XML

(Additional Parts – xml basic)

What is XML?

- XML stands for e**X**tensible **M**arkup **L**anguage
- XML is a markup language much like HTML
- XML was designed to describe data
- XML tags are not predefined. You must define your own tags
- XML uses a Document Type Definition (DTD) or an XML Schema to describe the data
- XML with a DTD or XML Schema is designed to be self-descriptive
- XML is a W3C Recommendation

The Main Difference Between XML and HTML

- XML was designed to carry data.
- XML is not a replacement for HTML.
  XML and HTML were designed with different goals:

  (1) XML was designed to describe data and to focus on what data is.
  (2) HTML was designed to display data and to focus on how data looks.
- HTML is about displaying information, while XML is about describing information.

XML Does not DO Anything

- Maybe it is a little hard to understand, but XML does not DO anything. XML was created to structure, store and to send information.

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

Someone must write a piece of software to send, receive or display it.

XML in Future Web Development

- XML is Free and Extensible

  XML tags are not predefined. You must "invent" your own tags.
- XML is a Complement to HTML

  XML is a cross-platform, software and hardware independent tool for transmitting information.
- XML in Future Web Development

  XML is going to be everywhere.

How can XML be Used?

It is important to understand that XML was designed to store, carry, and exchange data. XML was not designed to display data.

- XML can Separate Data from HTML
- XML is Used to Exchange Data
- XML and B2B
- XML Can be Used to Share Data
- XML Can be Used to Store Data
- XML Can Make your Data More Useful
- XML Can be Used to Create New Languages
- If Developers Have Sense

XML Syntax Rules

- XML documents use a self-describing and simple syntax.

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

XML Syntax Rules

- All XML Elements Must Have a Closing Tag
- XML Tags are Case Sensitive
- XML Elements Must be Properly Nested
- XML Documents Must Have a Root Element
- XML Attribute Values Must be Quoted
- With XML, CR / LF is Converted to LF
- Comments in XML is similar to that in HTML.
- There is Nothing Special About XML

XML Encoding

- XML documents may contain foreign characters. To let your XML parser understand these characters, you should save your XML documents as Unicode.

```
<?xml version="1.0" encoding="ISO-10646"?>
```

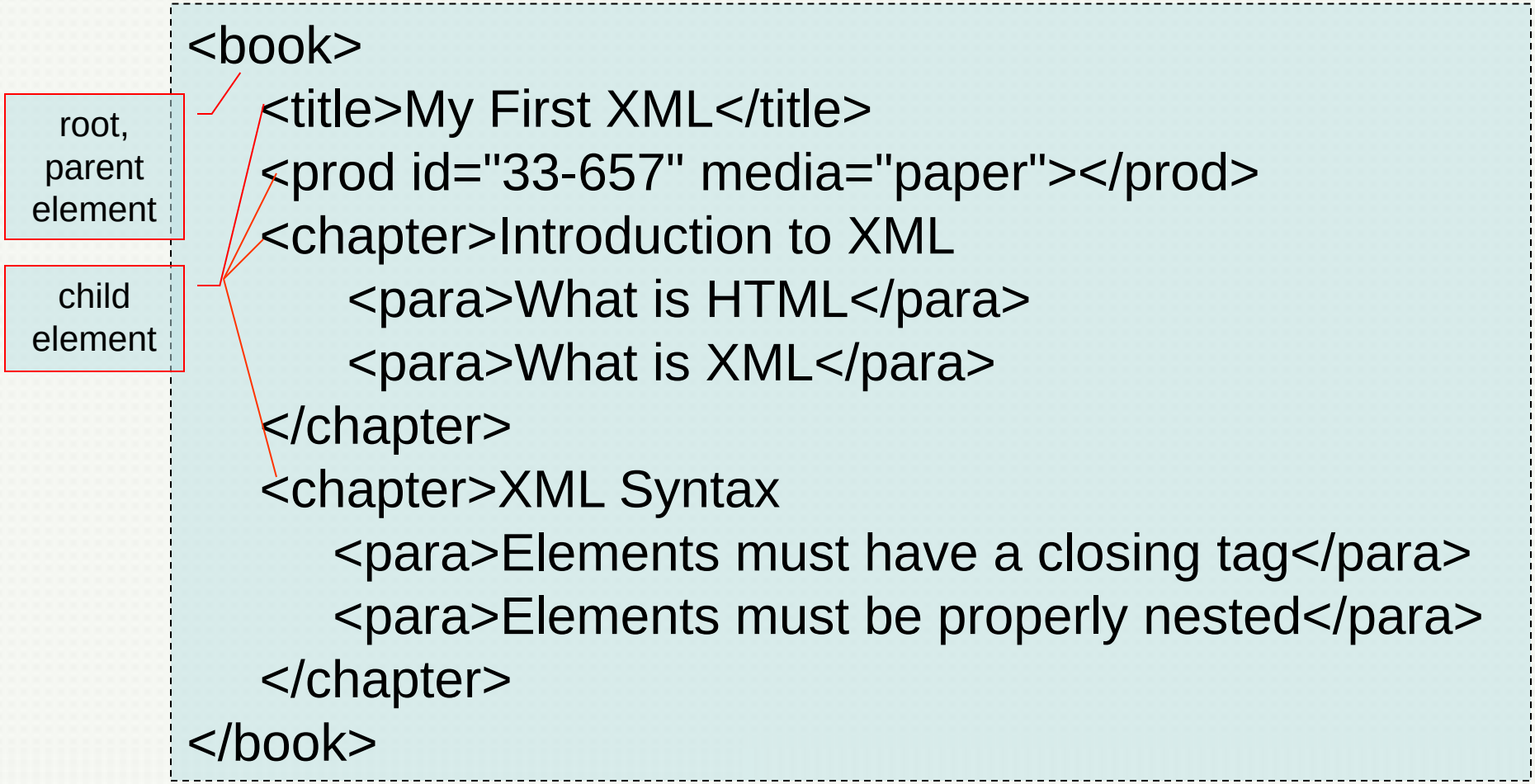
```
<?xml version="1.0" encoding="UCS-4"?>
```

```
<?xml version="1.0" encoding="UTF-16"?>
```

```
<?xml version="1.0" encoding="gb2312"?>
```

XML Elements

- Elements are related as parents and children.



XML Elements

- XML elements must follow these naming rules:

  (1) Names can contain letters, numbers, and other characters

  (2) Names must not start with a number or punctuation character

  (3) Names must not start with the letters xml (or XML, or Xml, etc)

  (4) Names cannot contain spaces

XML Elements

- Any name can be used, no words are reserved, but the idea is to make names descriptive. Names with an underscore separator are nice.

```
<first_name>        <last_name>
```

- Avoid "-" and "." in names.
- A good practice is to use the naming rules of your database for the elements in the XML documents.
- The ":" should not be used in element names because it is reserved to be used for something called namespaces

XML Attributes

- XML elements can have attributes in the start tag, just like HTML.
- Attributes are used to provide additional information about elements.
- Quote Styles, "female" or 'female'?

```
<person sex="female">
<person sex='female'>
```

Note: If the attribute value itself contains double quotes or single quotes

```
<gangster name= 'Bin "Base" Laden'>
<gangster name= "Bin 'Base' Laden">
```

Use of Elements vs. Attributes

- There are no rules. Use child elements if the information feels like data.

```
<person>
<sex>female</sex>
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

```
<person sex="female">
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

XML Validation

- XML with correct syntax is Well Formed XML.
- XML validated against a DTD or a Schema is Valid XML.
- With CSS or XSL you can add display information to your XML document.
- Errors in XML documents will stop your XML program.
- Validate your XML, we can use some software like that Altova's XMLSpy or XMLwriter.

XML Namespaces

- Since element names in XML are not predefined, a name conflict will occur when two different documents use the same element names.

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

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

XML Namespaces

STEP 1:

- Solving Name Conflicts Using a Prefix

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

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

STEP 2: Using 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.furniture.com/furniture">
  <f:name>Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

XML Namespaces

- XML Namespaces Syntax

xmlns:*namespace-prefix*="*namespaceURI*"

- When a namespace is defined in the start tag of an element, all child elements with the same prefix are associated with the same namespace.
- Note that the address used to identify the namespace is not used by the parser to look up information. The only purpose is to give the namespace a unique name.

XML Namespaces

- Default Namespaces

Defining a default namespace for an element 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

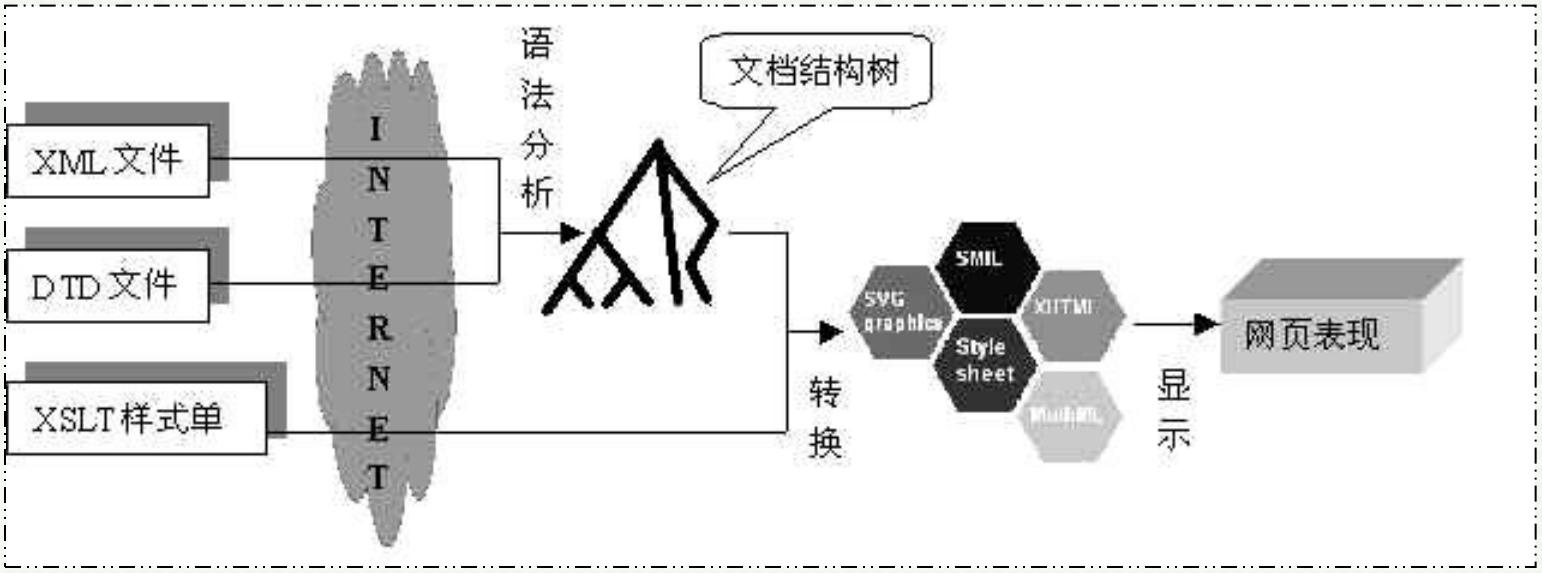
- All text in an XML document will be parsed by the parser. Only text inside a CDATA section will be ignored by the parser.

```
<![CDATA[      content       ]]>
```

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

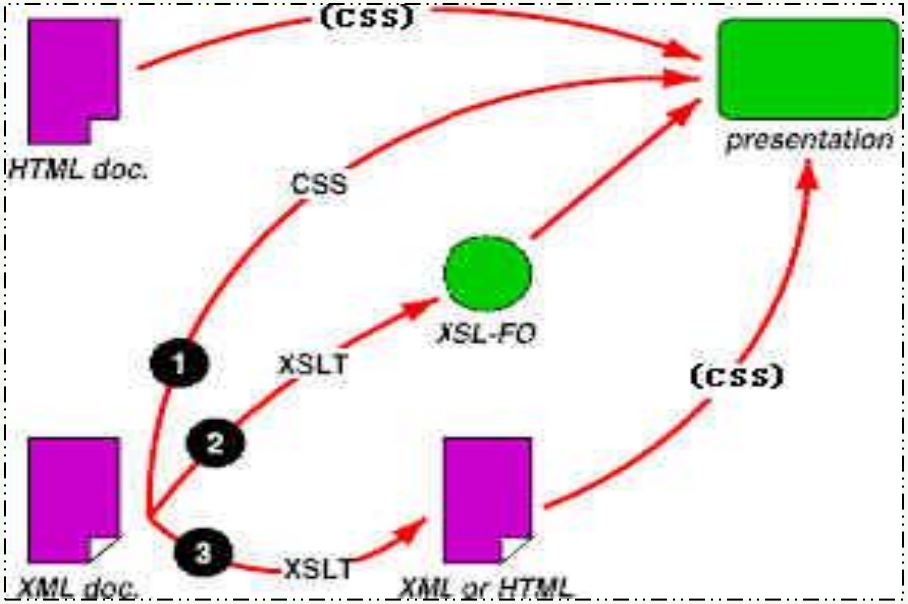
What string can't contain in a CDATA section?

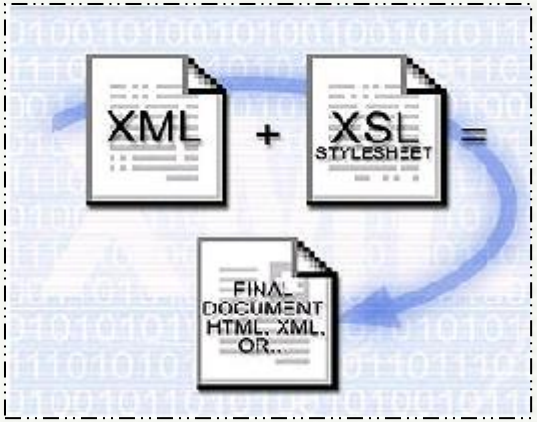
Displaying XML



- Formatting XML with CSS is NOT the future of how to style XML documents. XML document should be styled by using the W3C's XSL standard!

Displaying XML

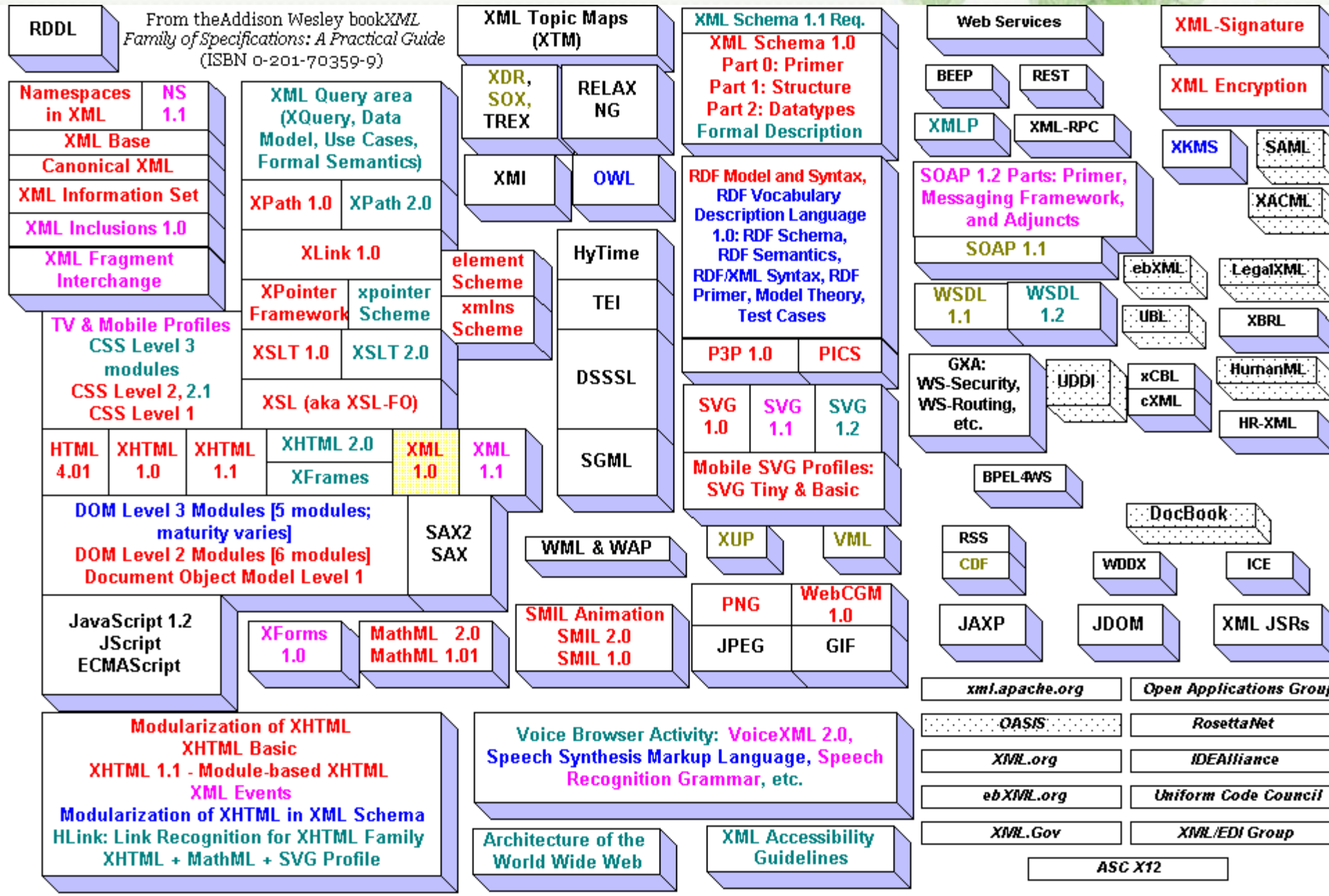





```
<?xml-stylesheet type="text/css" href="style.css"?>
```

```
<?xml-stylesheet type="text/xsl" href="simple.xsl"?>
```

- XSL (the eXtensible Stylesheet Language) is far more sophisticated than CSS.


RDDL
From theAddison Wesley bookXML Family of Specifications: A Practical Guide (ISBN 0-201-70359-9)
XML Topic Maps (XTM)
XML Schema 1.1 Req.
Web Services
XML-Signature
XDR, SOX, TREX
RELAX NG
XML Schema 1.0 Part 0: Primer Part 1: Structure Part 2: Datatypes Formal Description
BEEP
REST
XML Encryption
Namespaces in XML
NS 1.1
XML Query area (XQuery, Data Model, Use Cases, Formal Semantics)
XMI
OWL
XMLP
XML-RPC
XKMS
SAML
XML Base
XPath 1.0
XPath 2.0
RDF Model and Syntax, RDF Vocabulary Description Language 1.0: RDF Schema, RDF Semantics, RDF/XML Syntax, RDF Primer, Model Theory, Test Cases
SOAP 1.2 Parts: Primer, Messaging Framework, and Adjuncts
XACML
Canonical XML
XLink 1.0
element Scheme
HyTime
SOAP 1.1
XML Information Set
XPointer Framework
xpointer Scheme
xmlns Scheme
TEI
ebXML
LegalXML
WSDL 1.1
WSDL 1.2
XML Inclusions 1.0
UBL
XBRL
XML Fragment Interchange
XSLT 1.0
XSLT 2.0
DSSSL
P3P 1.0
PICS
GXA: WS-Security, WS-Routing, etc.
UDDI
xCBL
cXML
HumanML
TV & Mobile Profiles
CSS Level 3 modules
CSS Level 2, 2.1
CSS Level 1
XSL (aka XSL-FO)
SVG 1.0
SVG 1.1
SVG 1.2
HR-XML
HTML 4.01
XHTML 1.0
XHTML 1.1
XHTML 2.0
XFrames
XML 1.0
XML 1.1
SGML
Mobile SVG Profiles: SVG Tiny & Basic
BPEL4WS
DOM Level 3 Modules [5 modules; maturity varies]
DOM Level 2 Modules [6 modules]
Document Object Model Level 1
SAX2
SAX
WML & WAP
XUP
VML
DocBook
RSS
CDF
WDDX
ICE
JavaScript 1.2
JScript
ECMAScript
XForms 1.0
MathML 2.0
MathML 1.01
SMIL Animation
SMIL 2.0
SMIL 1.0
PNG
WebCGM 1.0
JPEG
GIF
JAXP
JDOM
XML JSRs
xml.apache.org
Open Applications Group
OASIS
RosettaNet
XML.org
IDEAlliance
ebXML.org
Uniform Code Council
XML.Gov
XML/EDI Group
ASC X12
Modularization of XHTML
XHTML Basic
XHTML 1.1 - Module-based XHTML
XML Events
Modularization of XHTML in XML Schema
HLink: Link Recognition for XHTML Family
XHTML + MathML + SVG Profile
Voice Browser Activity: VoiceXML 2.0, Speech Synthesis Markup Language, Speech Recognition Grammar, etc.
Architecture of the World Wide Web
XML Accessibility Guidelines

The End