

Document Type Definition (DTD)

Una **definición de tipo de documento** o **DTD** (*document type definition*) es una descripción de estructura y sintaxis de un documento XML o SGML



The screenshot shows an XML editor window titled 'XMLFile2.xml' with a 'Start Page' tab. The editor displays an XML Document Type Definition (DTD) and an XML document. The DTD defines a 'school' document type with two main elements: 'principal' and 'ClassTeacher'. The 'principal' element has attributes 'name', 'age', and 'address'. The 'ClassTeacher' element has attributes 'cno', 'td', and 'nos'. The XML document below the DTD uses these elements and attributes, with some values like 'sdasd', '456', 'jkhk', 'gyj', 'fgd', and '45'.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE school [
  <!ELEMENT school (principal|ClassTeacher)*>
  <!ELEMENT principal (name,age,address)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT age (#PCDATA)>
  <!ELEMENT address (#PCDATA)>
  <!ELEMENT ClassTeacher (cno,td*,nos)>
  <!ELEMENT cno (#PCDATA)>
  <!ELEMENT td (name,dob)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT dob (#PCDATA)>
  <!ELEMENT nos (#PCDATA)>
]>

<school>
  <principal>
    <name>sdasd</name>
    <age>456</age>
    <address>jkhk</address>
  </principal>

  <ClassTeacher>
    <cno>456</cno>
    <td>
      <name>gyj</name>
      <dob>fgd</dob>
    </td>
    <nos>45</nos>
  </ClassTeacher>
</school>
```

Sintaxis:

Dependiendo de como se quiera introducir el DTD(interno o externo) la sintaxis se escribirá de forma diferente pero ambas tienen una estructura básica parecida:

```
<!DOCTYPE elemento (identificador DTD)
[
    declaration1
    declaration2
    .....
]>
```

En este ejemplo podemos ver algunas las características de DTD:

- El documento empieza con “<!DOCTYPE”
- Después llega el “elemento” que es el punto que se va a tomar como la raíz del texto.
- A continuación se pone el “identificador del DTD”. Esto sirve para los casos de tipo externo en el que el identificador va a ser una dirección URL o dirección de un archivo del equipo.
- Por último se ponen unos corchetes “[]” en el que se van a introducir diferentes atributos de los elementos del documento.

Maneras de implementación:

Hay dos maneras de implementar un documento DTD, externa y interna.

DTD interno:

Se dice que un DTD es interno cuando los elementos del DOCTYPE se declaran DENTRO del propio documento.

Esta sería un ejemplo de DTD interno:

```
<!DOCTYPE address [
    <!ELEMENT address (name,company,phone)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT company (#PCDATA)>
    <!ELEMENT phone (#PCDATA)>
]>
<address>
    <name>Tanmay Patil</name>
    <company>TutorialsPoint</company>
    <phone>(011) 123-4567</phone>
</address>
```

DTD externo:

En los DTD externos, los elementos se declaran fuera del archivo XML. Se puede acceder a ellos especificando los atributos del sistema que pueden ser o un archivo legal *.dtd* o una dirección URL válida.

```
<!DOCTYPE name SYSTEM "file-name">
```

Donde pone “file-name” es donde se pone el nombre del archivo *.dtd*. En esta caso ponemos SYSTEM pero también se puede poner PUBLIC.

Componentes

Hay tres componentes que se describen en DTD, **elementos**, **atributos** y **entidades**.

Elementos

Los documentos xml contienen al menos un elemento. Estos elementos contienen texto, atributos, objetos de comunicación o incluso otros elementos. Todos los elementos DTD tienen esta estructura

```
<!ELEMENT elementName (content)>
```

Donde pone “content” especificamos que contenido tiene el elemento y puedo haber 3 casos.

Contenido Vacío:

Como su nombre bien indica estos elementos no tienen contenido y se declaran con la palabra EMPTY.

```
<!DOCTYPE hr[
  <!ELEMENT address EMPTY>
]>
<address />
```

Contenido de elemento:

Estos elementos tienen como contenido otros elementos.

```
<!DOCTYPE address [  
  <!ELEMENT address (name,company,phone)>  
<address>  
  <name>Tanmay Patil</name>  
  <company>TutorialsPoint</company>  
  <phone>(011) 123-4567</phone>  
</address>
```

Dentro del contenido de elementos hay una lista de operadores:

- Operador (+): Indica que el elemento secundario se puede dar una o más veces dentro del elemento principal.

```
<!ELEMENTO nombre del elemento (elemento secundario1+)>
```

- Operador (*): Indica que el elemento secundario se puede dar cero o más veces dentro del elemento principal

```
°°<!ELEMENTO nombre del elemento (elemento secundario1*)>
```

- Operador (?): Indica que el elemento secundario se puede dar cero veces o una vez dentro del elemento principal.

```
<!ELEMENTO nombre del elemento (elemento secundario1?)>
```

- Operador (,): Da una secuencia de los elementos secundarios separados por comas, los cuales deben ser incluidos en el nombre del elemento.

```
<!ELEMENTO nombre del elemento (elemento secundario1, elemento secundario2)>
```

- Operador (|): Permite hacer elecciones en el elemento secundario.

```
<!ELEMENTO nombre del elemento (elemento secundario1 | elemento secundario2)>
```

Contenido de elemento mixto;

Esta es la combinación de (#PCDATA) y de sus elementos secundarios. PCDATA representa datos de carácter analizados, como por ejemplo texto que no está revisado.

```
<!DOCTYPE address [  
  <!ELEMENT address (#PCDATA|name)*>  
  <!ELEMENT name (#PCDATA)>  
<address>  
  Here's a bit of text mixed up with the child element.  
  <name>Tanmay Patil</name>  
</address>
```

Atributos

Los atributos son los componentes de DTD que definen una propiedad de los elementos y su declaración de no se diferencia mucho de ellos. Cuando declaras un atributo estás declarando una lista de atributos permitidos dentro de un elementos. A estas listas las llamamos ATTLIST.

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

Hay tres tipos de atributos:

- **Tipo de cadena** (CDATA)
- **Tipología de caso** (ID, IDREF, IDREFS, ENTIDAD, ENTIDADES, NMTOKEN, NMTOKENS)
- **Tipos enumerados** (ANOTACIÓN, Enumeración)

Valores de los atributos

Valores Predeterminados

```
<!ATTLIST element-name attribute-name attribute-type "default-value">
```

Los valores predeterminados se ponen entre comillas simples o dobles y sirven para poner un valor predeterminado.

Valores Fijos

```
<!ATTLIST element-name attribute-name attribute-type #FIXED "value" >
```

Para poner un valor fijo a un atributo hay que poner #FIXED y el valor fijo entre comillas como se puede ver en el ejemplo de arriba

Valores Requeridos

```
<!ATTLIST element-name attribute-name attribute-type #REQUIRED>
```

Si se quiere que el atributo sea utilizado utilizamos #REQUIRED

Valores IMPLIED

```
<!ATTLIST element-name attribute-name attribute-type #IMPLIED>
```

Los atributos tienen que llevar un valor si o si pero si no le queremos poner ninguno de los anteriores le ponemos #IMPLIED para poner no especificar un valor.

Entidades

Las entidades sirven para definir atajos de caracteres especiales para XML y se pueden especificar de dos formas:

Entidad Interna

Si la entidad es interna se declararía como otro componente más de DTD

```
<!ENTITY entity_name "entity_value">
```

Entidad externa

En cambio si queremos declarar la entidad de forma externa utilizamos la siguiente sentencia para referirnos al link en que tengamos nuestra entidad.

```
<!ENTITY name SYSTEM "URI/URL">
```

Y estas entidades se pueden especificar de dos maneras, con SYSTEM y con PUBLIC.

Dentro de las entidades se pueden diferenciar 4 tipos (integradas, de carácter, generales y de parámetro)

Entidades Integradas

Hay 5 tipos de entidades integradas que se utilizan en XML:

- &: se utiliza como la palabra 'y'.
- ': se utiliza como comillas simples.
- >: mejor que...
- <: menos que...
- ": se utiliza como comillas dobles

Entidades de Carácter

Estas entidades se utilizan para nombrar caracteres que no se pueden escribir como por ejemplo el símbolo del copyright.

```
<!DOCTYPE author[  
<!ELEMENT author (#PCDATA)>  
<!ENTITY writer "Tanmay patil">  
<!ENTITY copyright "&#169;">  
<author>&writer;&copyright;</author>
```

Entidades Generales

Estas entidades representan más de un carácter como párrafos o incluso documentos enteros.

```
<!ENTITY name "text">
```

Fuentes de información:

https://www.w3schools.com/xml/xml_dtd_intro.asp

<https://searchmicroservices.techtarget.com/definition/Document-Type-Definition-DTD>

https://www.tutorialspoint.com/es/dtd/dtd_syntax.htm

https://es.wikipedia.org/wiki/Definici%C3%B3n_de_tipo_de_documento