

# XML Poster



Check for the latest version at: <http://xml.itposter.net>

### XML Schema

XML Schema defines the structure and content of XML documents. It consists of three parts:

- XSL Transformations (XSLT)
- The XPath Language
- XML Formatting Objects (XSL-FO)

Diagram illustrating XML elements and attributes:

- XML elements: Complex, Simple, Simple
- XML attributes: User-defined, Built-in types (string, date, int, boolean...), User-defined (restrictions of base type)

Definition of element-organisation (\*sequence\*, \*choice\*, \*all\*, \*group\*)

Definition of element-extension (\*simpleContent\* for attribute extension and \*complexContent\* for element extension)

User-defined (\*restriction\* or \*extension\* of base type)

Built-in types (string, date, int, boolean...)

User-defined (\*restrictions\* of base type)

Namespaces

### XSL & XSLT

The Extensible Stylesheet Language (XSL) is a family of recommendations for defining XML document transformation and presentation. Current version is 1.1.

Diagram illustrating XSL Transform:

Source tree (from XML) → XSL Transform → Result tree (the result of XSL process) → XSL Formatting objects

XSLT is a language for transforming XML documents into other XML documents. XSLT

- Current version: 2.0
- A transformation in the XSLT language is a well-formed XML document.
- A transformation expressed in XSLT is called a stylesheet.
- XSLT transformation describes rules for transforming a source tree into a result tree.
- The transformation is achieved by associating patterns with templates where pattern is matched against elements in source tree.
- More info at: <http://www.w3.org/TR/xslt>

Diagram illustrating XSL-FO Processor:

XML Document → XSL-FO Processor → HTML Document, XML Document, Other Structured Document, PDF Document, RTF Document, Other Presentable Document

XSL-FO is a language for formatting XML documents into formats for paper or screen.

- Current version:

High-level address: <http://www.w3.org/TR/xsl-fo/>

Full application: <http://www.w3.org/TR/xsl-fo/>

Control processing: <http://www.w3.org/TR/xsl-fo/>

Template processing: <http://www.w3.org/TR/xsl-fo/>

Namespaces and parameters: <http://www.w3.org/TR/xsl-fo/>

XML features: <http://www.w3.org/TR/xsl-fo/>

### XML

XML stands for eXtensible Markup Language. XML is a language for describing structure, content, and layout of data in a fashion that is based on standards and can be used anywhere. Its origins are in SGML (Structured Generalised Markup Language), a metalanguage to define different document types. HTML (HyperText Markup Language - the original language of the internet) has the same origins. But the difference is, that HTML is a markup language which is focused on layout and does not preserve the separation of structure, content and presentation. XML is not a programming language although it has some components (like stylesheets) that support formatting and presentation of data. XML is used to define both the rules for describing a particular data type (eg a Person type) and instances of the type (eg John Smith, Mary Jones). The definition of data type is called a Schema and the XML Language for defining types is called XSD (XML Schema Definition). In the past the DTD was used. XML involves only simple text, so that it is easy to transport and use anywhere. It is therefore machine, platform or program language independent and can be transmitted over almost any protocol ([http](http://http), <https>).

Diagram illustrating XML processing:

XML Document → XML Schema (DTD) → XSLT → XML XHTML, WML, ...

XML Document → XPath → XQuery → XSL-FO → PDF, RTF, another printable document

XML Document → XPointer → XLink → XSLT → XML XHTML, WML, ...

### XPath

Language for information search – navigation through XML elements

### XML Query

Language for querying XML data and XML data bases; based on XPath.

Document opening: `doc('book.xml')`

Expressing path: `doc('book.xml')/bookstore/book/title`

Conditions: `doc('book.xml')/bookstore/book[@id=01]`

FLWOR

```

xquery version "1.0";
for $e in
doc('book.xml')/bookstore/book
where $e/@id=01
return $e/title

```

```

xquery version "1.0";
doc('book.xml')/bookstore/book
where $e/@id=01
order by $e/author
return $e/title

```

### DOM (Document Object Model)

XML document is parsed and DOM tree is generated.

Diagram illustrating DOM Parser:

XML Document → DOM Parser → DOM Tree

Diagram illustrating Node:

Node (Parent) → Child Node (Element, Text, Comment, Processing Instruction)

Node (Parent) → Child Node (Text, Comment, Processing Instruction)

### SAX (Simple API For XML)

Parser iterates through XML document and raises events.

Diagram illustrating SAX Parser:

XML Document → SAX Parser → Event

### XML Serialization

Diagram illustrating XML Serialization:

XML Schema → XML Schema Compiling → Classes (Class1, Class2, Class3, Class N) → Objects → XML Document

XML Document → Unmarshalling → Objects → XML Document

XML Document → Marshalling → Objects → XML Document

### XLink, XPointer

Connecting XML documents; together defining a standard for creating links in XML documents. Using XPath expressions to navigate through XML document.

XML Linking Language

- Language for creating hyperconnections in XML documents
- Each XML element can behave as Xlink
- Supporting simple connections (single source connections) and complex connections (multiple source connection)
- Connections can be defined outside connected documents

Example: `<website xlink:type="simple" xlink:href="http://lisa.si/*">LISA</website>`

XML Pointer Language

- Enabling hyperlinks to show specific parts of XML documents

Example: `xlink:href="http://lisa.si/book1.xml#Football"`

### XHTML

### Literature used

### About the XML Poster

This poster is licenced as Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 Slovenia License

Authors:

Boštjan Šumak  
Maja Pušnik  
Email: [info@itposter.net](mailto:info@itposter.net)

University of Maribor  
Faculty of Electrical Engineering and Computer Science  
Institute of Informatics

Poster version: 0.1 (release early)