

XML Namespaces

(Week 7 Lecture 1)



Learning Objectives

- Understand the concept of Namespaces in XML documents.



Learning Objectives

- In the scheme of what we are doing in this unit:
 - We are studying how to use XML as an important set of Internet technologies to use as solutions in different areas.
 - Namespace is an integral part of XML. It allows us to use the same name for elements in different contexts. It allows us to clearly identify what names we are referring to.



Name Conflicts in XML

- Since any element names can be created in XML, very often a name conflict will occur when two different documents use the same names describing two different types of elements.
- In an environment where there are a lot of XML document being processes automatically by software, this can create many problems.



Example Name Conflict

HTML

```
<table>
  <tr>
    <td>Entry</td>
  </tr>
</table>
```

Can't put these two sets of tags together in a single document because the `<table>` element will be confused.

Furniture
Data

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```



Resolving Name Conflicts using Prefixes

```
<html:table>
  <html:tr>
    <html:td>Entry</html:td>
  </html:tr>
</html:table>
```

```
<furniture:table>
  <furniture:name>African Coffee Table
</furniture:name>
  <furniture:width>80</furniture:width>
  <furniture:length>120</furniture:length>
</furniture:table>
```



Namespace Declarations

```
<html:table xmlns:html="http://www.w3.org/TR/html4/">
  <html:tr>
    <html:td>Entry</html:td>
  </html:tr>
</html:table>
```

```
<furniture:table xmlns:furniture="http://www.table.org/">
  <furniture:name>African Coffee Table
</furniture:name>
  <furniture:width>80</furniture:width>
  <furniture:length>120</furniture:length>
</furniture:table>
```



Namespaces and the xmlns attribute

- Instead of using only prefixes, an **xmlns attribute** has been added to the <table> tag to give the element prefix a **qualified name (QName)** associated with a **namespace**.
- Although the namespace identifier is usually a URL, it doesn't really have to be a valid web address
 - It can be any arbitrary string to uniquely identify the particular namespace.



Namespace for XSLT

From last lecture:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <HTML>
      <BODY>
        <xsl:foreach select="/poetry/anthology/poem/stanza/line">
          <P>
            <xsl:value-of select="."/>
          </P>
        </xsl:foreach>
      </BODY>
    </HTML>
  </xsl:template>
</xsl:stylesheet>
```



Mixing Namespaces

```
<my_information
  xmlns:html="http://www.w3.org/TR/html4/"
  xmlns:furniture="http://www.table.org/" >

  <html:table>
    ...
  </html:table>

  <furniture:table>
    ...
  </furniture:table>

</my_information>
```



Default Namespaces

All tags without prefix

```
<my_information
  xmlns:"http://www.w3.org/TR/html4/" ← default
  xmlns:furniture="http://www.table.org/" >

  <table>...</table> ←
  <furniture:table>...</furniture:table>

</my_information>
```



Namespaces in DTDs and Schemas

- Namespaces are not natural in DTDs.
- If namespaces are used in a DTD, EVERY element type from each namespace must be included in the DTD - which defeats the purpose of using multiple namespaces from various independent sources.

Namespaces in DTDs and Schemas

- XML Schemas support Namespaces naturally.
Eg.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:p="http://www.poets.com.au">
...

```



Further Reading

- Sybex textbook page 34-37.
- Optional reading:
 - <http://www.w3.org/TR/REC-xml-names/>