

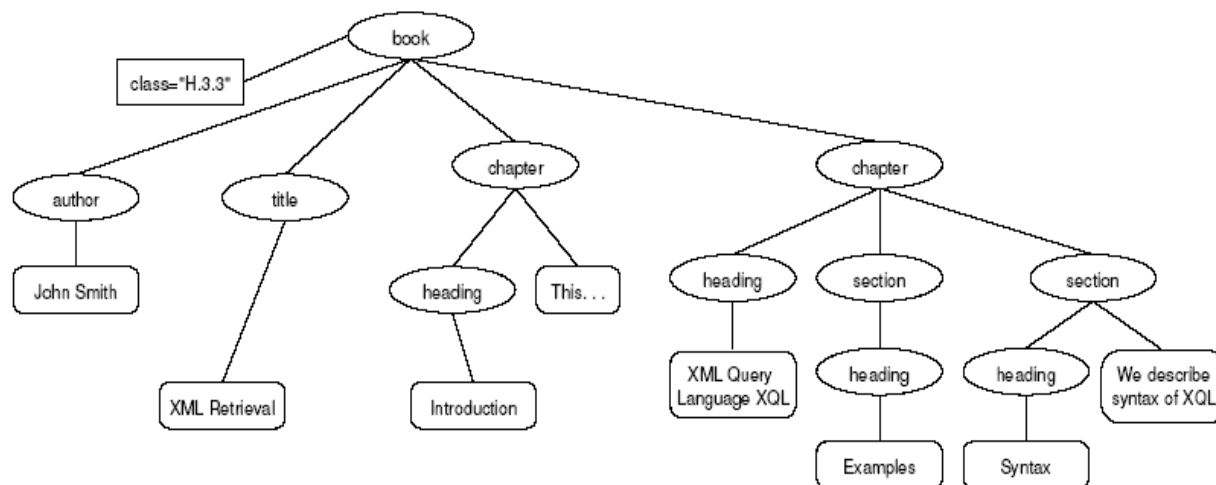
STRUCTURE OF XML DOCUMENTExample of XML document

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE book SYSTEM "/services/dtds/book.dtd">
<book class="H.3.3">
  <author>John Smith</author>
  <title>XML Retrieval</title>
  <chapter>
    <heading>Introduction</heading>
    This text explains all about XML and IR.
  </chapter>
  <chapter>
    <heading> XML Query Language XQL </heading>
    <section>
      <heading>Examples</heading>
    </section>
    <section>
      <heading>Syntax</heading>
      Now we describe the XQL syntax.
    </section>
  </chapter>
</book>

```

Tree structure of XML documents



An Example XML Document

```

<?xml version='1.0' ← Prologue is optional
encoding='iso-8859-1?>
<!DOCTYPE presentation SYSTEM ← Document Type Definition can be
"presentation.dtd"> used to validate the document
<presentation ← One root element
xmlns="http://logilab.fr/slides">
  <info>
    <title>XML and Python ← Namespaces can be used to avoid
Tutorial</title> name clashes
    <author>Alexandre Fayolle</author>
  </info> Elements are properly nested
  <section title="Introductions"> ← Attributes must be quoted
    <slide title="About me">
      <list>
        <entry>My name is ← Data is stored in attribute values or in
<em>Alexandre</em>, I'm French</entry> text nodes
        <entry>I've been using Python for ← It's possible to have mixed contents
the past 3 years</entry>
      </list>
    </slide>
    <!-- add more slides here -->
  </section>
</presentation>

```

Structure of document

- Prolog
 - XML Declaration
 - DTD Declaration
- Root Element (Document)
 - Elements (Nested Elements)
 - Element Attributes and Values
 - Data

The Prologue

The prologue, equivalent to the header in HTML, may include the following:

- ✓ An XML declaration (optional) such as: `<?xml version="1.0"?>`. The XML declaration states which version of XML you are using.
- ✓ A DTD or reference to one (optional). An example reference to an external DTD file: `<!DOCTYPE LONGLIST SYSTEM "langlist.dtd">`
DTD describes the rules your XML document must follow.

- ✓ Processing instructions - An example processing instruction that causes style to be determined by a style sheet: `<?xml-stylesheet type="text/css" href="xmlstyle.css"?>`

The XML prolog is an optional piece of information that must become before the root element when used.

XML Process Instructions

XML process instructions are normally included in the XML document prolog, commonly thought of as the header in HTML. Processing instructions may be placed anywhere in the document so long as they are outside the element tags.

The rules for the names of process instructions are similar to the rules for element names, however in the case of process instructions there are some reserved xml process instructions such as "xml-stylesheet".

The process instruction syntax:

```
<? target instruction ?>
```

The *target* is the application name the instruction is meant for. It can be a reserved value such as "xml-stylesheet" or the name of an external application such as a script program name.

The prolog looks something like this:

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

It tells you that your document follows XML version 1.0, is stand-alone (that is, not accompanied by a document type definition, or DTD), and uses Unicode Transformation Format 8-bit (UTF-8) encoding.

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>
```

The root element, also referred to as the "document". The root element must be the first element in an XML document and there can only be one root element per file!

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

The first line in the document - the XML declaration - defines the XML version and the character encoding used in the document. In this case the document conforms to the 1.0 specification of XML and uses the ISO-8859-1 (Latin-1/West European) character set.

The next line describes the root element of the document (like it was saying: "this document is a note"):

```
<note>
```

The next 4 lines describe 4 child elements of the root (to, from, heading, and body):

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

And finally the last line defines the end of the root element:

```
</note>
```

PRESENTING OR VIEWING 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.

Notepad

You can view the content of the xml file using notepad.

TWO TYPES OF XML DOCUMENTS

- Well Formed XML Documents
- Valid XML Documents

Well Formed XML Documents

A "Well Formed" XML document has correct XML syntax. A *well-formed* XML document—one that conforms to the XML syntax—starts with a *prolog* and contains exactly one *element*.

A "Well Formed" XML document is a document that conforms to the XML syntax rules those are given below:

- XML documents must have a root element [a]
- All other elements are children of the root element
- All elements are correctly paired [b]
- XML elements must have a closing tag [c]
- XML tags are case sensitive [d]
- XML attribute values must always be quoted [e]
- Attribute names are used only once within the same element [f]

Explanation of rule [a]

Consider file f1.xml given below,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date="12/11/2002">
<to>Tove</to>
<from>Jani</from>
</note>
```

This file is well formed according to XML rule [a]. Because there is one root element note. Consider a file f2.xml given below

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date="12/11/2002">
<to>Tove</to>
<from>Jani</from>
</note>
<book>
<title>Introducton to XML</title>
<para>XML is used to represent document</para>
</book>
```

This file is not well formed according to XML rule [a]. Because there is no root element in file f2.xml, Book element is not member of root element.

Example for rule [b]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<a>
  <b>
    <c>
      </b>
    </c>
  </a>
```

in this file element 'b' is closed before closing element 'c'. So this is not a well formed XML document.

Example for rule [c]

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

```
<a>
  <b>
    <c>
  </b>
```

```
</a>
```

in this file element 'c' is not closed using </c> tag. Every element in the XML document must have a starting and closing TAG.

Example for rule [d]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</FROM>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

This is not a well formed XML document. Because 'from' element not closed because <from> and <FROM> are different.

Example for rule [e]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
</note>
<book ref="ISBN 12.1">
<title>Introducton to XML</title>
<para>XML is used to represent document</para>
</book>
</note>
```

is a well formed xml document.

Example for rule [f]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
</note>
<book ref="ISBN 12.1" ref="123">
<title>Introduction to XML</title>
<para>XML is used to represent document</para>
</book>
</note>
```

is not a well formed xml document. Attribute ref used twice in element book.

Valid XML Documents

Valid XML document contains a DTD. **Valid XML document obeys the rules specified in the DTD.** To be of practical use, an XML document needs to be valid. A "Valid" XML document also conforms to a DTD. [we can specify set of rules in DTD, that rules must be obeyed by the XML document]. A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "InternalNote.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

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. Valid XML document is an XML document with elements and attributes conforming to a grammar. This grammar can be described with a DTD, an XML Schema, or some other mean.

- 1) explain different types of XML documents?(valid and well-formed)
- 2) Write the rules for using tags or elements in XML document
- 3) Define Markup language and markup
- 4) write a note on structure of an XML document
- 5) explain prologue instructions
- 6) What is meant by XML document