

REGLAS BÁSICAS DE SINTAXIS DE XML

Los documentos XML debe tener un elemento raíz

Los documentos XML deben contener un elemento **raíz** que es el **padre de todos los demás elementos**:

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

En este ejemplo **<note>** es el elemento raíz:

```
<?xml version="1.0" encoding="UTF-8 " ?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

El Prólogo XML

Esta línea se llama el **prólogo** XML:

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

El prólogo XML es opcional. Si existe, tiene que venir por primera vez en el documento.

Los documentos XML pueden contener caracteres internacionales, como la OA noruego o ee francés.

Para evitar errores, se debe especificar la codificación utilizada, o guardar los archivos XML como UTF-8.

UTF-8 es la codificación de caracteres por defecto para documentos XML.

UTF-8 es también la codificación predeterminada para HTML 5, CSS, JavaScript, PHP y SQL.

Todos los elementos XML debe tener una etiqueta de cierre

En XML, es ilegal para omitir la etiqueta de cierre. Todos los elementos **deben tener una etiqueta de cierre**:

```
<p>This is a paragraph.</p>
<br />
```

El prólogo XML no tiene una etiqueta de cierre.
Esto no es un error. El prólogo no es una parte del documento XML.

Etiquetas XML entre mayúsculas y minúsculas

Las etiquetas XML entre mayúsculas y minúsculas. La etiqueta <Letter> es diferente de la etiqueta <letter> .

las etiquetas de apertura y cierre deben ser escritos con el mismo procedimiento:

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

"Etiquetas de apertura y cierre" se refieren a menudo como "Start and end tags" .
Utilice lo que usted prefiere. Es exactamente lo mismo.

Elementos XML deben estar correctamente anidados

En XML, todos los elementos **must estar correctamente anidados unos dentro de otros**:

```
<b><i>This text is bold and italic</i></b>
```

En el ejemplo anterior, "Properly nested" significa simplemente que ya que el <i> elemento se abre dentro de la elemento, que debe estar cerrado dentro de la elemento.

Los valores de atributos XML podrán ser citados

elementos XML pueden tener atributos de pares de nombre / valor al igual que en HTML.

En XML, los valores de los atributos siempre deben ser citados.

INCORRECTO:

```
<note date=12/11/2007>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

CORRECTO:

```
<note date="12/11/2007">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

El error en el primer documento es que la date atributo en el elemento de nota no es citado.

Las referencias de entidad

Algunos caracteres tienen un significado especial en XML.

Si se coloca un personaje como "<" dentro de un elemento XML, generará un error porque el analizador lo interpreta como el inicio de un nuevo elemento.

Esto generará un error de XML:

```
<message>salary < 1000</message>
```

Para evitar este error, sustituya el "<" personaje con una **entity reference** :

```
<message>salary &lt; 1000</message>
```

Hay 5 predefinidos referencias a entidades en XML:

< < less than

> > greater than

& & ampersand

' ' apostrophe

" " quotation mark

Sólo <e Y son estrictamente ilegal en XML, pero es un buen hábito para reemplazar> con & gt; así como.

Los comentarios en XML

La sintaxis para escribir comentarios en XML es similar a la de HTML.

<! - Esto es un comentario ->

Dos guiones en el medio de un comentario no están permitidos.

No permitido:

```
<!-- This is a -- comment -->
```

Extraño, pero permitidas:

```
<!-- This is a - - comment -->
```

Espacios en blanco se conserva en XML

XML no trunca múltiples espacios en blanco (HTML trunca múltiples espacios en blanco a un solo espacio en blanco):

XML: Hello Tove

HTML: Hello Tove

Saltos de línea XML como LF

- Aplicaciones de Windows almacenan una nueva línea como: retorno de carro y avance de línea (CR+LF) .
 - Unix y Mac OS X utiliza LF.
 - Los viejos sistemas Mac utiliza CR.
 - XML almacena una nueva línea como LF.
-

XML bien formado

Los documentos XML que se ajustan a las reglas de sintaxis anteriores se dice que están "Well Formed" documentos XML.

Un documento XML contiene elementos XML.

¿Qué es un elemento XML?

Un elemento XML es todo, desde (including) inicio la etiqueta del elemento a (including) etiqueta final del elemento.

```
<price>29.99</price>
```

Un elemento puede contener:

- texto
- atributos
- otros elementos
- o una mezcla de los anteriores

```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

En el ejemplo anterior:

<title>, <author>, <year> y <price> tener **text content** , ya que contienen **texto (like 29.99)** .

<bookstore> y <book> tienen **element contents** , ya que contienen **elementos**.

<book> tiene un **attribute (category="children")** .

Elementos XML vacíos

Un elemento sin contenido se dice que es vacío.

En XML, puede indicar un elemento vacío de esta manera:

```
<element></element>
```

También puede utilizar un denominado etiqueta de cierre automático:

```
<element />
```

Las dos formas producen resultados idénticos en el software XML (lectores, analizadores, navegadores).

Los elementos vacíos pueden tener atributos.

Normas de denominación XML

elementos XML deben seguir estas reglas de nomenclatura:

- Los nombres de elementos entre mayúsculas y minúsculas
- Los nombres de elementos deben comenzar con una letra o un guión bajo
- Los nombres de elementos no pueden comenzar con el xml cartas (or XML, or Xml, etc)
- Los nombres de elementos pueden contener letras, dígitos, guiones, subrayados y puntos
- Los nombres de elementos no pueden contener espacios

Cualquier nombre se puede utilizar, no hay palabras están reservados (except xml)

.

Las mejores prácticas de nomenclatura

Crear nombres descriptivos, como esto: `<person>`, `<firstname>` , `<lastname>` .

Crear nombres cortos y simples, como esta: `<book_title>` no es como esto: `<the_title_of_the_book>` .

Evitar "-" . Si nombra algo "first-name" , algunos programas pueden pensar que desea restar "name" de la "first" .

Evitar "." . Si nombra algo "first.name" , algunos programas pueden pensar que "name" es una propiedad del objeto "first" .

Evitar ":" . Los dos puntos están reservados para los espacios de nombres (more later) .

cartas no están en inglés como EOA son perfectamente legales en XML, pero cuidado con los problemas si su software no es compatible con ellos.

Estilos de denominación

No hay estilos de denominación establecidos para los elementos XML. Pero aquí son algunas de uso común:

Estilo	Ejemplo	Descripción
Lower case	<firstname>	Todas las letras minúsculas
Upper case	<FIRSTNAME>	Todas las letras mayúsculas
Underscore	<first_name>	Subrayado separa las palabras
Pascal case	<FirstName>	Mayúscula la primera letra de cada palabra
Camello case	<firstName>	Mayúscula la primera letra de cada palabra, excepto el primero

Si elige un estilo de nomenclatura, es bueno para ser coherente!

Los documentos XML a menudo tienen una base de datos correspondiente. Una práctica común es utilizar las reglas de denominación de la base de datos de los elementos XML.

caso del camello es una regla de denominación común en JavaScript.

Elementos XML son extensibles

elementos XML pueden extenderse a transportar más información.

Mira el siguiente ejemplo XML:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```


Imaginemos que hemos creado una aplicación que extrae el `<a>`, `<desde>` y `<body>` elementos del documento XML para producir este resultado:

MESSAGE

To: Tove

From: Jani

No me olvides este fin de semana!

Imagine que el autor del documento XML añade alguna información adicional a ella:

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

En caso de que la aplicación romperse o fallar?

No. La aplicación debe todavía ser capaz de encontrar el `<to>`, `<from>` y `<body>` elementos del documento XML y producir la misma salida.

Esta es una de las bellezas de XML. Se puede extender sin romper aplicaciones.

elementos XML pueden tener atributos, al igual que en HTML.

Los atributos están diseñados para contener los datos relacionados con un elemento específico.

Los atributos XML podrá ser citado

Los valores de atributos siempre deben ser citados. De cualquier comillas simples o dobles se pueden utilizar.

Por sexo de una persona, la `<person>` elemento puede escribirse así:

```
<person gender="female">
```

o como esto:

```
<person gender='female'>
```

Si el valor del atributo en sí contiene comillas dobles se pueden utilizar comillas simples, como en este ejemplo:

```
<gangster name='George "Shotgun" Ziegler'>
```

o puede usar entidades de caracteres:

```
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

Elementos XML frente a Atributos

Echar un vistazo a estos ejemplos:

```
<person gender="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<person>
  <gender>female</gender>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

En el primer ejemplo de género es un atributo. En el pasado, el género es un elemento. Ambos ejemplos proporcionan la misma información. No hay reglas acerca de cuándo utilizar atributos o cuándo usar elementos en XML.

Mi forma favorita

Los siguientes tres documentos XML contienen exactamente la misma información:

Un atributo de fecha se utiliza en el primer ejemplo:

```
<note date="2008-01-10">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Un <date> elemento se utiliza en el segundo ejemplo:

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Un ampliado <date> elemento se utiliza en el tercer ejemplo: (THIS IS MY FAVORITE) :

```
<note>
  <date>
    <year>2008</year>
    <month>01</month>
    <day>10</day>
  </date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Evitar atributos XML?

Algunas cosas a considerar cuando se utiliza atributos son:

- Los atributos no pueden contener varios valores (elements can)
- Los atributos no pueden contener estructuras de árbol (elements can)
- atributos no son fácilmente ampliable (for future changes)

No terminan de esta manera:

```
<note day="10" month="01" year="2008"
to="Tove" from="Jani" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```

Atributos XML de metadatos

A veces, las referencias de identificación son asignados a los elementos. Estos identificadores se pueden utilizar para identificar elementos XML en mucho la misma manera que la id atributo en HTML. Este ejemplo muestra lo siguiente:

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

La id atributos anteriores son para la identificación de las diferentes notas. No es una parte de la propia nota.

Lo que estoy tratando de decir aquí es que los metadatos (data about data) deben ser almacenados como atributos, y los propios datos deben ser almacenados como elementos.