

Chuleta de DTD

{Abrirllave.com – Tutoriales de informática

Qué es DTD

- **DTD** (*Document Type Definition*).
- Sirve para definir la estructura de un documento SGML o XML, permitiendo su validación.
- Un documento XML es válido (*valid*) cuando, además de estar bien formado, no incumple ninguna de las normas establecidas en su estructura.

Declaración de tipo de documento

- Una DTD se puede escribir tanto interna como externamente a un archivo XML.
- En ambos casos hay que escribir una definición **DOCTYPE** (*Document Type Declaration*, Declaración de Tipo de Documento) para asociar el documento XML a la DTD. Asimismo, un archivo XML se puede asociar simultáneamente a una DTD interna y externa.
- Sintaxis DTD interna:
`<!DOCTYPE elemento-raíz [declaraciones]>`
- Sintaxis DTD externa privada:
`<!DOCTYPE elemento-raíz SYSTEM "URI">`
- Sintaxis DTD externa pública:
`<!DOCTYPE elemento-raíz PUBLIC "identificador-público" "URI">`
- Sintaxis DTD interna y externa:
`<!DOCTYPE elemento-raíz SYSTEM "URI" [declaraciones]>`
`<!DOCTYPE elemento-raíz PUBLIC "identificador-público" "URI" [declaraciones]>`

Estructura de un documento XML

- Un documento XML será válido si –además de no tener errores de sintaxis– cumple lo indicado en las declaraciones de elementos, atributos, entidades y notaciones, de la DTD a la que esté asociado.

Declaración de elementos

- Sintaxis:
`<!ELEMENT nombre-del-elemento tipo-de-contenido>`
- En el **tipo-de-contenido** se especifica el contenido permitido en el elemento, pudiendo ser:
 - Texto, (**#PCDATA**).
 - Otros elementos (hijos).
 - Estar vacío, **EMPTY**.
 - Mixto (texto y otros elementos), **ANY**.
- Un elemento vacío puede tener atributos.
- Un elemento (padre) puede ser declarado para contener a otro u otros elementos (hijos). En la sintaxis, los hijos –también llamados sucesores– tienen que escribirse entre paréntesis “ () ” y separados por comas “ , ”.
- Los elementos (hijos) de un elemento (padre), deben escribirse en el mismo orden en el que han sido declarados en la DTD.
- Operadores de cardinalidad en DTD:
 - ? (interrogación): 0-1
 - * (asterisco): 0-n
 - + (signo más): 1-n
- Los elementos declarados en una DTD sobre los que no actúe ningún operador de cardinalidad, tendrán que aparecer obligatoriamente una única vez, en el o los documentos XML a los que se asocie.
- En la DTD asociada a un documento XML, se pueden declarar elementos que contengan elementos opcionales. Para ello, se utiliza el *operador de elección*, representado por una barra vertical (|).
- Al utilizar el operador de elección (|) en una DTD, si una de las opciones es **#PCDATA**, esta debe escribirse en primer lugar.

Declaración de atributos

- Sintaxis:
`<!ATTLIST nombre-del-elemento nombre-del-atributo tipo-de-atributo valor-del-atributo>`

Tipos de declaración de atributos

- **valor** entre comillas dobles (") o simples (').
- **#REQUIRED**
- **#IMPLIED**
- **#FIXED valor** entre comillas dobles (") o simples (').

Tipos de atributos
<ul style="list-style-type: none"> CDATA, Enumerado, ID, IDREF, IDREFS, NMTOKEN, NMTOKENS, NOTATION, ENTITY, ENTITIES, Especiales
Declaración de entidades
<ul style="list-style-type: none"> En una DTD se pueden declarar entidades generales y paramétricas (de parámetro). Las entidades generales pueden utilizarse en el cuerpo de un documento XML y en su DTD. Sin embargo, las entidades paramétricas solo pueden utilizarse dentro de la DTD. Sintaxis entidad general interna analizable: <code><!ENTITY nombre-de-la-entidad "valor-de-la-entidad"></code> Sintaxis entidad general externa analizable privada: <code><!ENTITY nombre-de-la-entidad SYSTEM "URI"></code> Sintaxis entidad general externa analizable público: <code><!ENTITY nombre-de-la-entidad PUBLIC "identificador-público" "URI"></code> Sintaxis entidad general externa no analizable privada: <code><!ENTITY nombre-de-la-entidad SYSTEM "URI" NDATA notación></code> Sintaxis entidad general externa no analizable pública: <code><!ENTITY nombre-de-la-entidad PUBLIC "identificador-público" "URI" NDATA notación></code> Sintaxis entidad paramétrica interna analizable: <code><!ENTITY % nombre-de-la-entidad "valor-de-la-entidad"></code> Las entidades paramétricas tienen que declararse antes de ser referenciadas. Las entidades paramétricas internas pueden declararse en DTD internas o externas. Sin embargo, no pueden referenciarse desde una DTD interna. Sintaxis entidad paramétrica externa analizable privada: <code><!ENTITY % nombre-de-la-entidad SYSTEM "URI"></code> <code>%nombre-de-la-entidad;</code> Sintaxis entidad paramétrica externa analizable pública: <code><!ENTITY % nombre-de-la-entidad PUBLIC "identificador-público" "URI"></code> <code>%nombre-de-la-entidad;</code> Una entidad se puede usar dentro de otra. La referencia circular o recursiva de entidades no es correcta.
Declaración de notaciones
<ul style="list-style-type: none"> Sintaxis notación privada: <code><!NOTATION nombre-de-la-notación SYSTEM "identificador-del-sistema"></code> Sintaxis notación pública: <code><!NOTATION nombre-de-la-notación PUBLIC "identificador-público"></code> <code><!NOTATION nombre-de-la-notación PUBLIC "identificador-público" "identificador-del-sistema"></code> En una DTD, pueden existir elementos con atributos cuyo valor sea el nombre de una notación.
Secciones condicionales
<ul style="list-style-type: none"> Sintaxis: <code><![IGNORE [declaraciones]]></code> <code><![INCLUDE [declaraciones]]></code>
Espacios de nombres
<ul style="list-style-type: none"> Ver ejemplo en: http://www.abrirlave.com/dtd/espacios-de-nombres.php
Comentarios
<ul style="list-style-type: none"> En una DTD asociada a un documento XML, se pueden escribir comentarios entre los caracteres “<!--” y “-->”.

Chuleta resumen del Tutorial de DTD de “Abrirlave.com” <http://www.abrirlave.com/dtd/> con Licencia Creative Commons ([CC BY 4.0](#)).