

Namespaces

The need for Namespaces

- Applications may use the same name to refer to different things.
 - Ex.: `<book>ticket</book>`
&
`<book>title</book>`
- **Validation** can be difficult as elements with similar names can have different content.
- **Display in browsers** is not flexible.

What is a Namespace?

- It **groups all related** elements and attributes from a single source.
- It **distinguishes** between elements with the same name by assigning them to **URIs** (different namespaces).
- Elements with the same name but **different URIs are different elements**.
- **URLs** are widely used as URIs.

Declaring Namespaces

- By A Fixed Attribute : xmlns

```
<x xmlns:edi=http://ecom.org/schema >  
  <edi:price units='Euro'>32.18</edi:price>  
</x>
```

Associating the namespace **prefix** “edi” with the namespace **name** “http://ecom.org/schema”

'price' element's namespace is http://ecommerce.org/schema

- Prefixes beginning with the three-letter sequence x, m, l, in any case combination, are **reserved** for use by XML and XML-related specifications.

XML Namespace

- An **XML namespace** is a collection of names, identified by a URI reference which are used in XML documents.
- Names from XML namespaces **may appear** as **qualified names**, which contain a single colon, separating the name into a **namespace prefix** and a **local part**.
- The prefix, which is mapped to a URI reference, selects a namespace .

What does a namespace look like?

<edi:price units='Euro'>32.18</edi:price>

price → *local part*

edi → *prefix*

edi:price → *qualified name*.

- To refer to certain namespace its not required to be connected to the Internet.
- Each parser see certain applications
(certain namespaces)

Namespace Scoping

- The namespace declaration is considered to apply to **the element where it is specified** and to **all elements within the content of that element**, unless overridden by another namespace declaration.

```
<bbb xmlns:edi = "http://ecom.org/schema" >  
  <ccc />  
</bbb>
```

```
<BBB xmlns:library="http://library.com"></BBB>  
  <CCC />
```

in the second case ccc is not from library ns

- **Multiple namespace** prefixes can be declared as attributes of a single element

```
<bk:book xmlns:bk='X' xmlns:isbn='Y' >  
  <bk:title>Cheaper by the Dozen</bk:title>  
  <isbn:number>1568491379</isbn:number>  
</bk:book>
```

Namespace Defaulting

- If we don't specify any namespace, then the default is **none**.
- Then unprefixed elements in the scope of the declaration are not considered to be in any namespace.
- Default namespace is defined by:
<root_element xmlns="URI">....</ root_element >
- A default namespace is applied to all elements with **no prefix** and **no ns overridden** within the content of that element.

Default Namespaces

- Used in cases where all the content of a particular element come from **a particular XML application**.
- In this case the prefix can be omitted:
 - `<book xmlns="http://www.books.com">`
 `<title>book_title</title>`
 - `</book>`