**XML**

Sirve para transmitir información que pueda ser interpretada por una maquina

El xml es un lenguaje que se utiliza fundamentalmente para almacenar información y/o enviarla en formato texto plano.

El leer un xml recibe el nombre de parseo.

Formato estandar

<persona>

<nombre>Jorge</nombre>

<apellido>Sánchez</apellido>

</persona>

Formato JSON

Se utilizan para las aplicaciones RSS, en las que a través de una URL se te notifican las actualizaciones que se han implementado en la pagina para que puedas verlos sin necesidad de un navegador web e internet. Lo puedes abrir con bloc de notas o cualquier programa pero algunas aplicaciones, que están preparadas para parsealos te lo interpreta para mostrarlo de una forma mas visual, y no el texto organizado en ramas.

Cumplen un protocolo llamado DTD(**D**efinición de **T**ipo de **D**ocumento), el cual lo defines tú. NO es necesario que un xml tenga un dtd explicito. En caso de que figure el dtd dentro del documento tiene que ir al principio del documento xml. Esa parte del documento se llama el prólogo.

Lo primero que se manda en xml es como debes interpretar la información. Con utf-8, con Unicode, con ASCII. Ejemplo:

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

A continuación, irá el (único) elemento raíz, elemento obligatorio que se abre al principio y cierra al final del documento. Engloba todo el documento vaya

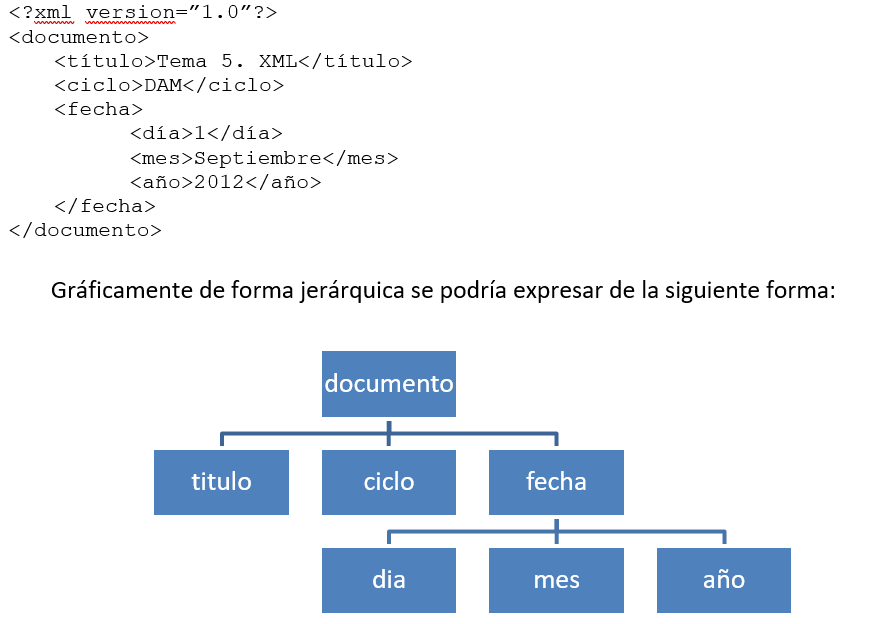
Texto- puede estar vacio. No se pueden usar caraceteres que podrían dar lugar a condusion como son los signos separadores.

Entidades: consiste en un nombre y su valor (su equivalente con programación es la constante) . Se definen mediante la etiqueta ENTITY.

<!ENTITY dam "Desarrollo de Aplicaciones Multiplataforma">

Una referencia a una entidad empieza con el caracater & y termina con ; . un ejemplo seria:

<ciclo>&dam;</dam>



Existen varias entidades predefinidas en el sistema que se utilizan para caracteres especiales que tienen otro significado dentro del lenguaje xml(>,<,”,&.’). Sirven como comandos de escape.

Reglas

Los textos de las etiquetas SÍ distinguen entre mayúsculas y minúsculas, por lo que hay que tenerlo en cuenta a la hora de referenciarlo. Deben empezar por una letra, aunque luego pueden utilizarse números.

Importante diferenciar entre elemento y etiqueta: el elemento es la porción que se abre y se cierra y lo que contiene en su interior. La etiqueta es cada una de las llaves, pero sin el contenido. En el ejemplo de arriba el elemento nombre sería: <nombre>Jorge</nombre>, mientras que la etiqueta sería <nombre>

Algunos elementos pueden estar vacíos de contenido, en ese caso la etiqueta de abrir y cerrar se puede fusionar en <nombre />

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

<alumno sexo="varon" fechaNacimiento="5/6/1990">

<nombre>Pablo</nombre>

<apellido>Pérez</apellido>

<telefono tipo="movil">622443322</telefono>

<direccion>Ronda de Burgos 83</direccion>

</alumno>

Para hacer esto hay que entrar en insetar🡪 en la parte de ilustraciones, Smart art🡪 jerarquía

Pagina para vaidar: https://www.w3schools.com/xml/xml\_validator.asp

Ejercicio 1

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

<coleccion>

<libro1>

<titulo>Manolito gafotas</titulo>

<autores>Elvira Lindo</autores>

<editorial>Alfaguara</editorial>

<fechadepublicacion>2/03/1995</fechadepublicacion>

<ISBN>es-2376rt343498</ISBN>

</libro1>

<libro2>

<titulo> mi vecino totoro</titulo>

<autores>Hayao Miyazaki</autores>

<editorial>EstudioGibli </editorial>

<fechadepublicacion></fechadepublicacion>

<ISBN></ISBN>

</libro2>

<libro3>

<titulo></titulo>

<autores></autores>

<editorial></editorial>

<fechadepublicacion></fechadepublicacion>

<ISBN></ISBN>

</libro3>

</coleccion>

Para que un documento XML esté bien formado cuando cumplen las siguientes reglas:

1. No se pueden utilizar los símbolos >,<,&,’,´´. Ya que tienen otros significados dentro del código.
2. En cuanto a los elementos, todo el documento debe depender de un único elemento raíz, deben abrirse y cerrarse las etiquetas. Los nombres de los elementos deben comenzar por letra. Los nombres de los elementos no pueden comenzar con el texto: ``xml´´tanto en minúscula o mayúsculas
3. En cuanto a los atributos siguen las mismas normas que con los elementos
4. Los atributos se tienen que colocar en las etiquetas de apertura, no en las de cierre. Y sus valores deben ir entrecomillados (ya sea simple o doble).
5. Las etiquetas pueden ir sin dato asociado pero los atributos no.

Un documento **xml válido** para que cualquier programa pueda interpretarlo es aquel que:

Se debe definir un DTD (Definicion de Tipo de Documento).

Al principio metemos !DOCTYPE y el nombre del elemento raiz

Ejemplo:

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

<!DOCTYPE persona [

<!ELEMENT persona (nombre)>

<!ELEMENT nombre (#PCDATA)>

]>

<!DOCTYPE raiz SYSTEM "http://www.empresa.com/docs.dtd">

<!ENTITY nombreEntidad "valorEntidad">

<!ENTITY pi “3.141592”>

<!ENTITY textFile SYSTEM "fichero.txt" >

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo EMPTY>

]>

Nombre elemento, donde se va a especificar que tipo de contenido va a haber.

- EMPTY

- (#PCDATA):Puede tener texto

-ANY: Puede contener cualquier cosa, ya sea texto como el PCDATA u otros elementos como otras etiquetas

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo ANY>

]>

-Si el documento debe tener una lista de elementos, estos se deben indicar separados por comas

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo (a, b)>

<!ELEMENT a EMPTY>

<!ELEMENT b EMPTY>

]>

Elementos incorrectos:

<ejemplo><a/><b/><c/></ejemplo> <!-- No es correcto porque contiene un elemento <c /> -->

<ejemplo><a/></ejemplo> <!-- No es correcto porque falta el elemento <b /> -->

<ejemplo><b/><a/

Ejemplos de elementos:

Elementos incorrectos:

<ejemplo>Esto es un ejemplo</ejemplo> <!-- No es correcto porque contiene texto -->

<ejemplo><a></a></ejemplo> <!-- No es correcto porque contiene un elemento <a> -->

Los elementos pueden contener elecciones opcionales. Para indicarlo se utiliza la barra vertical

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo (a | b)>

<!ELEMENT a EMPTY>

<!ELEMENT b EMPTY>

]>

Combinaciones de los elementos anteriores

Elementos incorrectos:

<ejemplo><a/><b/></ejemplo> <!-- No es correcto porque están los dos elementos -->

<ejemplo></ejemplo> <!-- No es correcto porque no hay ningún elemento -->

Elementos incorrectos:

<ejemplo><a/><b/></ejemplo> <!-- No es correcto porque están los dos elementos -->

<ejemplo></ejemplo> <!-- No es correcto porque no hay ningún elemento -->

<!ELEMENT coordUniversal (#PCDATA)>

]

Sería válido este documento:

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

<!DOCTYPE coordenada SYSTEM "coordenada.dtd">

<coordenada>

<longitud>234</longitud>

<latitud>-23</latitud>

</coordenada>

Sería válido también:

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

<!DOCTYPE coordenada SYSTEM "coordenada.dtd">

<coordenada>

<coordUniversal>1232332</coordUniversal>

</coord

La cardinalidad es el numero de veces que puede aparecer un determinado contenido en un elemento. Se indica mediante estos símbolos.

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo (a, b?)>

<!ELEMENT a EMPTY>

<!ELEMENT b EMPTY>

]>

Elementos incorrectos:

<ejemplo><b/></ejemplo> <!-- No es correcto porque falta el elemento <a/> -->

<ejemplo><b/><b/></ejemplo> <!-- No es correcto porque el elemento <b/> aparece dos veces -->

El asterisco (\*) sirve para explicar que es necesario 0 o mas elementos

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo (a\*, b)>

<!ELEMENT a EMPTY>

<!ELEMENT b EMPTY>

]>

<ejemplo><b/></ejemplo>

<ejemplo><a/><b/></ejemplo>

<ejemplo><a/><a/><b /></ejemplo>

El símbolo (+) especifica que el elemento tiene que aparecer una vez como mínimo

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo (a+, b)>

<!ELEMENT a EMPTY>

<!ELEMENT b EMPTY>

]>

Elementos correctos:

<ejemplo><a/><b/></ejemplo>

<ejemplo><a/><a/><b /></ejemplo>

Elemento incorrecto:

Para nombrar atributos:

<!ATTLIST nombreElemento nombreAtributo tipoAtributo valorPorDefecto>

Para nombrar atributos de un mismo elemento hay 2 formas de hacerlo:

<!ATTLIST nombreElemento nombreAtributo1 tipoAtributo1 valorInicialAtributo1>

<!ATTLIST nombreElemento nombreAtributo2 tipoAtributo2 valorInicialAtributo2>

Equivalente a:

<!ATTLIST nombreElemento

nombreAtributo1 tipoAtributo1 valorInicialAtributo1

nombreAtributo2 tipoAtributo2 valorInicialAtributo2

tipos de atributo:

CDATA: Cualquier tipo de caracteres:

Por ejemplo:

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo EMPTY>

<!ATTLIST ejemplo color CDATA #REQUIRED>

]>

Atributos correctos:

<ejemplo color=""></ejemplo>

<ejemplo color="amarillo"></ejemplo>

<ejemplo color="azul marino #000080"></ejemplo>

Atributo incorrecto:

<ejemplo></ejemplo> <!-- No es correcto porque falta el atributo "color" -->

ID: sirve para identificar el elemento que lo contiene, por lo que debe ser único a todos los elementos. Los valores de estos atributos no pueden empezar por número.

Ejemplo:

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo (libro\*)>

<!ELEMENT libro (#PCDATA) >

<!ATTLIST libro codigo ID #REQUIRED>

]>

Atributos correctos:

<ejemplo>

<libro codigo="L1">Poema de Gilgamesh</libro>

<libro codigo="L2">Los preceptos de Ptah-Hotep</libro>

</ejemplo>

Atributos incorrectos:

<ejemplo>

<libro codigo="1">Poema de Gilgamesh</libro> <!-- No es correcto porque el valor de un atributo de tipo ID no puede empezar con un número -->

<libro codigo="L2">Los preceptos de Ptah-Hotep</libro>

</ejemplo>

<ejemplo>

<libro codigo="L1">Poema de Gilgamesh</libro>

<libro codigo="L1">Los preceptos de Ptah-Hotep</libro> <!-- No es correcto porque el valor "L1" ya se ha utilizado -->

</ejemplo>

IDREF: El valor del atributo debe coincidir con el valor del atributo ID de otro elemento. Es decir, será una referencia a otra etiqueta. Por ejemplo:

<!DOCTYPE ejemplo [

<!ELEMENT ejemplo ((libro|prestamo)\*)>

<!ELEMENT libro (#PCDATA)>

<!ATTLIST libro codigo ID #REQUIRED>

<!ELEMENT prestamo (#PCDATA) >

<!ATTLIST prestamo libro IDREF #REQUIRED>

]>

Atributos correctos:

<ejemplo>

<libro codigo="L1">Poema de Gilgamesh</libro>

<prestamo libro="L1">Numa Nigerio</prestamo>

</ejemplo>

Atributos incorrectos:

<ejemplo>

<libro codigo="L1">Poema de Gilgamesh</libro>

<prestamo libro="L2">Numa Nigerio</prestamo> <!-- No es correcto porque el valor "L2" no es ID de ningún elemento -->

</ejemplo>

IDREFS: