

XML

-eXtensible Markup Language (Lenguaje de Marcas Extensible).

-Lenguaje de etiquetas.

-Separa el contenido de la presentación.

-Estándar de intercambio de datos entre diferentes programas de una manera segura, fiable y libre.

-Ver chuleta de XML.

Qué es XML

- Es un lenguaje de Marcas. Los contenidos están delimitados por etiquetas.
- Es un metalenguaje. Se pueden definir las etiquetas necesarias.
- Define una sintaxis general para maquetar datos.
- Es un formato estándar y flexible para documentos informáticos que puede ser adaptado al campo de aplicación que se desee.
- Es un formato de texto plano, muy adecuado para almacenar información y transmitirla.
- Un archivo XML contiene etiquetas y datos.

Que no es XML

- No es un protocolo de comunicación. No enviará datos por nosotros.
- No es un sistema gestor de bases de datos. Aunque una base de datos relacional pueda contener campos del tipo XML, incluso pueden existir bases de datos nativas XML
- No es un lenguaje de programación. No se pueden generar ejecuciones a partir de un documento XML.

¿Para que sirve XML?

- Es muy utilizado para intercambio de información. Ya que xml es texto plano es adecuado para almacenar información y transmitirla.
- Es muy utilizado para estructurar documentos. Podríamos decir que, lo mismo que las bases de datos relacionales son adecuadas para almacenar y tratar datos, XML es adecuado para almacenar y tratar documentos.
- XML no hace nada por si mismo. Estructura la información y permite que otros programas la utilicen.

Tecnologías XML

- XML. Lenguaje de marcas de uso general.
- Validación XML:
 - DTD. Sistema de validación de ficheros XML.
 - XSD, XML Schema. Sistema de validación de ficheros XML.
- XSL. Permite describir como se muestra XML. se compone de;
 - XSLT. Permite la transformación de documentos XML:
 - XSL-Fo. Indica como se van a mostrar los datos en pantalla. es un lenguaje de maquetado
 - XPath. Permite buscar y seleccionar en XML para XSLT
- XQuery. Parecido a SQL con capacidades de programación:

Chuleta de XML

{Abrirllave.com – Tutoriales de informática

Qué es XML

- XML (*eXtended Markup Language*).
- Desarrollado por **W3C** (*World Wide Web Consortium*).
- Basado en **SGML** (*Standard Generalized Markup Language*).
- Utilizado para el almacenamiento e intercambio de datos estructurados entre distintas plataformas.
- Es un metalenguaje empleado para definir otros lenguajes (dialectos XML): GML, MathML, RSS, SVG, XHTML, etc.

Elementos

- Un documento XML está formado por texto plano (sin formato) y contiene marcas (etiquetas) definidas por el desarrollador.
- Sintaxis: `<etiqueta>valor</etiqueta>`
- Elementos vacíos: `<etiqueta></etiqueta>` o `<etiqueta />`
- Un elemento (padre) puede contener a otro u otros elementos (hijos).
- Todo documento XML tiene que tener un único elemento raíz (padre) del que desciendan todos los demás. Ejemplo:

```
<persona>
  <nombre>Elsa</nombre>
  <mujer/>
  <fecha-de-nacimiento>
    <día>18</día>
    <mes>6</mes>
    <año>1996</año>
  </fecha-de-nacimiento>
  <ciudad>Pamplona</ciudad>
</persona>
```



- La estructura de cualquier documento XML se puede representar como un árbol invertido de elementos.
- Los elementos son los que dan estructura semántica al documento.
- Un elemento puede contener contenido mixto, es decir, texto y otros elementos.

Normas de sintaxis básicas

- Todos los nombres de los elementos son *case sensitive*, es decir, sensibles a letras minúsculas y mayúsculas.
- Pueden contener letras minúsculas, letras mayúsculas, números, puntos ".", guiones medios "-" y guiones bajos "_".
- Asimismo, pueden contener el carácter dos puntos ":". Pero, su uso se reserva para definir espacios de nombres.
- El primer carácter tiene que ser una letra o un guión bajo "_".
- Detrás del nombre de una etiqueta se permite escribir un espacio en blanco o un salto de línea.
- No puede haber un salto de línea o un espacio en blanco antes del nombre de una etiqueta.
- Los caracteres no ingleses (á, Á, ñ, Ñ...) están permitidos. Pero, al igual que el carácter *guión medio* "-" y el *punto* ".", se recomienda no utilizarlos para reducir posibles incompatibilidades o errores en programas que no los interpreten bien.

Atributos

- Un atributo sirve para proporcionar información extra sobre el elemento que lo contiene.
- Los nombres de los atributos deben cumplir las mismas normas de sintaxis que los nombres de los elementos.
- Todos los atributos de un elemento tienen que ser únicos.
- Los atributos contenidos en un elemento, deben separarse con espacios en blanco, no siendo significativo su orden.

Declaración XML

- Sintaxis: `<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`
- La declaración XML no es una instrucción de procesamiento (o proceso).
- Escribir la declaración XML es opcional. Pero, si se escribe, el atributo `version` es obligatorio indicarlo. Sin embargo, los atributos `encoding` y `standalone` son opcionales y, por defecto, sus valores son "UTF-8" y "no", respectivamente.
- `standalone="yes"` indica que el documento es independiente de otros.
- Cuando se escriba el atributo `encoding`, siempre deberá aparecer después de `version`. Y, respecto al atributo `standalone`, siempre que se escriba, deberá ser en último lugar.
- Si se incluye la declaración XML, tiene que aparecer en la primera línea del documento, y el carácter "<" debe ser el primero de dicha línea, es decir, antes no pueden aparecer espacios en blanco.

Instrucciones de procesamiento		
<ul style="list-style-type: none"> Una instrucción de procesamiento sirve para indicar cierta información al programa que procese dicho documento. Ejemplo: <code><?xml-stylesheet type="text/css" href="estilo-animales.css"?></code> En un documento XML, no es obligatorio que aparezcan instrucciones de procesamiento. 		
Referencias a entidades		
< (<i>menor que</i>)	lt (<i>less than</i>)	<
> (<i>mayor que</i>)	gt (<i>greater than</i>)	>
" (<i>comilla doble</i>)	quot (<i>quotation mark</i>)	"
' (<i>comilla simple</i>)	apos (<i>apostrophe</i>)	'
& (<i>ampersand</i>)	amp (<i>ampersand</i>)	&
<ul style="list-style-type: none"> El carácter <i>menor que</i> "<" es problemático porque indica el comienzo de una etiqueta. El carácter <i>ampersand</i> "&" es problemático, ya que, se utiliza para indicar el comienzo de una referencia a entidad. Uso de la comilla doble (") y de la comilla simple (') en atributos: <pre><dato caracter="comilla doble(&quot;)" /> <dato caracter='comilla simple(&apos;)' /></pre> Los valores de atributos escritos entre comillas dobles (") sí pueden contener al carácter comilla simple (') y a la inversa. <pre><dato caracter="comilla simple(')" /> <dato caracter='comilla doble("&quot;)" /></pre> 		
Referencias de caracteres		
<ul style="list-style-type: none"> Se pueden escribir referencias de caracteres Unicode con los símbolos &#, seguidos del valor decimal o hexadecimal del carácter Unicode que se quiera representar y, finalmente, añadiendo el carácter <i>punto y coma</i> ";". Ejemplo: el símbolo del Euro (€) puede escribirse <code>&#8364;</code> ó <code>&#x20AC;</code> 		
Comentarios		
<pre><!-- Esto es un comentario escrito en un documento XML --></pre> <ul style="list-style-type: none"> No se pueden escribir comentarios dentro de las etiquetas: <code><mujer <!-- elemento vacio --> /></code> En los comentarios no está permitido usar dos guiones seguidos. No es posible anidar comentarios en un documento XML. <pre><!-- Dos guiones seguidos -- en un comentario da error --></pre> 		
Secciones CDATA		
<ul style="list-style-type: none"> Un documento XML puede contener secciones CDATA para escribir texto que no se desea que sea analizado. Una sección CDATA se escribe comenzando con la cadena "<code><![CDATA[</code>" y terminando con los caracteres "<code>]]></code>". Dentro de una sección CDATA no se puede escribir la cadena "<code>]]></code>". Por tanto, no se pueden anidar secciones CDATA. No está permitido escribir espacios en blanco o saltos de línea en las cadenas de inicio "<code><![CDATA[</code>" o fin "<code>]]></code>". 		
Espacios de nombres		
<ul style="list-style-type: none"> Sintaxis para definir un espacio de nombres: <code>xmlns:prefijo="URI"</code> Los URI especificados en un documento XML no tienen porqué contener nada, su función es ser únicos. Ejemplos: <code>http://www.abrirllave.com/ejemplol</code> <code>http://www.w3.org/1999/xhtml/</code> Los espacios de nombres pueden definirse en el elemento raíz o, directamente, en los elementos que los vayan a utilizar. En un documento XML es posible definir todos los espacios de nombres que se necesiten, pudiéndose mezclar –si fuese necesario– los elementos de dichos espacios de nombres. Se puede definir un espacio de nombres por defecto mediante la siguiente sintaxis: <code>xmlns="URI"</code> En un documento XML, para indicar que determinados elementos –o todos– no pertenecen a ningún espacio de nombres, se escribe el atributo <code>xmlns</code> vacío, es decir, <code>xmlns=""</code>. 		
Espacios en blanco		
<ul style="list-style-type: none"> Los espacios en blanco, las tabulaciones y los retornos de carro pueden ser tratados de un modo especial. Los elementos pueden contener el atributo predefinido <code>xml:space</code> con el valor "preserve" para indicar que los espacios que aparecen en el contenido (texto) de dicho elemento, y sus sucesores, deben ser preservados. Los únicos valores que admite el atributo <code>xml:space</code> son "preserve" y "default", siendo este último su valor por defecto cuando no se escribe dicho atributo. "default" indica que la aplicación que haga uso del documento XML se encarga de decidir cómo tratar los espacios en blanco. Ahora bien, aún con "preserve", hay que tener en cuenta que no todos los programas reconocen este atributo. 		
Documentos XML bien formados y válidos		
<ul style="list-style-type: none"> Un documento XML está bien formado cuando no tiene errores de sintaxis. Un documento XML es válido cuando, además de no tener errores de sintaxis, no incumple ninguna de las normas establecidas en su estructura (DTD, XML Schema, RELAX NG...). 		

DTD: Definición de Tipo de Documento

-Opciones:

- DTD en el propio documento
 - **<!DOCTYPE nombre [**
 - **... declaraciones ...**
 - **]>**
- DTD en un documento externo:

<!DOCTYPE nombre SYSTEM "uri">

En todos estos casos:

- "nombre" es el nombre del tipo de documento XML, que debe coincidir con el nombre del elemento raíz del documento XML.
- "uri" es el camino (absoluto o relativo) hasta la DTD.

-Declaración de elementos

- Elementos:

<!ELEMENT nombreElemento (contenido)>

- Contenido:
 - **EMPTY**: elemento vacío.
 - **(#PCDATA)**: texto
 - **ANY**: cualquier cosa.
 - **,** (**coma**): elementos en el orden indicado.
 - **|** (**o lógico**): contiene uno de los dos elementos.
 - **?**: elemento puede aparecer o no, pero sólo una vez.
 - *****: elemento puede no aparecer o aparecer una o más veces.
 - **+**: elemento tiene que aparecer una o más veces (no puede no aparecer).
 - **()**: agrupar expresiones.

-Declaración de atributos

- Atributos:

```
<!ATTLIST nombreElemento nombreAtributo tipoAtributo valorInicialAtributo >
<!ATTLIST nombreElemento nombreAtributo1 tipoAtributo1 valorInicialAtributo1>
<!ATTLIST nombreElemento nombreAtributo2 tipoAtributo2 valorInicialAtributo2>
<!ATTLIST nombreElemento
  nombreAtributo1 tipoAtributo1 valorInicialAtributo1
  nombreAtributo2 tipoAtributo2 valorInicialAtributo2
>
```

en la que:

- "nombreElemento" es el nombre del elemento para el que se define un atributo.
- "nombreAtributo" es el nombre del atributo.
- "tipoAtributo" es el tipo de datos .
- "valorInicialAtributo" es el valor predeterminado del atributo (aunque también puede indicar otras cosas).
- Tipos de atributos:
 - **CDATA**: caracteres (sin restricciones).
 - **NMTOKEN**: letras, dígitos, y los caracteres punto ".", guión "-", subrayado "_" y dos puntos ":".
 - **NMTOKENS**: letras, dígitos, y los caracteres punto ".", guión "-", subrayado "_", dos puntos ":" (como el tipo NMTOKEN) y también espacios en blanco.

- **valores:** valores de una lista. Lista entre paréntesis, con términos separados por una barra vertical "|". términos entre comillas simples o dobles si contienen espacios en blanco.
- **ID:** valor no se puede repetir en otros elementos o atributos.
- **IDREF:** valor debe coincidir con el valor del atributo ID de otro elemento.
- **IDREFS:** valor es una serie de valores separados por espacios que coinciden con el valor del atributo ID de otros elementos.
- **ENTITY:** entidad definida en la DTD.
- **ENTITIES:** alguna de las entidades de una lista de entidades definida en la DTD.
- **NOTATION:** notación definida en la DTD.
- Valores iniciales:
 - **#REQUIRED:** el atributo es obligatorio, aunque no se especifica ningún valor predeterminado.
 - **#IMPLIED:** el atributo no es obligatorio y no se especifica ningún valor predeterminado.
 - **#FIXED valor:** el atributo tiene un valor fijo.
 - **valor:** el atributo tiene un valor predeterminado.