



## XML Tutorial

[XML HOME](#)[XML Introduction](#)[XML How to use](#)[XML Tree](#)[XML Syntax](#)[XML Elements](#)[XML Attributes](#)[XML Namespaces](#)[XML Display](#)[XML XSLT](#)[XML XPath](#)[XML XLink](#)[XML Doctypes](#)[XML Validator](#)[XML DTD](#)[XML Schema](#)[XML Server](#)[XML Applications](#)[XML Examples](#)[XML Quiz](#)

# XML Namespaces

[« Previous](#)[Next Chapter »](#)

XML Namespaces provide a method to avoid element name conflicts.

## Name Conflicts

In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.

This XML carries HTML table information:

```
<table>
  <tr>
    <td>Apples</td>
```

XML Certificate

## XML DOM

DOM Intro

DOM Nodes

DOM XMLHttpRequest

DOM Accessing

DOM Node Info

DOM Node List

DOM Traversing

DOM Navigating

DOM Get Values

DOM Change Nodes

DOM Remove Nodes

DOM Replace Nodes

DOM Create Nodes

DOM Add Nodes

DOM Clone Nodes

DOM Examples

## DOM Reference

DOM Node Types

DOM Node

DOM NodeList

DOM NamedNodeMap

DOM Document

```
<td>Bananas</td>
</tr>
</table>
```

This XML carries information about a table (a piece of furniture):

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

If these XML fragments were added together, there would be a name conflict. Both contain a `<table>` element, but the elements have different content and meaning.

A user or an XML application will not know how to handle these differences.

## Solving the Name Conflict Using a Prefix

Name conflicts in XML can easily be avoided using a name prefix.

This XML carries information about an HTML table, and a piece of furniture:

```
<h:table>
  <h:tr>
```

- DOM Element
- DOM Attribute
- DOM Text
- DOM CDATA
- DOM Comment
- DOM XMLHttpRequest
- DOM Parser

## Web Services

- XML Services
- XML WSDL
- XML SOAP
- XML RDF
- XML RSS

## XML DTD

- DTD Intro
- DTD Building Blocks
- DTD Elements
- DTD Attributes
- DTD Elements vs Attr
- DTD Entities
- DTD Examples

## XSD Schema

```
<h:td>Apples</h:td>
<h:td>Bananas</h:td>
</h:tr>
</h:table>

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

In the example above, there will be no conflict because the two `<table>` elements have different names.

## XML Namespaces - The xmlns Attribute

When using prefixes in XML, a **namespace** for the prefix must be defined.

The namespace can be defined by an **xmlns** attribute in the start tag of an element.

The namespace declaration has the following syntax. `xmlns:prefix="URI"`.

```
<root>

<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
```

- XSD Intro
- XSD Why Use
- XSD How To
- XSD <schema>

## XSD Simple

- XSD Elements
- XSD Attributes
- XSD Restrictions

## XSD Complex

- XSD Elements
- XSD Empty
- XSD Elements Only
- XSD Text Only
- XSD Mixed
- XSD Indicators
- XSD <any>
- XSD <anyAttribute>
- XSD Substitution
- XSD Example

## XSD Data

- XSD String
- XSD Date

```
</h:tr>
</h:table>

<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

In the example above:

The xmlns attribute in the first <table> element gives the h: prefix a qualified namespace.

The xmlns attribute in the second <table> element gives the f: prefix a qualified namespace.

When a namespace is defined for an element, all child elements with the same prefix are associated with the same namespace.

Namespaces can also be declared the XML root element:

```
<root
xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="http://www.w3schools.com/furniture">

  <h:table>
    <h:tr>
      <h:td>Apples</h:td>
```

XSD Numeric

XSD Misc

## XSD References

XSD Reference

```
<h:td>Bananas</h:td>
</h:tr>
</h:table>

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

</root>
```

**Note:** The namespace URI is not used by the parser to look up information.

The purpose of using an URI is to give the namespace a unique name.

However, companies often use the namespace as a pointer to a web page containing namespace information.

---

## Uniform Resource Identifier (URI)

A **Uniform Resource Identifier** (URI) is a string of characters which identifies an Internet Resource.

The most common URI is the **Uniform Resource Locator** (URL) which identifies an Internet domain address. Another, not so common type of URI is the **Universal Resource Name** (URN).

---

# Default Namespaces

Defining a default namespace for an element saves us from using prefixes in all the child elements. It has the following syntax:

```
xmlns="namespaceURI"
```

This XML carries HTML table information:

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

This XML carries information about a piece of furniture:

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

# Namespaces in Real Use

XSLT is a language that can be used to transform XML documents into other formats.

The XML document below, is a document used to transform XML into HTML.

The namespace "http://www.w3.org/1999/XSL/Transform" identifies XSLT elements inside an HTML document:

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

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr>
      <th style="text-align:left">Title</th>
      <th style="text-align:left">Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

```
</table>  
</body>  
</html>  
</xsl:template>  
  
</xsl:stylesheet>
```

If you want to learn more about XSLT, please read our [XSLT Tutorial](#).

« Previous

Next Chapter »





W3SCHOOLS EXAMS

## COLOR PICKER



## SHARE THIS PAGE



**Amazon Deal of the Day**



**960GB**  
46% off  
Today  
Only!

**SanDisk**  
Ultra II  
Solid State Drive

Everything You Do *Faster* with a SSD

4.6 ★★★★★

 **See Deal**

## Create Wireframes

Like You're Getting Paid. Interactive Wireframe Tool

## Online Consumer Survey

## File APIs for Programmers

## Big Data For Finance

[REPORT ERROR](#)  
[PRINT PAGE](#)  
[FORUM](#)  
[ABOUT](#)

---

## Top 10 Tutorials

[HTML Tutorial](#)  
[CSS Tutorial](#)  
[JavaScript Tutorial](#)  
[SQL Tutorial](#)  
[PHP Tutorial](#)  
[jQuery Tutorial](#)  
[Bootstrap Tutorial](#)  
[Angular Tutorial](#)  
[ASP.NET Tutorial](#)  
[XML Tutorial](#)

## Top 10 References

[HTML Reference](#)  
[CSS Reference](#)  
[JavaScript Reference](#)  
[Browser Statistics](#)  
[HTML DOM](#)  
[PHP Reference](#)  
[jQuery Reference](#)  
[HTML Colors](#)  
[HTML Character Sets](#)  
[XML Reference](#)

## Top 10 Examples

[HTML Examples](#)  
[CSS Examples](#)  
[JavaScript Examples](#)

[HTML DOM Examples](#)

[PHP Examples](#)

[jQuery Examples](#)

[XML Examples](#)

[ASP Examples](#)

[SVG Examples](#)

## Web Certificates

[HTML Certificate](#)

[HTML5 Certificate](#)

[CSS Certificate](#)

[JavaScript Certificate](#)

[jQuery Certificate](#)

[PHP Certificate](#)

[Bootstrap Certificate](#)

[XML Certificate](#)

---

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2015 by Refsnes Data. All Rights Reserved.

