



XML

(eXtensible Markup Language)

XML pavyzdys

<note>

<to>Tove</to>

<from>Jani</from>

<body>Don't forget me!</body>

</note>

XML pavyzdys

<note>

<to> </to>

<from> </from>

<body> </body>

</note>

What is XML?

- **EX**tensible **M**arkup **L**anguage
- is a **markup language** (like HTML)
- was designed to **describe data**
- XML tags are not predefined.
 - You must **define your own tags**
- XML uses
 - **XML Schema** to describe the data
 - **Document Type Definition (DTD)** was earlier
- XML with XML Schema (not DTD) is designed to be **self-descriptive**

XML and HTML

- **XML** was designed to **describe data** and to focus on **what data is**.
- **HTML** with **CSS** was designed to **display data** and to focus on **how data looks**.
- HTML has standard tags and structure
- XML allows the author to define his own
 - tags
 - document structure

XML does not DO anything

- XML was not designed to DO anything
- XML was created to structure, store and to send information

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

With XML:

- XML is a platform, software and hardware independent tool for transmitting information
 - Data can be exchanged between incompatible systems
- Plain text files can be used to share and store data
- XML is the mother of WAP, WML, RDF, RSS, OWL, SOAP, WSDL, ...

An example XML document

<?xml version="1.0" encoding="ISO-8859-1"?>

<note> *// root element*

<to>Tove</to> *// child element*

<from>Jani</from> *// child ...*

<heading>Reminder</heading> *// child ...*

<body>Don't forget me!</body>

</note>

XML Syntax Rules

- It is illegal to omit the element closing tag
- XML tags are **case sensitive**
- Improper nesting of tags makes no sense to XML
- XML Document must have:
 - **root element**
 - quotation marks around attribute values

Pavyzdys

<zinute>

<dalyviai>

<to> </**to**>

<to> </to>

<cc> </cc>

</**to**>???

</dalyviai>

</zinute>

<note>

<to>Tove

<from>Jani </to> ??? </from>

<body>Don't forget me!</body>

</note>

XML Syntax Rules (2)

- The white space in your document is not truncated
- `<!-- This is a comment -->`
- XML Elements are extensible and they have relationships

<zinute>

<dalyviai>

<to> </to>

<to> </to>

<cc> </cc>

</dalyviai>

</zinute>

Linus#Tove#Today is holiday

Linus #Tove #

Types of elements

```
<book>
  <title>Pirmas blynas</title>
  <prod id="33-657">
</prod>
  <chapter>Introduction to XML
    <para>What is HTML</para>
    <para>What is XML</para>
  </chapter>
</book>
```

- Content type:
 - root <book>
 - simple <para>
 - mixed <chapter>
 - empty <prod>
- Attributes:
 - name
 - value

Attributes

- provide additional information about elements
 - information that **is not** a part of the **data**
 - irrelevant to the data, but important to the software

```
<file type="gif" > computer.gif </file>
```

- metadata (data about data) should be stored as attributes, and data itself should be stored as elements

Nice way...

<note date="12/11/2002" > ... </note>



<note>

<date>12/11/2002</date>

</note>



<note>

<date>

<day>12</day>

<month>11</month>

<year>2002</year>

</date>

</note>



When avoid using attributes?

- Problems with using attributes:
 - Can not contain multiple values (child elements can)
 - are not easily expandable (for future changes)
 - cannot describe structures (child elements can)
 - are more difficult to manipulate by program code
 - attribute values are not easy to test against a Document Type Definition (DTD)

XML should not look like...

```
<note day="15" month="10" year="2007"  
to="CNN" from="Simpson"  
heading="Alarm!!!" body="Spider-Pig  
missing!">  
</note>
```

Browsers and XML

	v	XML	XSL	CSS
Mozilla Firefox	1.0.2	+	+	+
Internet Explorer	6	+	+	+
Netscape (uses Mozilla)	8	+	+	+
Opera	9	+	+	+

XML Namespaces – Why?

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

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

Using XML Namespaces

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

Namespace (xmlns) Attribute

- placed in the start tag of an element

xmlns: namespace-prefix="namespaceURI"

- All child elements with the same prefix are associated with the parent namespace.
- **Uniform Resource Identifier (URI)** is a string of characters which identifies an Internet Resource
 - URN would be more correct

Default Namespace

- ...saves us from using prefixes in all the child elements

`xmlns="namespaceURI"`

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


XML CDATA section

- All text in an XML document will be parsed by the parser
 - **except** CDATA section!

```
<message>This text is also parsed</message>
```

```
<name>  
  <first>Bill</first>  
  <last>Gates</last>  
</name>
```

CDATA

- Everything inside a CDATA section is ignored by the parser

```
<script>
<![CDATA[
function matchwo(a,b) {
if (a < b && a < 0) then {
    return 1
} else {
    return 0
} }
]]>
</script>
```

Escape characters

- Illegal XML characters have to be replaced by entity references

~~<message>if salary < 1000 then</message>~~

<message>if salary < 1000 then</message>

<	&lt;
>	&gt;
&	&amp;
'	&apos;
“	&quot;

XML encoding

- the encoding attribute has to specify the encoding used when the document was saved

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <from>Studentas</from>
  <to>Dėsytojas</to>
  <message>Jūsų paskaitos labai normalios!!!
</message>
</note>
```

■ Only for Internet Explorer

```
<xml src="cd_catalog.xml" id="xmldso" > </xml>
```

```
<table datasrc="#xmldso" width="100%" border="1">
```

```
<thead>
```

```
<th>Title</th>
```

```
<th>Artist</th>
```

```
<th>Year</th>
```

```
</thead>
```

```
<tr align="left">
```

```
<td><span datafld="TITLE"></span></td>
```

```
<td><span datafld="ARTIST"></span></td>
```

```
<td><span datafld="YEAR"></span></td>
```

```
</tr>
```

```
</table>
```

Well formed XML

- must begin with the XML declaration
- must have one unique root element
- all start tags must match end-tags
- XML tags are case sensitive
- all elements must be closed
- all elements must be properly nested
- all attribute values must be quoted
- XML entities must be used for special characters

CR and LF

- **With XML, CR / LF is always converted to LF**
- Once upon a time where was a typewriter...
- After you have typed one line of text on a typewriter, you have to manually return the printing carriage (that's CR) to the left margin position and manually feed the paper up one line (that's LF).
- In Windows applications, a new line is normally stored as a pair of characters: carriage return (CR) and line feed (LF).
- In Unix applications, a new line is normally stored as a LF character.
- Macintosh applications use only a CR character to store a new line.

Typewriter



XML naming rules

- Names can contain letters, numbers, and other characters
- Names must not start with a number or punctuation character
- Names must not start with the letters xml (or XML, or Xml, etc)
- Names cannot contain spaces

XML naming examples

<note > ...

<my note> ... (wrong example)

<my_note> ...

<my note="231"> ...

XML naming rules (2)

- Names should be descriptive and short:
 - `<book_title>`
 - not like this: `<the_title_of_the_book>`
- Use “_” sign
- Not recommended to use: “.” “-”
- Should not be used: “:”
- It helps a lot transferring data to database, so use the same rules for naming the elements

Validating XML

- (if not talked about)
- Differences between browser (IE, FF, ...) validator and real validator

XML displaying with CSS

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    ...
  </CD>
</CATALOG>
```

```
CATALOG {
  background-color: #ffffff;
  width: 100%;
}
CD {
  display: block;
  margin-bottom: 30pt;
  margin-left: 0;
}
```