



XML





Repaso de XML

- ☐ Qué es
- ☐ Para qué sirve
- ☐ XML vs HTML
- ☐ Qué no hace
- ☐ Ventajas
- ☐ Árbol
- ☐ Sintaxis, elementos y atributos
- ☐ Analizadores: DOM y SAX
- ☐ Serializador
- ☐ SOAP

Índice



Qué es XML

- ◆ XML (eXtensible Markup Lenguaje)
- ◆ Estándar del W3C
- ◆ Metalenguaje de definición de etiquetas estructuradas
- ◆ Define reglas semánticas → Organizar laXML (eXtensible Markup Lenguaje)
- ◆ Estándar del W3C
- ◆ Metalenguaje de definición de etiquetas estructuradas
- ◆ Define reglas semánticas → Organizar la información
- ◆ XML no hace nada en particular (Sólo sirve para presentar datos/información)

Dónde se encuentra el XML



Para qué sirve

- Almacenar y transportar datos
- Interpretable por: máquina y humano
- Distribuir datos a través de Internet
- Es texto plano

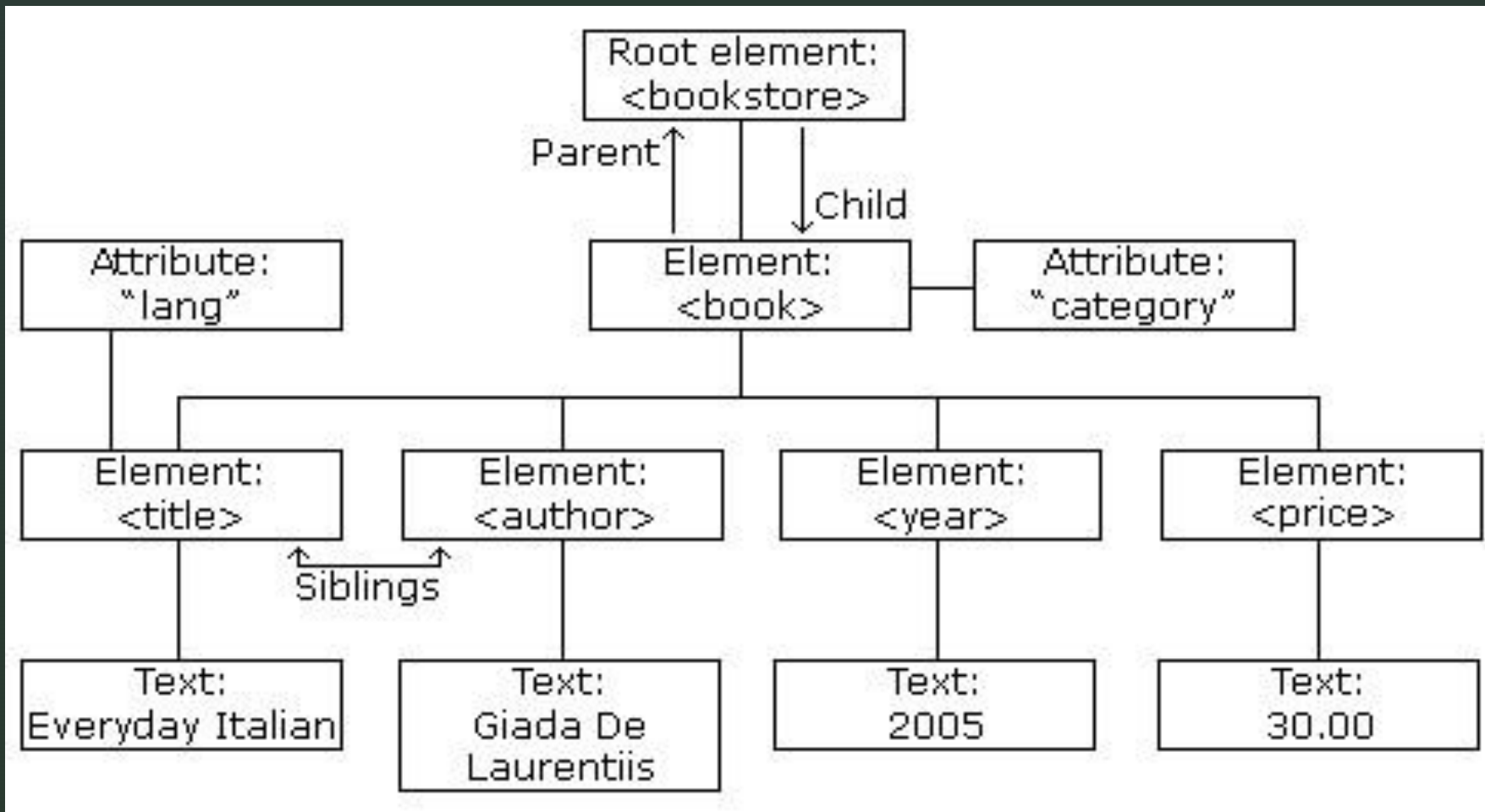
XML vs HTML

- Objetivos diferentes
- XML: Transportar datos
- HTML: mostrar datos, dar formato
- Usan lenguaje de etiquetas diferentes.
- XML no utiliza etiquetas predefinidas. Podemos añadir etiquetas inventadas
- HTML usa etiquetas definidas y conocidas <p> <h1> <table>...

Ventajas XML

- Bien formado
- Extensible
- Facilidad de conversión
- Fácil de leer
- Auto descriptivo
- Intercambiable
- Lectura/Interpretación -> Parser

Árbol XML (raíz y hojas)



Sintaxis XML

- Texto
- Atributos - Siempre entre comillas.
- Contiene eltos. XML mezcla de los anteriores.

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="infantiles">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>25.00</price>
    <quantity>8</quantity>
  </book>
  <book category="sistemas de televisión">
    <title>Sistemas Audiovisuales</title>
    <author>Francesc Tarrés Ruíz</author>
    <year>2000</year>
    <price>20.00</price>
    <quantity>3</quantity>
  </book>
</bookstore>
```

- DTD (document type definition)
 - Escrito en XML
 - No soporta DataTypes
 - No soporta namespaces
 - No define número y orden
 - No se puede manipular
 - Se puede malinterpretar
 - No extensible

- XSD (XML Schema Definition)

- Escrito en derivación de SGML
- DataTypes para elementos y atributos
- Soporta namespaces
- Define número y orden hijos eltos.
- Pueden ser manipulados por DOM
- Más seguro
- Extensible

1

Similitudes XSD vs DTD

Ambos tienen:

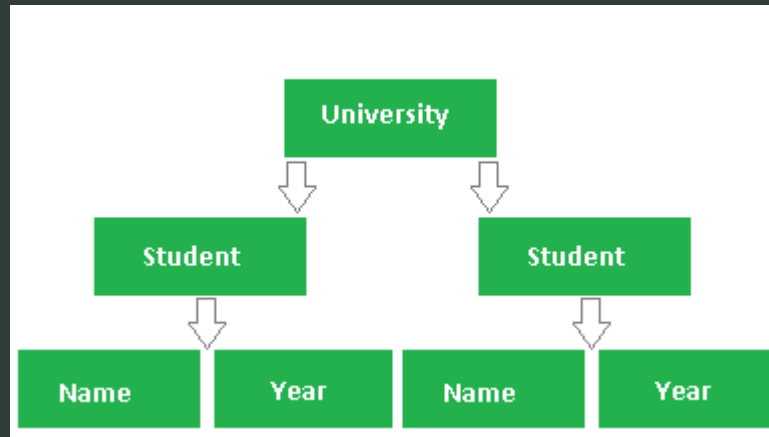
elements, attributes, anidación, orden, #occurences

Diferencias XSD vs DTD

XSD tiene data types, (typed) pointers, namespaces, keys y más.... DTD no

XML Validation

- Ej. Universidad con múltiples estudiantes. Tienen dos elementos: nombre y año.



DTD (document type definition)

- `<?xml version="1.0" encoding="UTF-8"?>`
- `<!DOCTYPE university[// --> university as root element`
- `<!ELEMENT university (student*)> // --> university has * = Multiple students`
- `<!ELEMENT student (name,year)> // --> Student has elements name and year`
- `<!ELEMENT name (#PCDATA)> // --> name as Parsed character data`
- `<!ELEMENT year (#PCDATA)> // --> year as Parsed character data`
- `]>`
- `<university>`
- `<student>`
- `<name>`
- `John Niel //---> I can also use an Integer,not good`
- `</name>`
- `<year>`
- `2000 //---> I can also use a string,not good`
- `</year>`
- `</student>`
- `</university>`

XSD (XML Schema Definition)

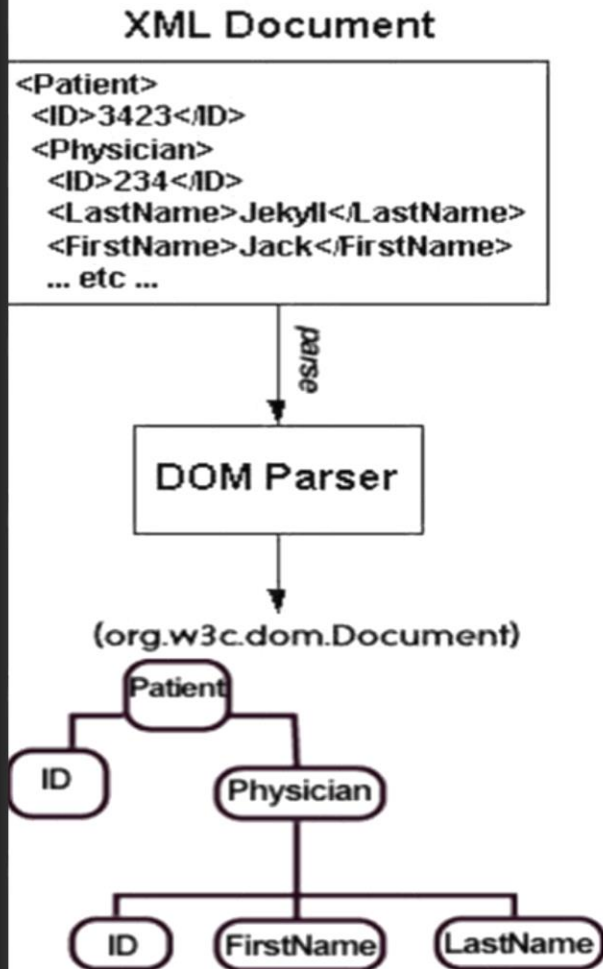
XML Validation

- `<?xml version="1.0" encoding="UTF-8"?>`
- `<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">`
- `<xsd:complexType name="uniType"> //--> complex datatype uniType`
- `<xsd:sequence>`
- `<xsd:element ref="student" maxOccurs="unbounded"/> //--> has unbounded no.of students`
- `</xsd:sequence>`
- `</xsd:complexType>`
- `<xsd:complexType name="stuType"> //--> complex datatype stuType`
- `<xsd:sequence>`
- `<xsd:element ref="name"/> //--> has element name`
- `<xsd:element ref="year"/> //--> has element year`
- `</xsd:sequence>`
- `</xsd:complexType>`
- `<xsd:element name="university" type="uniType"/> //--> university of type UniType`
- `<xsd:element name="student" type="stuType"/> //--> student of type stuType`
- `<xsd:element name="name" type="xsd:string"/> //--> name of datatype string`
- `<xsd:element name="year" type="xsd:integer"/> //--> year of datatype integer`
- `</xsd:schema>`

XML Parser (Analizadores)

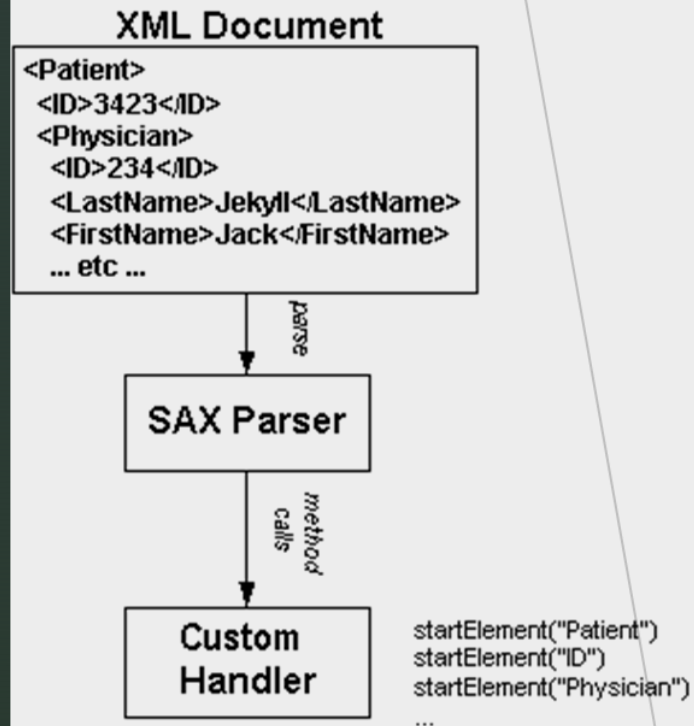
- Biblioteca o software - aporta interfaz para apps trabajen con documentos XML.
- Comprueba el formato adecuado XML
- Valida los documentos XML
- La mayoría de navegadores tienen un XML parser integrado.
- Los más conocidos son:
- DOM: en forma de árbol
- SAX: forma secuencial

XML Parser: DOM



- Acceso a los XML a través del DOM
- Modelo de objetos estándar
- Interfaz de programación estándar
- Independiente de la plataforma
- Estándar del W3C

XML Parser: SAX



- Acceso elementos del XML a través del SAX
- Lee el documento de elemento en elemento
- Consume menos memoria

Conversión

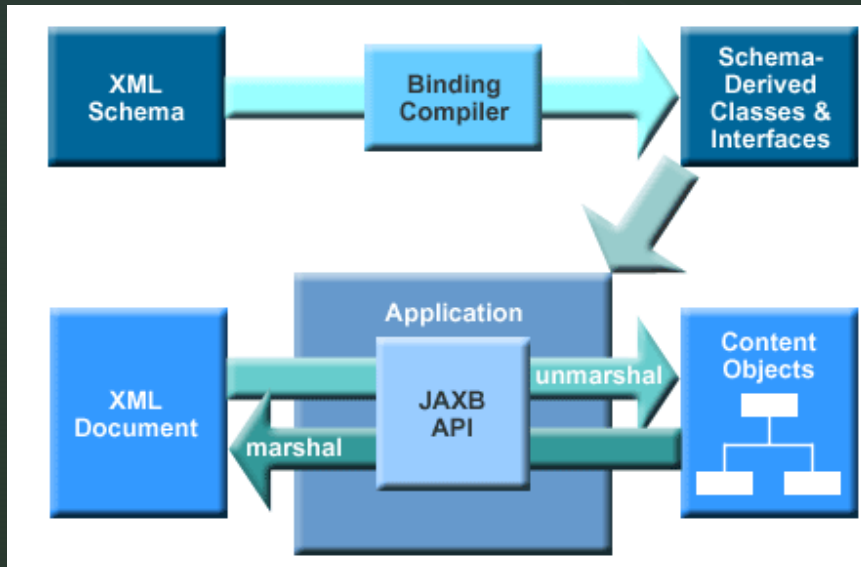
- XML
- Plantilla de conversión (.xsl eXtensible Stylesheet Language)
- Source (XML y .xsl)
 - TransformerFactory -> de la plantilla
- Result (.html, csv.....)

XML serializer

- Convierte objetos Java en documentos XML.
- Existen Frameworks específicos que pueden cargarse desde Apache Maven.
- Intercambio de objetos entre aplicaciones o operaciones CRUD en bases de datos.
- El propio DOM
- `DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();`
- JAXB
- Xstream

Serialización

- **JAXB** (Java Architecture for XML Binding)
- Estándar Java transformación a objetos
- API JAXB
 - mapear objeto Java a XML ("marshal")
 - De XML crear cjto Objs Java ("unmarshal")
- **Xstream** (simple library to serialize objects to XML)
- No permite generar clases java complejas
- Más sencillo
- No necesita etiquetas adicionales



Serialización

- JAXB
 - Eclipse 3.6 (Helios) y Java 1.6. (lib inc.)
 - Anotaciones
 - `@XmlRootElement(name = " ")`
 - `@XmlType(propOrder = { " ", " ", " ", " " })`
 - `@XmlElement(name = " ")`
 - `@XmlElementWrapper(= " ")`
- Xstream
 - puede serializar directamente hacia/desde cualquier estructura árbol (tree structure) no sólo XML
 - `.toXML(obj);`
 - `.fromXML();`

JSON (JavaScript Object Notation)

- Características
 - formato Datos en texto - sintaxis objeto de JavaScript
 - para transmitir datos en aplicaciones web
- APIs:
 - JAXB
 - XStream
 - GSON (google)
- Ej.fich config VisualStudioCode
- BDMongoDB