KickStart Tutorial XML

version 1.0

free ebooks by spiderpro

General information

The tutorial is available at SpiderPro in the following versions:

- **Online HTML** http://www.spiderpro.com/bu/buxmlm001.html
- **PDF** http://www.spiderpro.com/ebooks/kickstartxml.pdf
- **Zipped PDF** http://www.spiderpro.com/ebooks/kickstartxml.zip

The *KickStart Tutorial XML* is free. If somebody asked you money for it, you've been swindled!

Happy XML-ing

Jan Kampherbeek
Webmaster of SpiderPro
http://www.spiderpro.com/
jan@spiderpro.com

This tutorial

In this tutorial you will learn what XML is about.

You'll understand the basic XML syntax. An you will know what's needed to make XML usable.

You won't be an XML expert after following this kickstart tutorial. But you'll understand the basics of XML. And you'll be able to understand XML Documents and most of XML DTD's.

Index

Why do we need XML?

Data-exchange

XML is used to aid the exchange of data. It makes it possible to define data in a clear way.
Both the sending and the receiving party will use XML to understand the kind of data that's been sent. By using XML everybody knows that the same interpretation of the data is used

Replacement for EDI

EDI (**E**lectronic **D**ata **I**nterchange) has been for several years the way to exchange data between businesses.
EDI is expensive, it uses a dedicated communication infrastructure.
And the definitions used are far from flexible.
XML is a good replacement for EDI. It uses the Internet for the data exchange. And it's very flexible.

More possibilities

XML makes communication easy. It's a great tool for transactions between businesses.
But it has much more possibilities.
You can define other languages with XML. A good example is WML (Wireless Markup Language), the language used in WAP-communications.
WML is just an XML dialect.

What is XML ?

Simpler SGML

XML is a meta-language.
A meta-language is a language that's used to define other languages. You can use XML for instance to define a language like WML.
XML is a smaller version of SGML. It's easy to master and that's a major advantage compared to SGML which is a very complex meta-language.

XML: What it can do

With XML you can :

- Define data structures
- Make these structures platform independent
- Process XML defined data automatically
- Define your own tags

With XML you cannot

- Define how your data is shown. To show data, you need other techniques.

The general structure of XML

Define your own tags

In XML, you define your own tags.
If you need a tag **<TUTORIAL>** or **<STOCKRATE>**, that's no problem.

DTD or Schema

If you want to use a tag, you'll have to define it's meaning.
This definition is stored in a **DTD** (**D**ocument **T**ype **D**efinition). You can define your own DTD or use an existing one.
Defining a DTD actually means defining a XML language.
An alternative for a DTD is Schema.

Showing the results

Often it's not necessary to display the data in a XML document. It's for instance possible to store the data in a database right away.
If you want to show the data, you can. XML itself is not capable of doing so.
But XML documents can be made visible with the aid of a language that defines the presentation.
XSL (e**X**tensible **S**tylesheet **L**anguage) is created for this purpose. But the presentation can also be defined with **CSS** (**C**ascading **S**tyle **S**heets).

XML Tags

Tags

XML tags are created like HTML tags. There's a start tag and a closing tag.
<TAG>content**</TAG>**
The closing tag uses a slash after the opening bracket, just like in HTML.
The text between the brackets is called an element.

Syntax

The following rules are used for using XML tags:

- Tags are case sensitive. The tag **<TRAVEL>** differs from the tags **<Travel>** and **<travel>**
- Starting tags always need a closing tag
- All tags must be nested properly
- Comments can be used like in HTML: **<!--** Comments **-->**
- Between the starting tag and the end tag XML expects the content. **<amount>**135**</amount>** is a valid tag for an element amount that has the content 135

Empty tags

Besides a starting tag and a closing tag, you can use an empty tag. An empty tag does not have a closing tag.
The syntax differs from HTML: **<TAG/>**

Elements and sub elements

Elements and children

With XML tags you define the type of data. But often data is more complex. It can consist of several parts.
To describe the element car you can define the tags <car>mercedes</car>. This model might look like this:

```
<car>
<brand>volvo</brand>
<type>v40</type>
<color>green</color>
</car>
```

Besides the element car three other elements are used: **brand**, **type** and **color**.
Brand, type and color are sub-elements of the element car. In the XML-code the tags of the sub-elements are enclosed within the tags of the element car. Sub-elements are also called children

XML documents

The XML declaration

The first line of an XML document is the XML declaration.
It's a special kind of tag:

```
<?xml version="1.0"?>
```

The version 1.0 is the actual version of XML.
The XML declaration makes clear that we're talking XML and also which version is used.
The version identification will become important after new versions of XML are used.

The root element

All XML documents must have a root element.
All other elements in the same document are children of this root element. The root element is the top level of the structure in an XML document.

Structure of an XML page

```
<?xml version="1.0"?>
<root>
        <element>
                <sub-element>
                        content
                </sub-element>
                <sub-element>
                        content
                </sub-element>
        element>
</root>
```

All elements must be nested. The level of nesting can be arbitrarily deep.

A real XML page

```
<?xml version="1.0"?>
<sales>
        <shop>
                <number>
                        100
                </number>
                <manager>
                        Ray Bradbury
                </manager>
        </shop>
        <product>
                <name>
                        carrots
                </name>
                <totalprice>
                        10
                </totalprice>
        </product>
</sales>
```

XML Attributes

Attributes

Elements in XML can use attributes. The syntax is:

```
<element attribute-name = "attribute-value">....</element>
```

The value of an attribute needs to be quoted, even if it contains only numbers.
An example

```
<car color = "green">volvo</car>
```

The same information can also be defined without using attributes:

```
<car>
	<brand>volvo</brand>
	<color>green</color>
</car>
```

Avoid attributes

When possible try to avoid attributes. Data structures are more easy described in XML-tags.
Software that checks XML-documents can do a better job with tags than with attributes.

Well formed XML documents

Well formedness

An XML document needs to be well formed. Well formed means that the document applies to the syntax rules for XML.

The rules

To be well formed a document needs to comply to the following rules:

- it contains a root element
- all other elements are children of the root element
- all elements are correctly paired
- the element name in a start-tag and an end-tag are exactly the same
- attribute names are used only once within the same element

Note

There are more rules, some of them have to do with entities. In this quick tutorial, entities are not covered.

Valid XML documents

Valid

To be of practical use, an XML document needs to be valid. To be valid an XML document needs to apply to the following rules:

- The document must be well formed. (More on well formed in the previous page).
- The document must apply to the rules as defined in a Document Type Definition (DTD), (More on DTD's in the next page)

If a document is valid, it's clearly defined what the data in the document really means.

There's no possibility to use a tag that's not defined in the DTD. Companies that exchange XML-documents can check them with the same DTD.
Because a valid XML document is also well formed, there's no possibility for typo's in the tags.

Valid is about structure

A valid XML-document has a structure that's valid. That's the part you can check. There's no check for the content.

XML: the DTD

Defining the language

To use XML you need a **DTD** (**D**ocument **T**ype **D**efinition).
A DTD contains the rules for a particular type of XML-documents.
Actually it's the DD that defines the language.

Elements

A DTD describes elements. It uses the following syntax:
The text **<! ELEMENT**, followed by the name of the element, followed by a description of the element.
For instance:
<!ELEMENT brand (#PCDATA)>
This DTD description defines the XML tag **<brand>**.

Data

The description **(#PCDATA)** stands for parsed character data.
It's the tag that is shown and also will be parsed (interpreted) by the program that reads the XML document.
You can also define **(#CDATA)**, this stands for character data.
CDATA will not be parsed or shown.

Sub elements

An element that contains sub elements is described thus:

```
<!ELEMENT car (brand, type) >
<!ELEMENT brand (#PCDATA) >
<!ELEMENT type (#PCDATA) >
```

-16-

This means that the element car has two subtypes: brand and type. Each subtype can contain characters.

Number of sub elements

If you use **<!ELEMENT car (brand, type) >**, the sub elements brand and type can occur once inside the element car. To change the number of possible occurrences the following indications can be used:

- + must occur at least one time but may occur more often
- * may occur more often but may also be omitted
- ? may occur once or not at all

The indications are used behind the sub element name. For instance:
<!ELEMENT animal (color+) >

Making choices

With the sign '|' you define a choice between two sub elements.
You enter the sign between the names of the sub elements.
<!ELEMENT animal (wingsize|legsize) >

Empty elements

Empty elements get the description EMPTY.
For instance
<!ELEMENT separator EMPTY>
that could define a separator line to be shown if the XML document appears in a browser.

DTD: external

A DTD can be an external document that's referred to.
Such a DTD starts with the text
<!DOCTYPE name of root-element SYSTEM "address">
The address is an URL that points to the DTD.

In the XML document you make clear that you'll use this DTD with the line:
<!DOCTYPE name of root-element SYSTEM "address">
that should be typed after the line <?xml version="1.0"?>

DTD: internal

A DTD can also be included in the XML document itself.
After the line **<?xml version="1.0"?>** you must type
<!DOCTYPE name of root-element [
followed by the element definitions.
The DTD part is closed with
]>

Presenting XML documents

Showing XML documents

XML is about defining data. With XML you can define documents that are understood by computers.
But to make these documents understandable to humans, you need to show them.

CSS

Cascading Style sheets (CSS offer possibilities to show XML.
It works just like adding styles to HTML elements.

XSL

The preferred solution is using XSL (eXtensible Style sheet Language).
XSL can convert XML documents into HTML.
It can be used client side but the best solution is to use XSL server side. You can convert your XML documents to HTML, thus making them visible to any browser.

About SpiderPro

This eBook is published by SpiderPro. SpiderPro is a website that presents information for professional webmasters.
Well, with professional I don't want to frighten anyone. If you just want to build a nice personal homepage you'll find lots of info at SpiderPro.
But you'll also find info that you can use only if you have access to a webserver. Or if you know how to write programs.
With SpiderPro I hope to reach people that are strongly involved with web development. Maybe for a living, maybe just for the hack of it...

The URL of SpiderPro is http://www.spiderpro.com/

Mailing List

SpiderPro's mailing list informs you when new information becomes available. Subscription is free.
If you subscribe, you'll receive mails about updates to SpiderPro and about new released ebooks.
Expect to receive this mails about 2 times a month.
SpiderPro will use your email address for the sole purpose of sending you mails from the list.
We will never use your address for any other purpose. We will never give your address to anyone and we will never sell your address.
You can subscribe at:
http://www.spiderpro.com/ab/abmlist.html

Disclaimer

The autor of this ebook is dutch, the use of english surely won't be perfect. If you find a more or less annoying mistake I'll be glad to receive a mail at:
corr@spiderpro.com

Jan Kampherbeek, May 5, 2001