

XML

Contenido

Introducción.....	2
Estructura y sintaxis.....	2
CDATA.....	3
Validaciones XML	4
Normas de formación.....	4
Validez de XML.....	4
DTD.....	4

Introducción

XML es un metalenguaje, es decir, permite construir otros lenguajes y dialectos. A diferencia de los lenguajes de marcas, no define etiquetas, sino reglas

Es sencillo, robusto y de formato abierto

Actualmente se utiliza la quinta versión 1.0 de XML

Estructura y sintaxis

Un XML tiene la siguiente estructura:

- Declaración
 - o Versión
 - o Codificación
 - o Dependencias
- Contenido
 - o Raíz
 - Resto de elementos

Tiene una estructura de árbol, empezando en una raíz y teniendo distintos contenedores que, a su vez, pueden contener más contenedores

Las relaciones entre elementos son:

- Root: Elemento raíz
- Children: Elementos que se contienen dentro de un elemento padre
- Parent: Elementos que contienen otros elementos dentro
- Sibling: Elementos que tienen padre común

Los elementos tienen una etiqueta de apertura, una de cierre y atributos de forma opcional, compuestos de un parámetro y un valor

```
<nombre atributo1="valor1" atributo2="valor2">  
</nombre>
```

Los elementos pueden estar vacíos, tener texto o contener elementos. Además, siguen una serie de **reglas de asignación**:

1. Se diferencian mayúsculas de minúsculas
2. Pueden empezar con letras o guiones bajos
3. Los nombres de la apertura tienen que ser iguales a las del cierre
4. Los nombres pueden tener caracteres alfanuméricos, guiones, puntos y guiones bajos.
5. Los nombres no pueden tener espacios, ya que eso haría que el programa pensara que el resto es un atributo

Para crear atributos también hay unas reglas:

1. Deben tener un valor
2. Van entrecomillados, da igual si simples o dobles (deben ser igual tanto en apertura y cierre, eso sí)

3. Diferencian mayúsculas y minúsculas
4. Empiezan por una letra o un guión bajo
5. Pueden formarse de caracteres alfanuméricos, guiones, guiones bajos y puntos.

Los comentarios funcionan igual que en html, empezando con `<!--` y terminando en `-->`

```
▼<tabla1>
  <columna1>1</columna1>
  <columna2 id="2"/>
  ▼<columna3>
    <id> 3 </id>
    ▼<subcolumna_3_1>
      <id>3.1</id>
    </subcolumna_3_1>
  </columna3>
</tabla1>
<!-- Todo correctamente cerrado, y este comentario no importa donde esté -->
```

Al igual que HTML, cuenta con unas **ENTIDADES** para acceder al uso de caracteres que no se podrían usar de otra forma:

- < - <
- > - >
- & - &
- ' - ‘
- " - “

CDATA

La sección CDATA contiene caracteres que no se tienen en cuenta en el analizador:

Se inicia con `<![CDATA [` y termina con `]]>`

Validaciones XML

Hay dos formas de validar un XML:

- Por su forma, es decir, que cumple las normas de XML (no da error de ejecución)
- Por su validez, es decir, cumple unas reglas definidas por el creador o diseñador

Normas de formación

- Solo un elemento raíz
- Todos los elementos tienen que estar bien cerrados
- Los elementos tienen que estar anidados correctamente
- Todos los valores de los atributos entrecomillados
- Cada valor tiene que cumplir sus reglas

Lo bueno que tiene XML es que, al más mínimo fallo, salta un fallo al tratar de visualizar el archivo, dando la pista de dónde está ese fallo.

(Ejemplo quitando el cierre de id en la subcolumna 3.1;

This page contains the following errors:

error on line 12 at column 26: Opening and ending tag mismatch: id line 11 and subcolumna_3_1

Below is a rendering of the page up to the first error.

Validez de XML

Hay dos formas de validarlo:

- Cumpliendo las reglas de un DTD
- Cumpliendo las reglas de un XML-SCHEMA

DTD

Document Type Definition (Definición de Tipo de Documento) son una serie de reglas que sirven para validar un documento XML. Puede estar incluido en el propio documento XML (DTD Interno), o almacenado en otro documento externo y referenciarlo desde el XML por medio de `<!DOCTYPE nombre SYSTEM "nombreDTD">`

Entidades

Una entidad es una constante en programación: Se declara en el DTD y se usa en el XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fichas [
  <!ENTITY poblacion "Tejeda de Tiétar, Cáceres">
]>
<fichas>
  <ficha>
    <empresa nombre="Tractores Toledo"></empresa>
    <direccion>&poblacion;</direccion>
  </ficha>
</fichas>
```

Anotación

Se utiliza para identificar el formato de entidades que no son XML y que no se van a procesar como valor válido para un atributo. (Igual que todas las anotaciones en los lenguajes tanto de programación como de marcas)

Elemento

Un elemento de un DTD indica las condiciones que debe tener dicho elemento en el XML a validar.

```
<!ELEMENT nombre (condición)>
```

Tipos de contenido en un elemento:

- EMPTY: Vacío
- ANY: Puede tener cualquier contenido
- #PCDATA: Solo puede tener datos de tipo carácter (Parsed Character Data)
- (nombreElemento) solo puede contener ese elemento, teniendo en cuenta su **cardinalidad**:
 - o nombreElemento: Una única ocurrencia
 - o nombreElemento?: 0 o 1 ocurrencias
 - o nombreElemento+: 1 a N referencias
 - o nombreElemento*: 0 a N referencias
- (nombreElemento1, nombreElemento2): Debe contener todos los elementos que sigan las comas
- (nombreElemento1 | nombreElemento2): Debe contener uno u otro, (abre un condicional XOR)

Atributo

Los DTD también permiten restringir reglas y condiciones de los atributos en los documentos XML que validan

```
<!ATTLIST elemento nombreAtributo tipoAtributo valorAtributo>
```

Tipos de atributo:

- CDATA: Cadena de caracteres
- (valor1 | valor2...): Alguno de los valores escritos
- ID: Un identificador único
- IDREF: Una referencia a un solo identificador
- IDREFS: Una lista de referencias a identificadores de otros elementos
- NMTOKEN: Un nombre XML válido
- NMTOKENS: Una lista de nombres XML válidos separados por espacios
- ENTITY: Una entidad
- ENTITIES: Varias entidades
- NOTATION: nombre de una notación
- Xml:Lang: Idioma del contenido
- Xml:space: qué hacer con espacios, tabulaciones, etc

Valores de atributo

- 'Valor': Valor por defecto
- #REQUIRED: obligatorio
- #IMPLIED: Opcional
- #FIXED valor: Valor del atributo