

[HTML](#)[CSS](#)[JAVASCRIPT](#)[SQL](#)[TUTORIALS ▾](#)

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 Elements

[« Previous](#)[Next Chapter »](#)

An XML document contains XML Elements.

What is an XML Element?

An XML element is everything from (including) the element's start tag to (including) the element's end tag.

```
<price>29.99</price>
```

An element can contain:

XML Examples

XML Quiz

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

- text
- attributes
- other elements
- or a mix of the above

```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

In the example above:

<title>, <author>, <year>, and <price> have **text content** because they contain text (like 29.99).

<bookstore> and <book> have **element contents**, because they contain elements.

<book> has an **attribute** (category="children").

DOM NamedNodeMap
DOM Document
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

Empty XML Elements

An element with no content is said to be empty.

In XML, you can indicate an empty element like this:

```
<element></element>
```

You can also use a so called self-closing tag:

```
<element />
```

The two forms produce identical results in XML software (Readers, Parsers, Browsers).



Empty elements can have attributes.

XML Naming Rules

XML elements must follow these naming rules:

- Element names are case-sensitive
- Element names must start with a letter or underscore
- Element names cannot start with the letters xml (or XML, or Xml, etc)

XSD Schema

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

- Element names can contain letters, digits, hyphens, underscores, and periods
- Element names cannot contain spaces

Any name can be used, no words are reserved (except xml).

Best Naming Practices

Create descriptive names, like this: <person>, <firstname>, <lastname>.

Create short and simple names, like this: <book_title> not like this: <the_title_of_the_book>.

Avoid "-". If you name something "first-name", some software may think you want to subtract "name" from "first".

Avoid ".". If you name something "first.name", some software may think that "name" is a property of the object "first".

Avoid ":". Colons are reserved for namespaces (more later).

Non-English letters like éòá are perfectly legal in XML, but watch out for problems if your software doesn't support them.

Naming Styles

There are no naming styles defined for XML elements. But here are some commonly used:

Style	Example	Description
Lower case	<firstname>	All letters lower case

XSD Date

XSD Numeric

XSD Misc

XSD References

XSD Reference

Upper case	<FIRSTNAME>	All letters upper case
Underscore	<first_name>	Underscore separates words
Pascal case	<FirstName>	Uppercase first letter in each word
Camel case	<firstName>	Uppercase first letter in each word except the first

If you choose a naming style, it is good to be consistent!

XML documents often have a corresponding database. A common practice is to use the naming rules of the database for the XML elements.



Camel case is a common naming rule in JavaScripts.

XML Elements are Extensible

XML elements can be extended to carry more information.

Look at the following XML example:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```

Let's imagine that we created an application that extracted the <to>, <from>, and <body> elements from the XML document to produce this output:

MESSAGE

To: Tove

From: Jani

Don't forget me this weekend!

Imagine that the author of the XML document added some extra information to it:

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Should the application break or crash?

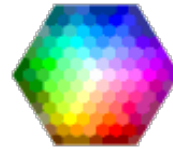
No. The application should still be able to find the <to>, <from>, and <body> elements in the XML document and produce the same output.

This is one of the beauties of XML. It can be extended without breaking applications.

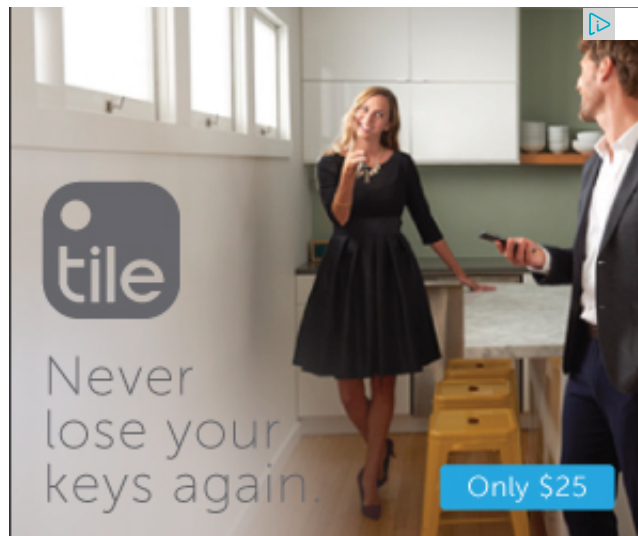


W3SCHOOLS EXAMS

COLOR PICKER



SHARE THIS PAGE



[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.

