

Tema 4

Definition de esquemas y ficheros xml DTD

Tema 4 DTD

Tema 4

- 1.- Estructura y sintaxis de un documento XML
- 2.- DTD
- 3.- Elementos XML
- 4.- Atributos XML
- 5.- Entidades XML
- 6.- Anotaciones XML

1 Estructura y sintaxis de un documento XML

- Documento XML Básico está formado por :
 - Prologo
 - Informa al interprete de las características del documento
 - Definición de XML
 - `<?xml version="1.0" encoding="UTF-8"?>`
 - Declaración del tipo de documento
 - Representado por la cadena `<!DOCTYPE`
 - Ejemplar:
 - Contiene la estructura de los datos

Declaración del Tipo de Documento

- Permite al autor definir restricciones y características del documento
- Declaración del tipo de documento
 - Indica el nombre del tipo de documento
 - `<?xml version="1.0" encoding="UTF-8"?>`
- Definición del tipo de documento : DTD
 - Define las características de los elementos del documento
 - Elementos, atributos y notaciones
 - Restricciones y valores por defecto
 - Declaración DOCTYPE
 - Tipo de declaraciones
 - Internas
 - Externas

2.- DTD

- DTD internas
 - Declaraciones que afectan exclusivamente al documento
 - Se localizan dentro del propio documento XML
 - Enterradas entre []
 - <!DOCTYPE elemento-raíz [declaraciones]>
- Ejemplo **DTDInterno.xml**

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE marcadores [
3      <!ELEMENT marcadores (pagina)*>
4      <!ELEMENT pagina (nombre, descripcion, url)>
5      <!ELEMENT nombre (#PCDATA)>
6      <!ELEMENT descripcion (#PCDATA)>
7      <!ELEMENT url (#PCDATA)>
8  ]>
9
10 <marcadores>
11     <pagina>
12         <nombre>Abrirllave</nombre>
13         <descripcion>Tutoriales de informática.</descripcion>
14         <url>http://www.abrirllave.com/</url>
15     </pagina>
16     <pagina>
17         <nombre>Wikipedia</nombre>
18         <descripcion>La enciclopedia libre.</descripcion>
19         <url>http://www.wikipedia.org/</url>
20     </pagina>
21     <pagina>
22         <nombre>W3C</nombre>
23         <descripcion>World Wide Web Consortium.</descripcion>
24         <url>http://www.w3.org/</url>
25     </pagina>
26 </marcadores>
```

2.- DTD

- DTD Externas
 - Existen dos tipos DTD externas :
 - Publicas
 - Privadas
 - Privadas
 - <!DOCTYPE elemento-raíz SYSTEM "URI">
 - Publica
 - <!DOCTYPE elemento-raíz PUBLIC "identificador-público" "URI">

2.- DTD

- DTD Externas Privadas
 - Se define las características en un fichero extensión .dtd
 - **Ejemplo : marcadores.dtd**
 - Se añade al documento XML mediante la etiqueta SYSTEM

```
1 <!ELEMENT marcadores (pagina)*>
2 <!ELEMENT pagina (nombre, descripcion, url)>
3 <!ELEMENT nombre (#PCDATA)>
4 <!ELEMENT descripcion (#PCDATA)>
5 <!ELEMENT url (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE marcadores SYSTEM "marcadores.dtd">

<marcadores>
  <pagina>
    <nombre>Abrirllave</nombre>
    <descripcion>Tutoriales de informática.</descripcion>
    <url>http://www.abrirllave.com/</url>
  </pagina>
  <pagina>
    <nombre>Wikipedia</nombre>
    <descripcion>La enciclopedia libre.</descripcion>
    <url>http://www.wikipedia.org/</url>
  </pagina>
  <pagina>
    <nombre>W3C</nombre>
    <descripcion>World Wide Web Consortium.</descripcion>
    <url>http://www.w3.org/</url>
  </pagina>
</marcadores>
```

2.- DTD

- DTD Externas Publicas
 - Se incluyen mediante la etiqueta PUBLIC

```
<?xml version= 1.0 ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <title>Titulo</title>
  </head>
  <body>
    <p>Párrafo</p>
  </body>
</html>
```


2.- DTD

- DTD internas y externas
 - Interna
 - <!DOCTYPE elemento-raíz SYSTEM "URI" [declaraciones]>
 - Externa privada
 - <!DOCTYPE elemento-raíz PUBLIC "identificador-público" "URI" [declaraciones]>
 - Ejemplo:
 - **marcadoresDTDInternaExterna.xml**
 - **marcadoresInterno.dtd**

3.- espacio de nombres

- Espacios de nombres
 - Sintaxis
 - xmlns:prefijo="URI"
 - Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<e:ejemplo xmlns:e1="ejemplo1.xml"
  xmlns:e2="ejemplo2.xml">
  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>
  <e2:carta>
    <e2:carnes>
      <e2:filete_de_tenera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>
</e:ejemplo>
```

3.- Elementos XML

- Definición de la sintaxis de documentos XML
 - Criterios:
 - Condiciones para ser **elemento**
 - Contiene subestructuras.
 - Es de un tamaño considerable.
 - Su valor cambia frecuentemente.
 - Su valor va a ser mostrado a un usuario o aplicación.
 - Condiciones para ser **atributo**
 - El dato es de pequeño tamaño y su valor raramente cambia,
 - Existen casos espaciales
 - El dato solo puede tener unos cuantos valores fijos.
 - El dato guía el procesamiento XML pero no se va a mostrar.

3.- Elementos XML

- Definición de elemento
 - <!ELEMENT nombre-del-elemento tipo-de-contenido>
 - **Tipo de contenido**
 - Texto, (#PCDATA).
 - Otros elementos (hijos).
 - Estar vacío, EMPTY.
 - Mixto (texto y otros elementos), ANY



3.- Elementos XML

- **Definición de elemento**

- Tipo de contenido
 - Texto, (#PCDATA).
 - Indicamos que el elemento puede contener una cadena de caracteres analizable

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ciudad [
    <!ELEMENT ciudad (#PCDATA)>
]>

<ciudad>Roma</ciudad>
```



3.- Elementos XML

- **Definición de elemento**

- Tipo de contenido
 - Otros elementos (hijos).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ciudad [
    <!ELEMENT ciudad (nombre, pais)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT pais (#PCDATA)>
]>

<ciudad>
    <nombre>Roma</nombre>
    <pais>Italia</pais>
</ciudad>
```



3.- Elementos XML

- Definición de elemento
 - Tipo de contenido
 - Operador opción, ?.
 - Indica que el elemento no es obligatorio.
 - ejemplo el subelemento trabajo es opcional.
 - <!ELEMENT telefono (trabajo?, casa)
 - Operador uno-o-más, +.
 - Define un componente presente al menos una vez.
 - Ejemplo: definimos un elemento formado por el nombre de una provincia y otro grupo, que puede aparecer una o varias veces.
 - <!ELEMENT provincia (nombre, (cp, ciudad)+)

3.- Elementos XML

- **Definición de elemento**

- Tipo de contenido
 - Operador cero-o-mas, *.
 - Define un componente presente cero, una o varias veces.
 - En el ejemplo el grupo (cp, ciudad) puede no aparecer o hacerlo varias veces.
<!ELEMENT provincia (nombre, (cp, ciudad)*)
 - Operador de elección, | .
 - Cuando se utiliza sustituyendo las comas en la declaración de grupos indica que para formar el documento XML hay que elegir entre los elementos separados por este operador.
 - Ejemplo el documento XML tendrá elementos provincia que estarán formados por el elemento nombre y el cp (código postal), o por el elemento nombre y la ciudad.
<!ELEMENT provincia (nombre, (cp | ciudad))

3.- Elementos XML

- **Definición de elemento**

- Tipo de contenido
 - Estar vacío, EMPTY.
 - Los elementos vacíos no pueden tener contenido **PERO SI ATRIBUTOS**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE persona [
  <!ELEMENT persona (nombre, mayor_de_edad, ciudad)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT mayor_de_edad EMPTY>
  <!ELEMENT ciudad (#PCDATA)>
]>

<persona>
  <nombre>Elsa</nombre>
  <mayor_de_edad/>
  <ciudad>Pamplona</ciudad>
</persona>
```

3.- Elementos XML

- **Definición de elemento**

- Tipo de contenido
 - Mixto (texto y otros elementos), ANY

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE persona [
  <!ELEMENT persona ANY>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT ciudad (#PCDATA)>
]>

<persona>
  <nombre>Elsa</nombre> vive en <ciudad>Pamplona</ciudad>.
</persona>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE persona [
  <!ELEMENT persona ANY>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT ciudad (#PCDATA)>
]>

<persona>
  <nombre>Elsa</nombre> vive en Pamplona.
</persona>
```

3.- Elementos XML

- Definición de elemento
 - Tipo de contenido
 - Mixto (texto y otros elementos), ANY

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE persona [
  <!ELEMENT persona ANY>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT ciudad (#PCDATA)>
]>

<persona>
  <nombre>Elsa</nombre>
  ..
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE persona [
  <!ELEMENT persona ANY>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT ciudad (#PCDATA)>
]>

<persona/>
```

4.- Atributos

- Definición de atributos

- La sintaxis básica para declarar un atributo en una DTD es:

<!ATTLIST **nombre-del-elemento** **nombre-del-atributo** **tipo-de-atributo**
valor-del-atributo **modificadores**>

- Tipo de atributos:

- Cuando se declaran los atributos, se puede especificar cómo el procesador usará los datos que aparezcan en el valor. Se puede dividir la tipología de atributos en tres categorías:

- **Tipo de cadena**
 - **Tipología de caso (en inglés token)**
 - **Tipos enumerados**

4.- Atributos XML

Tipo	Descripción
CDATA	CDATA es un caracter de datos (texto y no revisión). Es un <i>tipo de atributo de cadena</i> .
Identificación (ID)	Es el único identificador del atributo. No debe aparecer más de una vez. Es un <i>Tipo de atributo de caso</i> .
IDREF	Se usa para hacer referencia a la identidad de otro elemento. Se utiliza para establecer conexiones entre los elementos. Es un <i>Tipo de atributo de caso</i> .
IDREFS	Se usa para referenciar múltiples identidades. Es un <i>Tipo de atributo de caso</i> .
ENTIDAD	Representa un entidad externa en el documento. Es un <i>Tipo de atributo de caso</i> .
ENTIDADES	Representa una lista de entidades externas en el documento. Es un <i>Tipo de atributo de caso</i> .

4.- Atributos XML

Tipo	Descripción
NMTOKEN	Es similar al CDATA y el valor del atributo consiste de un nombre XML válido. Es un <i>Tipo de atributo de caso</i> .
NMTOKENS	Es similar al CDATA y el valor del atributo consiste en una lista nombres XML válidos. Es un <i>Tipo de atributo de caso</i> .
ANOTACIÓN	Un elemento será referenciado a una anotación declarada en el documento DTD. Es un <i>Tipo de atributo enumerado</i> .
Enumeración	Permite definir un aspecto de la lista de valores donde uno de los valores debe coincidir. Es un <i>Tipo de atributo enumerado</i> .

4.- Atributos

- **CDATA** : el atributo contiene caracteres (sin restricciones).

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo EMPTY>  
  <!ATTLIST ejemplo color CDATA #REQUIRED>  

```



```
<ejemplo color="" />
```



```
<ejemplo color="amarillo" />
```



```
<ejemplo color="azul marino #000080" />
```



```
<ejemplo />
```

```
<!-- ERROR: falta el atributo "color", obligatorio debido al #REQUIRED -->
```



```
<ejemplo sabor="dulce" />
```

```
<!-- ERROR: el atributo "sabor" no está definido -->
```

4.- Atributos

- **NMTOKEN**: el atributo sólo contiene letras, dígitos, y los caracteres punto ".", guion "-", subrayado "_" y dos puntos ":"

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo EMPTY  
  <!ATTLIST ejemplo color NMTOKEN #REQUIRED  
>  
>
```



```
<ejemplo color="" />
```



```
<ejemplo color="azul-marino" />
```



```
<ejemplo color="1" />
```



```
<ejemplo color="azul marino" />
```

```
<!-- ERROR: hay un espacio en blanco -->
```



```
<ejemplo color="#F0F0F0" />
```

```
<!-- ERROR: contiene el carácter # -->
```


4.- Atributos

- **NMTOKENS**: el atributo sólo contiene letras, dígitos, y los caracteres punto ".", guion "-", subrayado "_", dos puntos ":" (como el tipo NMTOKEN) y también espacios en blanco.

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo EMPTY>  
  <!ATTLIST ejemplo color NMTOKENS #REQUIRED>  
>
```



```
<ejemplo color="" />
```



```
<ejemplo color="1" />
```



```
<ejemplo color="azul marino" />
```



```
<ejemplo color="2*2" />
```

```
<!-- ERROR: hay un asterisco -->
```

4.- Atributos

- **valores:** el atributo sólo puede contener uno de los términos de una lista. La lista se escribe entre paréntesis, con los términos separados por una barra vertical "|".

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo EMPTY>  
  <!ATTLIST ejemplo color (azul|blanco|rojo) #REQUIRED>  
>
```



```
<ejemplo color="" />
```



```
<ejemplo color="azul" />
```



```
<ejemplo color="verde" />
```

```
<!-- ERROR: "verde" no está en la lista de valores -->
```

4.- Atributos



- **ID:** el valor del atributo (no el nombre) debe ser único y no se puede repetir en otros elementos o atributos

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo (libro*)>  
  <!ELEMENT libro (#PCDATA) >  
  <!ATTLIST libro codigo ID #REQUIRED>  

```



```
<ejemplo>
  <libro codigo="L1">Poema de Gilgamesh</libro>
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>
</ejemplo>
```



```
<ejemplo>
  <libro codigo="1">Poema de Gilgamesh</libro>
  <!-- ERROR: el valor de un atributo de tipo ID no puede
  empezar con un número -->
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>
</ejemplo>
```



```
<ejemplo>
  <libro codigo="L1">Poema de Gilgamesh</libro>
  <libro codigo="L1">Los preceptos de Ptah-Hotep</libro> <!-- ERROR: no se puede repetir un atributo de tipo ID -->
</ejemplo>
```

4.- Atributos



- **IDREF**: el valor del atributo debe coincidir con el valor del atributo ID de otro elemento.

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo ((libro|prestamo)*)>  
  <!ELEMENT libro (#PCDATA) >  
  <!ATTLIST libro codigo ID #REQUIRED>  
  <!ELEMENT prestamo (#PCDATA) >  
  <!ATTLIST prestamo libro IDREF #REQUIRED>  

```



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <prestamo libro="L1">Numa Nigerio</prestamo>  
</ejemplo>
```



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <prestamo libro="L2">Numa Nigerio</prestamo>  
</ejemplo>
```

```
<!-- ERROR: el valor "L2" no es ID de ningún elemento -->
```

4.- Atributos



- **IDREFS**: el valor del atributo es una serie de valores separados por espacios que coinciden con el valor del atributo ID de otros elementos.

```
<!DOCTYPE ejemplo [  
  <!ELEMENT ejemplo ((libro|prestamo)*)>  
  <!ELEMENT libro (#PCDATA) >  
  <!ATTLIST libro codigo ID #REQUIRED>  
  <!ELEMENT prestamo (#PCDATA) >  
  <!ATTLIST prestamo libro IDREFS #REQUIRED>  

```



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L1 L2">Numa Nigerio</prestamo>  
</ejemplo>
```



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L1">Numa Nigerio</prestamo>  
</ejemplo>
```



```
<ejemplo>  
  <libro codigo="L1">Poema de Gilgamesh</libro>  
  <libro codigo="L2">Los preceptos de Ptah-Hotep</libro>  
  <prestamo libro="L3">Numa Nigerio</prestamo>  
</ejemplo>
```

<!-- ERROR: el valor "L3" no es ID de ningún elemento -->

4.- Atributos



Declaración del atributo

- En cada declaración de atributo, se debe especificar cómo aparecerá el valor en el documento. Se puede especificar si un atributo:
 - pueden tener un valor predeterminado
 - un valor fijo
 - se requiere
 - se asume
- Valores predeterminados
 - Contiene un valor predeterminado. Los valores se pueden rodear con comillas independientes (') o dobles("")
 - Sintaxis
<!ATTLIST element-name attribute-name attribute-type "default-value">

4.- Atributos

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA "0">
]>
<address>
  <name id="123">
    Tanmay Patil
  </name>
</address>
```

4.- Atributos

- Valores fijos
 - #palabras clave fijas seguidas del valor fijo se usan cuando se quiere especificar que el valor del atributo es constante y no se puede cambiar. Un uso común de atributos fijos se da especificando números de versión.
 - Sintaxis

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

```
<?xml version="1.0"?>
<!DOCTYPE address [
  <!ELEMENT address (company)*>
  <!ELEMENT company (#PCDATA)>
  <!ATTLIST company name NMTOKEN #FIXED "tutorialspoint">
]>
<address>
  <company name="tutorialspoint">we are a free online tea
</address>
```


4.- Atributos

- Valores requeridos

- Cuando se quiera especificar la necesidad de un atributo, se debe usar la palabra clave #REQUIRED.

- Sintaxis

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

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA #REQUIRED>
]>
<address>
  <name id="123">
    Tanmay Patil
  </name>
</address>
```

4.- Atributos

- Valores IMPLIED (asumidos/sobreentendidos)
 - Cuando se declaran los atributos siempre se debe especificar una declaración de valor.
 - Si el atributo que se declara no tiene un valor predeterminado, no tiene valor fijo, ni es requerido, entonces hay que declarar el atributo como sobreentendido.
 - La palabra clave #IMPLIED se usa para especificar un atributo como sobreentendido.
 - Sintaxis

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

Atributos

- En este ejemplo hemos usado la palabra clave #IMPLIED ya que no nos interesa especificar ningún atributos para ser incluido en elemento nombre. Es opcional

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA #IMPLIED>
]>
<address>
  <name />
</address>
```

4.- Atributos

5.- Entidades externas

- Entidades Externas

- Permiten establecer una relación entre el documento XML y otro documento a través de la URL de éste último.
- Privadas con la palabra SYSTEM
 - `<!ENTITY nombre_entidad SYSTEM "http://localhost/docsxml/fichero_entidad.xml">`
- Publicas con la palabra PUBLIC
 - `<!ENTITY nombre-de-la-entidad PUBLIC "identificador-público" "URI">`

5.- Entidades externas

- Entidades Externas
 - escritor.txt

Miguel de Cervantes

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE textos [
  <!ELEMENT textos (texto)+>
  <!ELEMENT texto (#PCDATA)>

  <!ENTITY escritor SYSTEM "escritor.txt">
]>

<textos>
  <texto>El Quijote fue escrito por &escritor;.</texto>
</textos>
```

5.- Entidades externas

- Entidades con Parámetros

- Permite dar nombres a partes de un DTD y hacer referencia a ellas a lo largo del mismo.
 - útiles cuando varios elementos del DTD comparten listas de atributos o especificaciones de contenidos.
 - Se denotan por %entidad;
 - <!ENTITY %direccion "calle, numero?, ciudad, cp">
 - <!ENTITY alumno (dni, %direccion;)>
 - <!ENTITY ies (nombre, %direccion;)>

6.- Notaciones

- Notaciones
 - Indica la aplicación que debe usarse cuando incluimos ficheros binarios
 - Sintaxis
 - <!NOTATION nombre SYSTEM aplicacion>
 - Ejemplo
 - <!NOTATION gif SYSTEM "gifEditor.exe">
 - Declaramos una notación llamada **gif** donde se indica la aplicación gifEditor.exe para visualizar imágenes

6.- Notaciones

- Notaciones

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE frutas [
  <!ELEMENT frutas (fruta)*>
  <!ELEMENT fruta EMPTY>
  <!ATTLIST fruta foto ENTITY #REQUIRED>

  <!ENTITY manzana SYSTEM "manzana.gif" NDATA gif>
  <!ENTITY naranja SYSTEM "naranja.gif" NDATA gif>

  <!NOTATION gif SYSTEM "image/gif">
]>

<frutas>
  <fruta foto="manzana"/>
  <fruta foto="naranja"/>
</frutas>
```

- Declaramos los valores para las entidades **naranja y manzana**
- Con NDATA asociamos a las entidades manzana y naranja la notación gif

6.- Notaciones

- Secciones condicionales
 - Permite incluir o ignorar partes de la declaración de un DTD
 - INCLUDE
 - permite que se vea esa parte de la declaración del DTD.
 - Su sintaxis es:
 - - <![INCLUDE [Declaraciones visibles]] >
 - - Ejemplo:
 - - <![INCLUDE [<!ELEMENT nombre (#PCDATA)>]] >
 - IGNORE
 - permite ocultar esa sección de declaraciones dentro del DTD.
 - Su sintaxis es:
 - - <![IGNORE [Declaraciones ocultas]] >
 - - Ejemplo:
 - - <![IGNORE [<!ELEMENT clave (#PCDATA)>]] >