

XML

Introducción a los lenguajes de marcas

XML (eXtensible Markup Language)

- Es un metalenguaje que permite diseñar lenguajes de marcas
- Desarrollado por el World Wide Web Consortium (W3C)
- Un documento XML sigue una estructura jerárquica de árbol
- Tiene dos partes: el prólogo y el cuerpo
- El prólogo está compuesto por las primeras líneas del documento, todas aquellas de tipo cabecera. Estas líneas se distinguen por empezar por <? y acabar por ?>
- El cuerpo es el resto del documento

Ejemplo

```
<?xml version="1.0" encoding="iso-8859-1" ?>  
<cine>  
  <pelicula titulos="El resplandor">  
    <director>Stanley Kubrick</director>  
    <duracion>146</duracion>  
  </pelicula>  
</cine>
```

Elementos XML

- Un elemento XML es todo lo que se define desde la apertura de una etiqueta hasta su cierre
- Cada elemento puede contener:
 - Texto
 - Atributos
 - Otros elementos anidados
- Algunos ejemplos

```
<director>Stanley Kubrick</director>
```

```
<director nombre="Stanley" apellido="Kubrick"></director>
```

```
<director>
```

```
  <nombre>Stanley</nombre>
```

```
  <apellido>Kubrick</apellido>
```

```
</director>
```

Elementos anidados y texto

- Un elemento puede tener otros elementos anidados
- Un elemento puede contener o bien texto, o bien otros elementos.
- El valor del atributo debe ir rodeado de comillas dobles o simples
- En el siguiente ejemplo el elemento película contiene un elemento anidado director, el cual, contiene un texto.

```
<pelicula titulo="El resplandor" duracion='146'>
```

```
  <director>Stanley Kubrick</director>
```

```
</pelicula>
```

Atributos

- Un elemento puede tener atributos además de otros elementos anidados o texto.
- Cada elemento solo puede definir un mismo atributo una sola vez
- El valor del atributo debe ir rodeado de comillas dobles o simples

```
<pelicula titulo="El resplandor" duracion='146'>  
</pelicula>
```

XML bien formados

- El W3C recomienda empezar por la cabecera que indica la versión de XML
- Todo documento ha de tener una estructura jerárquica en el cuerpo. Debe estar formado por un único elemento raíz, a partir del cual se define el resto. Este elemento debe ser el primero que se abre y el último que se cierra
- Todas las etiquetas deben estar correctamente anidadas unas dentro de otras.
- Los valores de los atributos deben estar rodeados de comillas simples o dobles.
- No puede haber más de un atributo con el mismo nombre en un mismo elemento.

XML bien formados

- XML es case sensitive, es decir, diferencia entre mayúsculas y minúsculas. Por esto, hay que cerrar las etiquetas con las mismas mayúsculas / minúsculas.
- Los nombres de etiquetas no pueden empezar por un número, ni usar caracteres especiales.
- Los comentarios no pueden ir dentro de la etiqueta
- Es posible crear elementos vacíos, es decir, sin texto ni subelementos utilizando una etiqueta de cierre automático (llevan una / después del nombre <element />)

Enumerar elementos

- Cuando queremos que un elemento defina como elementos hijos una lista de elementos del mismo tipo, podemos crear una etiqueta que contenga todos esos elementos del mismo tipo
- Dentro de esta etiqueta podemos poner cada uno de los elementos de la lista con sus respectivos atributos y subelementos

```
<pelicula titulo="El resplandor" duracion="146">  
  <protagonistas>  
    <protagonista interprete="Jack Nicholson">Jack Torrance</protagonista>  
    <protagonista interprete="Shelley Duvall">Wendy Torrance</protagonista>  
    <protagonista interprete="Danny Lloyd">Danny Torrance</protagonista>  
  </protagonistas>  
  <director>Stanley Kubrick</director>  
</pelicula>
```

Ejercicio

Conflictos en los nombres

- En ocasiones, puede ocurrir que elementos de distinto tipo compartan nombre, esto puede venir ocasionado porque se estén utilizando dos diccionarios de nombres distintos en un documento XML
- Por ejemplo, podríamos querer hacer un documento XML con el mobiliario urbano de una calle y los establecimientos de la misma.
- Entre los establecimientos podría haber bancos (entidades financieras) y entre el mobiliario urbano podría haber también bancos (de sentarse)
- Los nombres de los establecimientos y del mobiliario vienen definidos por dos entidades distintas y en ambos casos utilizan la palabra "banco" para designar tanto la entidad financiera como el de sentarse

Conflictos en los nombres

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

```
<Calle>
```

```
  <Nombre>Alcalá</Nombre>
```

```
  <Localidad>Madrid</Localidad>
```

```
  <PuntosDeInteres>
```

```
    <Punto>
```

```
      <Km>1.15</Km>
```

```
      <Banco>Santander</Banco>
```

```
    </Punto>
```

```
    <Punto>
```

```
      <Km>1.2</Km>
```

```
      <Banco>De madera blanco</Banco>
```

```
    </Punto>
```

```
  </PuntosDeInteres>
```

```
</Calle>
```

Namespaces

- Los namespaces son un mecanismo que resuelve la ambigüedad que se puede producir cuando varias etiquetas de distinto tipo tienen el mismo nombre, pero distinta semántica.
- Los nombres de elementos deben ser únicos dentro de cada namespace.
- Los namespace se pueden declarar en cualquier elemento del XML creando un atributo con el prefijo xmlns: seguido del nombre del namespace que vamos a usar.

```
<elemento xmlns:nombreNamespace="identificador"></elemento>
```

Namespaces

- Se considera buena práctica declarar los namespaces en el elemento raíz.
- El valor del atributo que declara el namespace debe ser único en un documento y suele ser una URI (Uniform Resource Identifier). Opcionalmente, alojado en esta URI, puede haber información sobre ese espacio de nombres.
- Es posible definir varios namespaces en un mismo documento XML.
- Cuando se va a declarar un elemento perteneciente un namespace se debe poner el prefijo nombreNamespace: en la etiqueta
`<mobiliario:banco>De madera blanco</mobiliario:banco>`

Ejemplo de uso de namespaces

```
<?xml version="1.0" encoding="utf-8" ?>
<Calle xmlns:mobiliario="http://www.w3.org/mobiliario/"
      xmlns:establecimiento="http://www.w3.org/establecimiento/" >
  <Nombre>Alcalá</Nombre>
  <Localidad>Madrid</Localidad>
  <PuntosDeInteres>
    <Punto>
      <Km>1.15</Km>
      <establecimiento:Banco>Santander</establecimiento:Banco>
    </Punto>
    <Punto>
      <Km>1.2</Km>
      <mobiliario:Banco>De madera blanco</mobiliario:Banco>
    </Punto>
  </PuntosDeInteres>
</Calle>
```

Namespaces por defecto

- Es posible definir un namespace por defecto que afectará a todas las etiquetas que no definan explícitamente a otro namespace.
- La forma de definir el namespace por defecto es con el atributo xmlns="URI"

```
<biblioteca xmlns="http://w3.org/biblioteca"
            xmlns:autor="http://w3.org/autor">
  <libro titulo="Cien años de soledad" >
    <autor:autor nombre="Gabriel García Márquez" />
  </libro>
</biblioteca>
```

- En este ejemplo tanto 'biblioteca' como 'libro' pertenecen al namespace por defecto porque no tienen ningún namespace asignado explícitamente.

Ejercicio