

Session 23

XML

XML Reading and Reference

- Reading

 - <https://en.wikipedia.org/wiki/XML>

- Reference:

 - XML in a Nutshell (Ch. 1-3), available in Safari On-line

Lecture Objectives

- Understand the goal of application specific markup languages
- Understand XML as a meta language that defines application specific languages
- Understand general concept of tree-structured access to an XML document
- Be familiar with DTDs as a way of defining the rules of an XML document

© Robert Kelly, 2018

3

XML

- Extensible Markup Language
- Set of rules for encoding documents in a format that is readable by humans and machines
- Encountered in
 - Application support files (web.xml, persistence.xml)
 - Industry standards for data exchange
 - Thymeleaf
 - Large complex data standards

© Robert Kelly, 2018

4

XML Document

- Structures textual information
- Does not contain styling information
- Defines a hierarchical structure
- Contains elements and attributes
- Follows basic XML syntax rules
- Usually adheres to a set of domain rules
 - Element names
 - Attribute names
 - Containment rules

© Robert Kelly, 2018

5

Example - Recipe

```
<?xml version="1.0"?>
<!DOCTYPE Recipe SYSTEM "recipe.dtd">
<Recipe>
  <Name>Lime Jello Marshmallow Cottage Cheese Surprise</Name>
  <Description> My grandma's favorite (may she rest in peace).
  </Description>
  <Ingredients>
    <Ingredient>
      <Qty unit="box">1</Qty>
      <Item>lime gelatin</Item>
    </Ingredient>
    <Ingredient>
      <Qty unit="g">500</Qty>
      <Item>multicolored tiny marshmallows</Item>
    </Ingredient>
  </Ingredients>
  <Instructions>
    <Step>Prepare lime gelatin according to package instructions
    </Step>
    <!-- And so on... -->
  </Instructions>
</Recipe>
```

Notice that the element names and attribute names refer to recipes

© Robert Kelly, 2001-2006

6

Well-Formed (Parsable) XML

- **Basic Rules (common to all XML documents)**
 - Document contains only properly encoded Unicode characters
 - No unclosed tags (empty tags use the empty tag symbol)
 - Tags must be properly nested (i.e., no overlapping tags)
 - Tag names are case-sensitive (start and end tags must match precisely)
 - Attribute values must be enclosed in quotes
 - Special syntax characters (e.g., >, <, ", and &) must always be represented by character entities
 - A single root element contains all the other elements

© Robert Kelly, 2018

7

XHTML

- **Extensible Hypertext Markup Language**
- An official W3C recommendation
- Designed to bring the structure and accuracy of XML to HTML
- If an HTML page conforms to an XML DTD you can:
 - Easily extract information
 - Ensure consistent display
 - Convert to other markup languages (i.e., device specific languages)

HTML5 specification includes both an XML version and a non-XML version

© Robert Kelly, 2018

8

XHTML Syntax ...


- **Conforms to XML syntax rules (embedding, null tags, etc.)**
- **Major differences with earlier versions of html:**
 - Elements must be properly nested
 - Documents must be well-formed
 - Tag names and attribute names must be in lower case
 - All elements must be closed
 - Attribute values must be quoted

© Robert Kelly, 2018

9

... XHTML Syntax ...

- **Attribute minimization is forbidden**

<code><dl compact></code>		<code><dl compact="compact"></code>
<code><input checked></code>		<code><input checked="checked"></code>
<code><input readonly></code>		<code><input readonly="readonly"></code>
<code><input disabled></code>		<code><input disabled="disabled"></code>
<code><option selected></code>		<code><option selected="selected"></code>
<code><frame noresize></code>		<code><frame noresize="noresize"></code>

© Robert Kelly, 2018

10

Application-Specific XML Rules

- Rules define each unique XML language (e.g. the simple recipe language)
- Examples of document rules:
 - Names of the elements and attributes
 - Rules for the maximum and minimum number of ingredients in a recipe
 - Rules for the maximum and minimum number of quantities in an ingredient
- Defined in a schema
 - DTD (Document Type Definition)
 - XML Schema
 - Other languages (RELAX NG, Schematron, DSDL, etc.)

© Robert Kelly, 2018

11

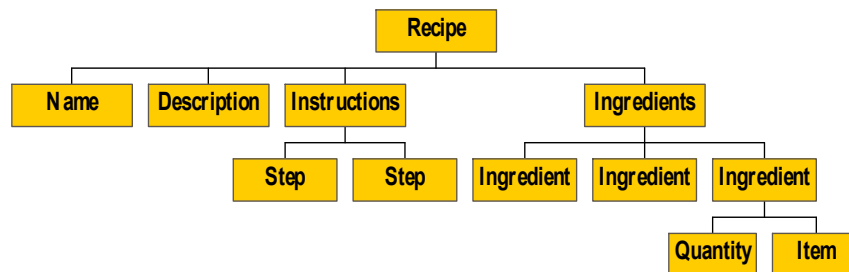
Simple Recipe DTD

```
<!ELEMENT Recipe (Name, Description?, Ingredients?, Instructions?)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Ingredients (Ingredient)*>
<!ELEMENT Ingredient (Qty, Item)>
<!ELEMENT Qty (#PCDATA)>
<!ATTLIST Qty
  unit CDATA #REQUIRED
>
<!ELEMENT Item (#PCDATA)>
<!ATTLIST Item
  optional CDATA "0"
  isVegetarian CDATA "true"
>
<!ELEMENT Instructions (Step)+>
<!ELEMENT Step (#PCDATA)>
```

© Robert Kelly, 2001-2006

12

The Simple Recipe as a Tree



© Robert Kelly, 2018

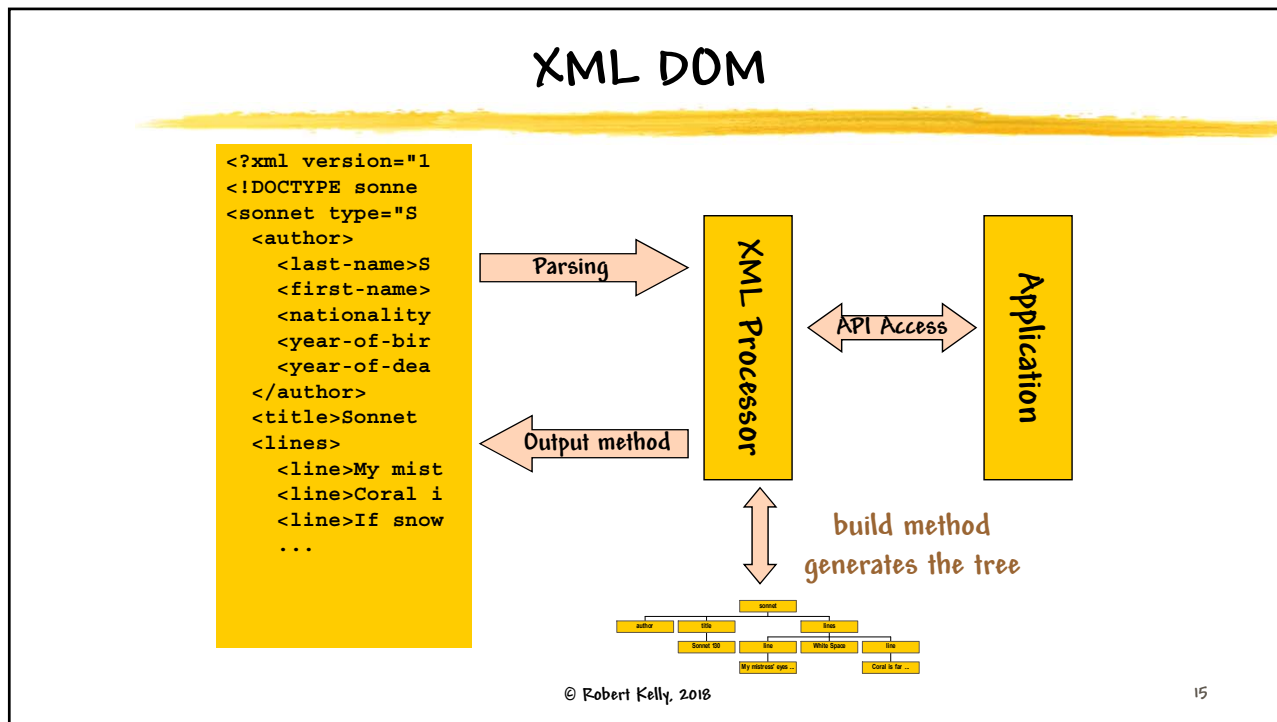
13

Document Object Model (DOM)

- Hierarchical object representation of an XML document
 - Produced by XML parsers
- Your Java/JavaScript program can
 - Extract a given node (element)
 - Walk the tree
 - Search for particular nodes or data (e.g., img tags)
 - Modify the nodes
 - Generate a new document as
 - A DOM object
 - An XML text file

© Robert Kelly, 2018

14



- ## Document Validity
- Well-formed - follows the rules of XML
 - Valid - Corresponds to the specified schema
- © Robert Kelly, 2018 16

XML Schema (XSchema)

- W3C standard
- Individual schemas define a class of XML documents (a schema file usually has an .xsd extension)
- An individual document that conforms to a particular schema is called an instance document

© Robert Kelly, 2018

18

Example - DTD/Schema

```

<!ELEMENT note (to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
    
```

← DTD


```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.w3schools.com"
  xmlns="http://www.w3schools.com"
  elementFormDefault="qualified">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
    
```

Corresponding schema

root →

Namespace declaration →

Corresponds to namespace declaration in XML document →

© Robert Kelly, 2018

19

XML Namespaces

- You might need to use more than one set of vocabularies (element and attribute names) in the same document
- Example - SVG pictures and MathML equations in HTML5 for non-html5 browsers
- Approach: namespaces
- Example

```
<!DOCTYPE html SYSTEM "http://www.thymeleaf.org/dtd/xhtml1-strict-  
thymeleaf-4.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml"   
      xmlns:th="http://www.thymeleaf.org">
```

© Robert Kelly, 2018

20

Namespace Example

- Within the document, you refer to an element or an attribute within a namespace by using the prefix of the namespace

```
<head>  
  <title>Good Thymes Virtual Grocery</title>  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
  <link rel="stylesheet" type="text/css" media="all"   
        href="../../../css/gtvg.css" th:href="@{/css/gtvg.css}" />  
</head>
```

Namespace prefix

Notice the use of the empty tag designator

© Robert Kelly, 2018

21

HTML as XML ...

- Original html was extended from SGML (MIME type of text/html)
 - Various versions, none as well-formed XML
 - DTDs were developed for each version of html - to check validity
- Browsers developed ways to correct errors in a non-standard way
- XML reformulated as XML with XHTML (MIME type of application/xhtml+xml - or text/html)
 - MS IE did not support application/xhtml+xml
- WHATWG developed HTML5

This provides a background for some of the concepts in ThymeLeaf

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd" © Robert Kelly, 2018
```

22

... HTML as XML

- HTML5
 - Evolved from HTML4 and XHTML
 - Error handling included in specification
 - MIME type of text/html
 - No DTD in DOCTYPE tag - rules cannot be express in DTD language

`<!DOCTYPE html>` is the minimum required by a browser

© Robert Kelly, 2018

23

Have You Satisfied the Lecture Objectives?

- Understand the goal of application specific markup languages
- Understand XML as a meta language that defines application specific languages
- Understand general concept of tree-structured access to an XML document
- Be familiar with DTDs as a way of defining the rules of an XML document