

- **XLM Namespaces**

- Es una recomendación de **W3C**
- Uso similar a la de otros lenguajes de programación: C++, Java, ...
- XML permite usar ***distintos vocabularios*** de etiquetado en el mismo documento:
 - Se persigue la ***modularidad*** y la ***reutilización*** de dichos vocabularios
 - **Problema: colisión de nombres**
 - Cuando manejamos varios ficheros XML, puede ocurrir que se repitan los nombres de las etiquetas, aunque con distinto significado.
 - Etiquetas con el mismo nombre
 - Una misma etiqueta puede tener distinto significado según el contexto
 - **Ej:** ¿significa lo mismo <reserva> en un **DTD** de una **biblioteca** que en un **DTD** de una **librería**? ¿Qué ocurre si juntamos ambos DTD's?
 - **Solución:** para diferenciar los elementos que se llaman igual, hay que emplear los ***espacios de nombres de XLM*** –proporcionan un nombre único universal-

- **XLM Namespaces**

- Proporcionan un nombre único universal

- **Objetivos:**

- **Evitar colisiones de nombres** –elementos y atributos- cuando se usan en un mismo documento XML diferentes vocabularios de etiquetado (XML debe permitir reutilización de vocabularios y modularidad)
 - Tener **varios espacios de nombre** en el **mismo documento**
 - Poder utilizar los **mismos nombres de etiquetas** en diferentes dominios del problema
 - *Desambiguar* el significado de elementos con el mismo nombre pero que residen en diferentes espacios de nombre
 - **Ej:** un *ratón* en informática no es lo mismo que en veterinaria
 - ***En un mismo espacio de nombres, un nombre sólo puede tener un significado***
 - El significado puede cambiar si cambio de espacio de nombres

- **XLM Namespaces**

- **Ej:** crea un documento XML llamado *prestamos.xml* para realizar préstamos de libros en una biblioteca

- **Planteamiento:**

- Elemento persona del vocabulario **persona.dtd**
 - Declarar los **elementos**: *persona, nombre, apellidos, email*
 - Elemento libro del vocabulario **libro.dtd**
 - Declarar los **elementos**: *libro, portada, título, edición, autor, nombre, apellidos, email, cuerpo, contraportada*
 - En ***prestamos.xml*** se incorporan **ambos** DTD's
 - Cuando en el documento aparece el elemento “nombre”, ¿a cuál se refiere?
 - Los elementos repetidos de ambos vocabularios provocarán un error tipo “*el elemento nombre ya está declarado*”

- XLM Namespaces

- **persona.dtd:**

```
<!ELEMENT persona (nombre, apellidos, email?)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

- **libro.dtd:**

```
<!ELEMENT libro (portada, cuerpo, contraportada)>
<!ELEMENT portada (titulo, edicion, autor, editor)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT edicion (#PCDATA)>
<!ELEMENT autor (nombre, apellidos, email?)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
<!ELEMENT contraportada (#PCDATA)>
```

■ XLM Namespaces

■ Sol:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE prestamos[
  <!ELEMENT prestamos (persona+|libro)+>
  <!ELEMENT persona (nombre, apellidos, email?)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apellidos (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
  <!ELEMENT libro (portada, cuerpo, contraportada)>
  <!ELEMENT portada (titulo, edicion, autor, editor)>
  <!ELEMENT titulo (#PCDATA)>
  <!ELEMENT edicion (#PCDATA)>
  <!ELEMENT autor (nombre, apellidos, email?)>
  <!--ELEMENT nombre (#PCDATA)-->
  <!--ELEMENT apellidos (#PCDATA)-->
  <!--ELEMENT email (#PCDATA)-->
  <!--ELEMENT editor (#PCDATA)-->
  <!--ELEMENT cuerpo (#PCDATA)-->
  <!--ELEMENT contraportada (#PCDATA)-->
]>
```

```
<prestamos>
  <persona>
    <nombre>nombre:persona</nombre>
    <apellidos>apellidos:persona</apellidos>
    <email>email:persona</email>
  </persona>
  <libro>
    <portada>
      <titulo>titulo</titulo>
      <edicion>edicion</edicion>
      <autor>
        <nombre>nombre:autor</nombre>
        <apellidos>apellidos:autor</apellidos>
        <email>email:autor</email>
      </autor>
      <editor>editor</editor>
    </portada>
    <cuerpo>cuerpo</cuerpo>
    <contraportada>contraportada</contraportada>
  </libro>
</prestamos>
```

Error at line 13, column 17: element 'nombre' has already been declared

- Se entiende mejor la **necesidad de crear espacios de nombres** cuando los elementos tienen el mismo identificador pero el significado es totalmente diferente.
 - Ej: ¿a qué nos referimos con el elemento **capital**? ¿y si lo situas en otro contexto?

- XLM Namespaces

- Declaración

- Se utiliza la palabra reservada **xmlns** (*XML NameSpace*)
- Si se declara en el **elemento raíz**, **todos** los elementos del documento pasan a formar parte del espacio de nombres
- Se suele asociar con una cadena de texto que debe ser única
- Se suele emplear una **URL**, que accede a un documento con información sobre el **NS**
- Ej: `<?xml version="1.0" encoding="utf-8"?>`

```
<productos xmlns="http://www.servidor.com/ns/productos/">
  <producto>
    <cod>LACT02330993</cod>
    <descrip>Leche entera 1L</descrip>
  </ producto >
  < producto >
    ...
  </ producto >
  ...
</productos>
```


- XLM Namespaces
- Declaración
 - Usar un **prefijo** cuando se utiliza **más** de un **espacio de nombres** en el mismo documento XML
 - Ej: **xmlns:cl="..."** y **xmlns:pr="..."**

```
<?xml version="1.0" encoding="utf-8"?>
<venda xmlns="http://www.atendadepaco.com/espazosdenomes/vendas/"
xmlns:cl="http://www.atendadepaco.com/espazosdenomes/clientes/"
xmlns:pr="http://www.atendadepaco.com/espazosdenomes/produtos/">
  <cl:cliente>
    <cl:cod>CL09384</cl:cod>
    <cl:nome>Uxío Fuentes Neira</cl:nome>
    <cl:enderezo>Rúa Europa 24, 3oA</cl:enderezo>
    <cl:teléfono>555098433</cl:teléfono>
  </cl:cliente>
  <pr:produtos>
    <pr:produto>
      <pr:cod>LACT02330993</pr:cod>
      <pr:descripción>Leite enteira envase 1L</pr:descripción>
    </pr:produto>
    <pr:produto>
      <pr:cod>LACT00493112</pr:cod>
      <pr:descripción>Margarina vexetal tarrina 250g</pr:descripción>
    </pr:produto>
  </pr:produtos>
  <importe_total>16,34€</importe_total>
</venda>
```

- **XLM Namespaces**
- Declarar *espacios de nombres*
 - Asociar un prefijo con referencia global y única a cada elemento
 - **nombre global = prefijo + nombre_local**
 - **prefijo**: uno para cada contexto
 - No necesita ser siempre el mismo
 - Puede cambiar de un documento XML a otro
 - Necesario asegurarse de que sea **global** y **único** => **evitar colisiones**
 - Hay que **declararlos antes** de usarlos
 - En dicha declaración se asocia una **URI** con cada prefijo
 - **URI (Uniform Resource Identifier)**
 - ✓ Apariencia URL, aunque no tiene que existir realmente
 - ✓ Se emplea porque los *nombres de dominio* son *únicos*
 - **nombre_local**: único en cada espacio de nombres

- XLM Namespaces

- Diferencias entre **URI**, **URN** y **URL**

- **URI: *Uniform Resource Identifier***

- Identificación inequívoca de un recurso por su **nombre** o por su **ubicación**, o por ambas

- **URN: *Uniform Resource Name***

- Define la identidad de algo
 - No lo localiza
 - Ej: [urn:isbn:9788467028317](#)

- **URL: *Uniform Resource Locator***

- Proporciona un método para encontrar *algo* en Internet

- Redefinición XLM Namespaces:

Colección de nombres –nombres de elementos y atributos- identificados por una URI



- XLM Namespaces
- Sintaxis
 - Nombre cualificado: **prefijoEspacioNombres:parteLocal**
 - **prefijoEspacioNombres:**
 - Cualquier cadena de caracteres válida como nombres de elemento o atributo –aquellos que no comienzan con “xml”-
 - Asociado al *Espacio de Nombres*
 - Los nombres cuyos prefijos son asociados con el mismo URI están en el mismo Espacio de Nombres
 - Los nombres cuyos prefijos son asociados con diferentes URIs están en diferentes Espacios de Nombres
 - **parteLocal:** nombre del elemento o atributo
 - Ej: **xlink: type**, **xls: template**, ...

- **XLM Namespaces**
- **Enlazar prefijos a espacios de nombres URI**
 - Adjuntar el atributo **xmlns:prefijo** al elemento con prefijo o uno de sus antecesores
 - **Ej:** si tenemos el elemento *svg* y el prefijo que creamos es también *svg*, ir asociando el prefijo a todos los elementos
 - ```
<svg:svg xmlns:svg="http://www.w3.org/Graphics/SVG/SVG-19991203.dtd" width="12cm" height="10cm">
```

```
 <svg:ellipse rx="110" ry="130" />
```

```
 <svg:rect x="4cm" y="1cm" width="3cm" height="6cm" />
```

```
</svg:svg>
```

- **XLM Namespaces**
- **Generalidades sobre los prefijos y los espacios de nombres**
  - **Ámbito:** *conjunto de elementos en el que podemos utilizar un NS*
  - Los prefijos tienen **ámbito** dentro del elemento en el que son declarados y dentro de todos sus descendientes
  - No es necesario definirlos todos en el elemento raíz.
  - Asociar un prefijo a un **URI ANTES** de usarlo
  - Dos URIs son equivalentes si son idénticos carácter a carácter
    - Ej: <http://www.example.org/~wilbur> - <http://www.Example.org/~wilbur>
    - **No son URIs iguales** (aunque sí serían **URLs iguales**)
  - Se puede redefinir el espacio de nombres por defecto volviéndolo a declarar
  - Si se usa el atributo **xmlns=""** en el interior de una etiqueta, indica que ese elemento y sus hijos no usan ningún espacio de nombres
- **Espacio de nombres por defecto**
  - Un elemento sin prefijo y todos sus elementos descendientes sin prefijo, pertenecen a un espacio de nombres en concreto, asociándole un atributo **xmlns** sin prefijo
  - Si a un elemento se le añade el atributo **xmlns**, queda cambiado su espacio de nombres –y el de los elementos hijo–

- XLM Namespaces
- Ejemplo de uso: espacio de nombres por defecto a *facturas*, prefijo para el espacio de nombres de los *productos* (pr) y otro prefijo para el de *clientes* (cl)

```
<?xml version="1.1" encoding="utf-8"?>
<facturas xmlns="http://www.at.com/namespace/facturas/"
xmlns:cl="http://www.at.com/namespace/clientes/"
xmlns:pr="http://www.at.com/namespace/produtos/">

 <cl:cliente>
 <cl:cod>CL09384</cl:cod>
 <cl:nome>Uxío Fuentes Neira</cl:nome>
 <cl:endereco>Rúa Europa 24, 3ºA</cl:endereco>
 <cl:teléfono>555098433</cl:teléfono>
 </cl:cliente>

 <pr:produtos>
 <pr:produto>
 <pr:cod> REF453 </pr:cod>
 <pr:descricao>Balón balonmano</pr:descricao>
 </pr:produto>
 <pr:produto>
 <pr:cod> REF454 </pr:cod>
 <pr:descricao>Balón baloncesto</pr:descricao>
 </pr:produto>
 ...
 </pr:produtos>
 ...
 <importe_total>46,34€</importe_total>
</facturas>
```

- XLM Namespaces
- Ejemplo de uso
  - Página HTML que incluye un lenguaje para especificar libros
    - Se especifica un prefijo para cada espacio de nombres (**h** y **xdc**)

```
<h:html xmlns:xdc="http://www.xml.com/books"
 xmlns:h="http://www.w3.org/HTML/1998/html4">
 <h:head><h:title>Revisiones de libros</h:title></h:head>
 <h:body>
 <xdc:bookreview>
 <xdc:title>XML en 10 minutos</xdc:title>
 <h:table>
 <h:tr align="center">
 <h:td>Autor</h:td><h:td>Precio</h:td>
 <h:td>Páginas</h:td><h:td>Fecha</h:td></h:tr>
 <h:tr align="left">
 <h:td><xdc:author>Sergio Luján Mora</xdc:author></h:td>
 <h:td><xdc:price>31.98</xdc:price></h:td>
 <h:td><xdc:pages>352</xdc:pages></h:td>
 <h:td><xdc:date>2003</xdc:date></h:td>
 </h:tr>
 </h:table>
 </xdc:bookreview>
 </h:body>
</h:html>
```

- XML Namespaces
- Ejemplo de uso
  - Para evitar indicar un espacio de nombres a los elementos HTML, se utiliza el *espacio de nombres predeterminado*, ahorrando código

```
<html xmlns="http://www.w3.org/HTML/1998/html4"
 xmlns:xdc="http://www.xml.com/books">
 <head><title>Revisiones de libros</title></head>
 <body>
 <xdc:bookreview>
 <xdc:title>XML en 10 minutos</xdc:title>
 <table>
 <tr align="center">
 <td>Autor</td><td>Precio</td>
 <td>Páginas</td><td>Fecha</td></tr>
 <tr align="left">
 <td><xdc:author>Sergio Luján Mora</xdc:author></td>
 <td><xdc:price>31.98</xdc:price></td>
 <td><xdc:pages>352</xdc:pages></td>
 <td><xdc:date>2003</xdc:date></td>
 </tr>
 </table>
 </xdc:bookreview>
 </body>
</html>
```



- **XLM Namespaces**

- **Recomendaciones**

- ✓ Definir el espacio de nombres en el elemento raíz del documento para que afecte a todos los elementos del documento
  - Es lo más habitual
  - Se gana en claridad y sencillez
- ✓ También se puede definir en un elemento determinado, afectándole sólo a él y a sus elementos hijos
- ✓ Si definimos el espacio de nombres por defecto, el documento ganará en claridad y se escribe menos código
- ✓ También se puede cualificar los atributos
  - Casi no tiene utilidad
  - No se suele utilizar para facilitar la lectura de los documentos