el espacio para que funcione correctamente en todos los navegadores.

- Utilidad de XHTML en los sistemas de gestión de información

XHTML es utilizado para generar documentos y contenidos en forma de hipertexto que una vez publicados en la web adoptan una forma general.

- Modelo de objetos del documento DOM.

El modelo de objetos del documento representa un documento y nos permite cambiarlo, navegar por ellos o por su estructura, añadir elementos o modificarlos y trabajar sobre aquellos que sean abstractos. Es un modelo jerárquico formado por nodos que se relacionan entre si, hay distintos tipos de nodos, pero todos mantienen unas características básicas. En conclusión esta API es realmente útil para tener todo bajo control y poder modificar o corregir de manera mas ágil y eficiente.

B. Lenguajes para el almacenamiento y transmisión de información:

- Tipos de lenguajes: de marcas (XML); de listas (JSON).

Los lenguajes de marcas sirve en definitiva para almacenar datos con un formato en concreto que luego es interpretado por el agente de usuario. En concreto XML como ya he explicado permite asignar atributos a etiquetas, utiliza un esquema para definir de forma exacta las etiquetas y los atributos y la estructura y el diseño son independientes. Actualmente está siendo sobrepasado en la mayoría de sus usos por JSON.

A diferencia de los lenguajes de marcas, los lenguajes de listas como JSON son formatos estándar abiertos, versátiles que se utilizan para serializar, intercambiar y transmitir datos estructurados y complejos entre un servidor y una aplicación web, JSON en concreto esta basado en el lenguaje de programación Javascript, sin embargo es completamente independiente de este. JSON es mas pequeño, más rápido y mas fácil de interpretar que XML.

C. Introducción al XML

- Objetivos de XML

XML es un lenguaje de texto plano y eso lo hace útil para guardar la información y transmitirla por lo tanto se usa mucho para intercambiar información, al igual que también se utiliza mucho para estructurar documentos. XML se limita a estructurar la información y permite que otros programas la utilicen. Es de uso general y al ser un metalenguaje no tiene un conjunto ya definido de etiquetas sino que permite definir elementos con estructuras más adaptadas a tus necesidades.

- Reglas sintácticas. Partes de un documento XML

Las principales partes que definen un documento XML son, en primer lugar una declaración XML que es la sentencia que declara que un documento es XML y no cualquier otro, en segundo lugar encontraremos una declaración de tipo documento que enlaza el documento con su DTD (definición de tipo documento), el DTD puede estar incluido en la propia declaración y por ultimo uno o más comentarios e instrucciones para procesarlo. El cuerpo debe tener solo un elemento raíz, después encontraremos algunos elementos que pueden tener contenido o no y esos elementos tienen atributos que irán entre comillas y por ultimo se encuentran los caracteres CDATA que permite que caracteres especiales o rompan esa estructura.

Ejemplo básico :

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

    <NOMBRE>RAQUEL</NOMBRE>
    <APELLIDO1>GARCIA</APELLIDO1>
    <EMAIL>raquel@gmail.com</EMAIL>
    <TELEFONO>654747515</TELEFONO>

</CLIENTES>
```

- Etiquetas

Encontramos tres tipos de etiquetas las de apertura, las de cierre y las etiquetas vacías. Cada etiqueta de apertura debe tener su etiqueta de cierre para su buen funcionamiento y deben estar al mismo nivel en el documento. Las etiquetas vacías no contienen nada estas deben contener una barra inclinada “/” antes de su cierre

- Nodos

Los nodos o conjunto de información se organizan basándose en una jerarquía que permite a los desarrolladores buscar información específica y que esto se convierta en una labor más sencilla, esto como ya he explicado antes, proporciona una API que permite agregar, mover, editar o eliminar nodos en el árbol jerárquico.

- Ignorar partes del documento

(<!--y -->) se utiliza para crear comentarios en XML sin que estos sean interpretados por el navegador. Son de gran utilidad para crear notas, para orientar a alguien que tenga que trabajar sobre el después etc.

- Juego de caracteres

En XML hay algunos caracteres que pueden ser de gran utilidad, por ejemplo un conjunto de caracteres se corresponde con un carácter concreto y un número Z-90. La codificación establece como se representan estos códigos numéricos en bytes. Sólo hay 5 entidades predefinidas: **<**, **>**, **&**, **'** (') y **"** ("). Unicode es un estándar que da cobertura a los lenguajes.

- Herramientas para comprobar que un documento esté bien formado

Para validar un documento XML utilizaremos **DTD** que es el formato de esquema nativo para validarlos y también encontramos **XSD** o **XML schema** que es un lenguaje de esquema más complejo y más potente que se basa en la gramática para proporcionar una potencia expresiva mayor.

- Uso de los espacios de nombres

Los espacios de nombres en xml se usan para proporcionar datos con un nombre único para evitar la ambigüedad que produce que elementos o atributos se llamen igual se les da un espacio de nombres a estos. Por ejemplo imaginemos que un cliente solicita un producto, pues tanto el cliente como el producto tendrán asociados un id para evitar problemas y que sea menos ambiguo.

Referencias:

<https://www.w3.org/2005/03/DOM3Core-es/introduccion.html>

http://www.codexemplar.org/curso/curso_4_3.php

<https://sites.google.com/site/dwebhtml/tutorials/xhtmll#TOC-Creando-un-documento-XHTML>

<https://sites.google.com/site/todoxmldtd/>

https://validator.w3.org/#validate_by_upload

