Telerik Academy

XML Basics

Basic XML Concepts, Schemas, XPath, XSL

Telerik Software Academy
Learning & Development Team
http://academy.telerik.com

Telerik Academy

Table of Contents

Telerik Academy

What is XML?


```
<?xml version="1.0"?>
<library name=".NET Developer's Library">
  <book>
    <title>Programming Microsoft .NET</title>
    <author>Jeff Prosise</author>
    <isbn>0-7356-1376-1</isbn>
  </book>
  <book>
    <title>Microsoft .NET for Programmers</title>
    <author>Fergal Grimes</author>
    <isbn>1-930110-19-7</isbn>
  </book>
</library>
```

What is XML?

- XML (eXtensible Markup Language)
  - Universal language (notation) for describing structured data using text with tags
  - The data is stored together with the meta-data about it
  - Used to describe other languages (formats) for data representation
- XML looks like HTML
  - Text based language, uses tags and attributes

What is XML? (2)

- Worldwide standard, supported by the W3C (www.w3c.org)
- Independent of the hardware platform, OS and programming languages

XML – Example


```
<?xml version="1.0"?>
<library name=".NET Developer's Library">
  <book>
    <title>Professional C# 4.0 and .NET 4</title>
    <author>Christian Nagel</author>
    <isbn>978-0-470-50225-9</isbn>
  </book>
  <book>
    <title>Silverlight in Action</title>
    <author>Pete Brown</author>
    <isbn>978-1-935182-37-5</isbn>
  </book>
</library>
```

XML – Example (2)

```
<?xml version="1.0"?>
<library name=".NET Developer's Library">
  <book>
    <title>Professional C# 4.0 and .NET 4</title>
    <author>Christian Nagel</author>
    <isbn>978-0-470-50225-9</isbn>
  </book>
  <book>
    <title>Silverlight in Action</title>
    <author>Pete Brown</author>
    <isbn>978-1-935182-37-5</isbn>
  </book>
</library>
```

XML header tag (prolog)

Attribute (key / value pair)

Root (document) element

Opening tag

Element

Closing tag

Element value

XML and HTML

- Similarities between XML and HTML
  - Both are text based notations
  - Both use tags and attributes
- Differences between XML and HTML
  - HTML is a language, and XML is a syntax for describing other languages
  - HTML describes the formatting of information, XML describes structured information
  - XML requires the documents to be well-formatted

Well-Formatted XML Documents

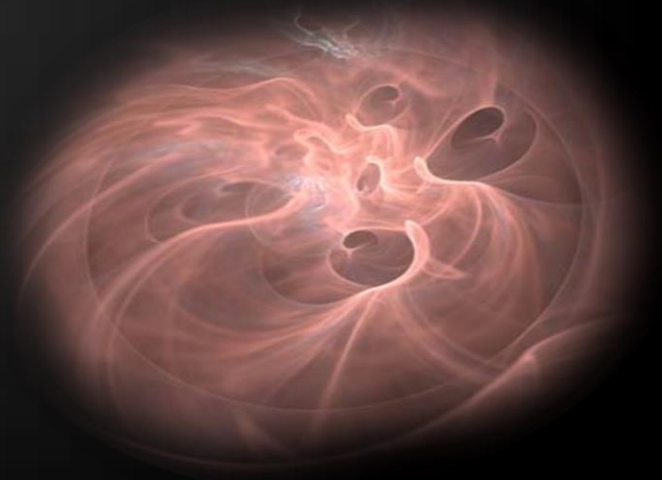
- Well-formatted XML:
  - All tags should be closed in the correct order of nesting
  - Attributes should always be closed
  - The document should contain only one root element
  - Tag and attribute names retain certain restrictions

Well-Formed XML Documents (2)

- Example of incorrect XML document

```
<xml>
    <button bug! value="OK name="b1">
        <animation source="demo1.avi"> 1 < 2 < 3
    </click-button>
< / xml >
```

- Open / close tags do not match
- Unclosed attribute values
- Invalid characters
- …



When to Use XML?

- Advantages of XML:
  - XML is human readable (unlike binary formats)
  - Any kind of structured data can be stored
  - Data comes with self-describing meta-data
  - Custom XML-based languages can be developed for certain applications
  - Information can be exchanged between different systems with ease
  - Unicode is fully supported

When to Use XML? (2)

- Disadvantages of XML:
  - XML data is bigger (takes more space) than in binary formats
    - More memory consumption, more network traffic, more hard-disk space
  - Decreased performance
    - Need of parsing / constructing the XML tags
- XML is not suitable for all kinds of data
  - E.g. graphics, images and video clips

Telerik Academy

XML Namespaces

Namespaces

- XML namespaces are defined like this:

```
<?xml version="1.0" encoding="utf-8"?>
<sample:towns
  xmlns:sample="http://www.academy.com/towns/1.0">
  <sample:town>
    <sample:name>Sofia</sample:name>
    <sample:population>1200000</sample:population>
  </sample:town>
  <sample:town>
     <sample:name>Plovdiv</sample:name>
     <sample:population>700 000</sample:population>
  </sample:town>
</sample:towns>
```

Namespaces (2)

- Namespaces in XML documents allow using different tags with the same name:

```
<?xml version="1.0" encoding="UTF-8"?>
<country:towns
    xmlns:country="urn:academy-com:country"
    xmlns:town="http://www.academy.com/towns/1.0">
  <town:town>
    <town:name>Plovdiv</town:name>
    <town:population>700 000</town:population>
    <country:name>Bulgaria</country:name>
  </town:town>
</country:towns>
```

Namespaces – Example

Namespace with prefix "country" and URI "urn:academy-com:country"

```
<?xml version="1.0" encoding="UTF-8"?>
<country:towns xmlns:country="urn:academy-com:country"
     xmlns:town="http://www.academy.com/towns/1.0">
  <town:town>
    <town:name>Sofia</town:name>
    <town:population>1 200 000</town:population>
    <country:name>Bulgaria</country:name>
  </town:town>
  <town:town>
    <town:name>Plovdiv</town:name>
      <town:population>700 000</town:population>
      <country:name>Bulgaria</country:name>
  </town:town>
</country:towns>
```

Tag named "name" from namespace "country" → the full tag name is "urn:academy-com:country:name"

Namespaces – Example (2)

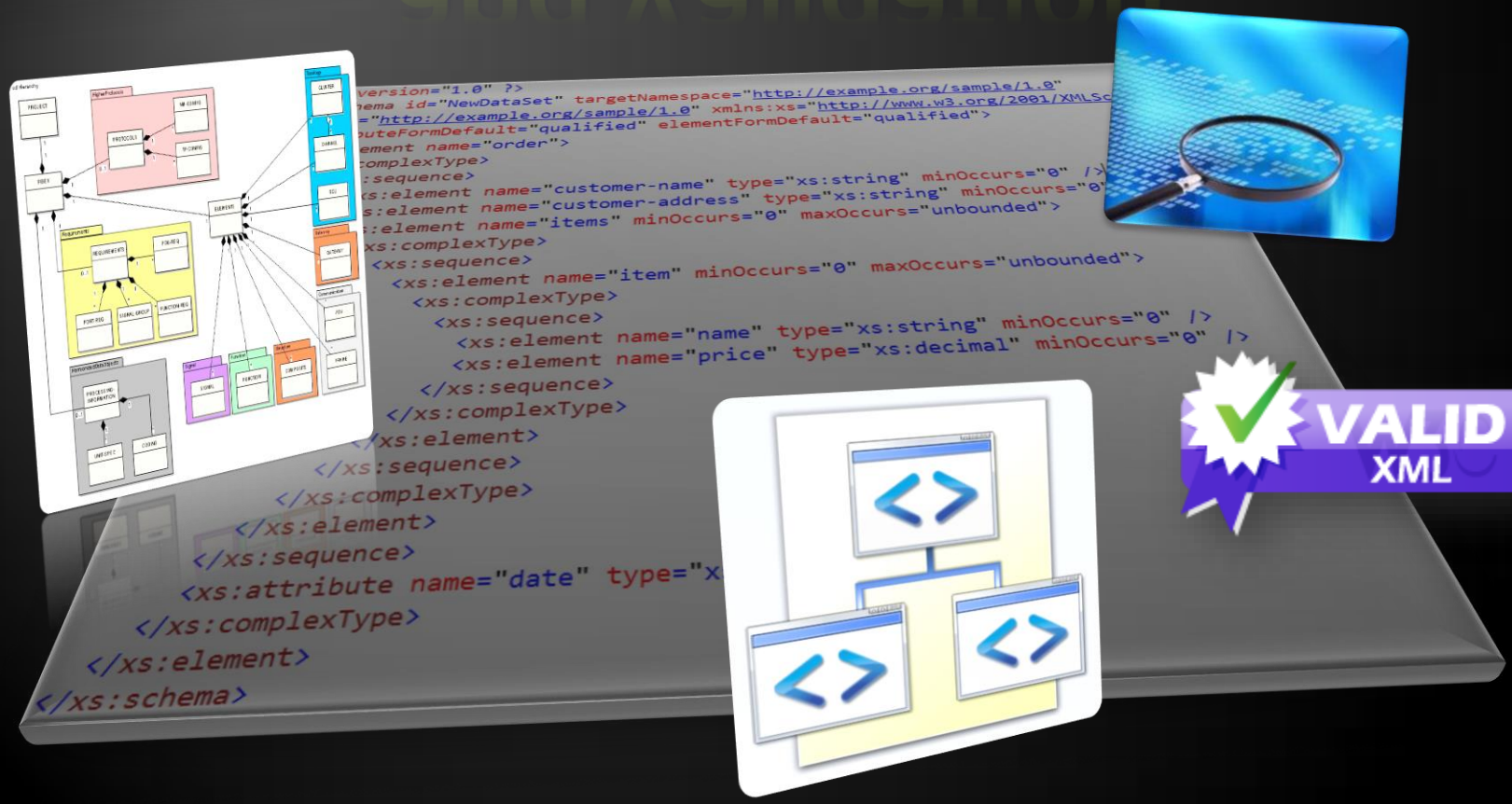
- Default namespaces can save some code:

```
<?xml version="1.0" encoding="windows-1251"?>
<order xmlns="http://www.supermarket.com/orders/1.1">
  <item>
    <name>Beer "Zagorka"</name>
    <ammount>8</ammount>
    <measure>bottle</measure>
    <price>5.60</price>
  </item>
  <item>
    <name>Meat balls</name>
    <ammount>12</ammount>
    <measure>amount</measure>
    <price>5.40</price>
  </item>
</order>
```

Default namespace – applied for the entire XML document

Telerik Academy

XML Schemas and Validation



Telerik Academy

XML Schemas and Validation

- The XML documents structure is defined by schemas
- XML schemas describes:
  - Possible attributes and tags and their values
  - Tags order
  - Default values and number of appearances
- There are few XML Schema standards:
  - DTD – Document Type Definition
  - XSD – XML Schema Definition Language

The DTD Language

- DTD (Document Type Definition) is:
  - Formal language for describing XML document structures
  - Contains a set of rules for the tags and their attributes in a document
  - Text-based language, but not XML based
  - Substituted by XSD (relatively rarely used)

```
<!ELEMENT library (book+)>
<!ATTLIST library name CDATA #REQUIRED>
<!ELEMENT book (title, author, isbn)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
```

XSD Schemas

- XSD (XML Schema Definition Language)
  - Powerful XML-based language for describing the structure of XML documents
  - Contains a set of rules for the tags and their attributes in a document
  - Specifies few standard data types
    - Numbers, dates, strings, etc.
  - XSD Schemas have greater descriptive power than DTD

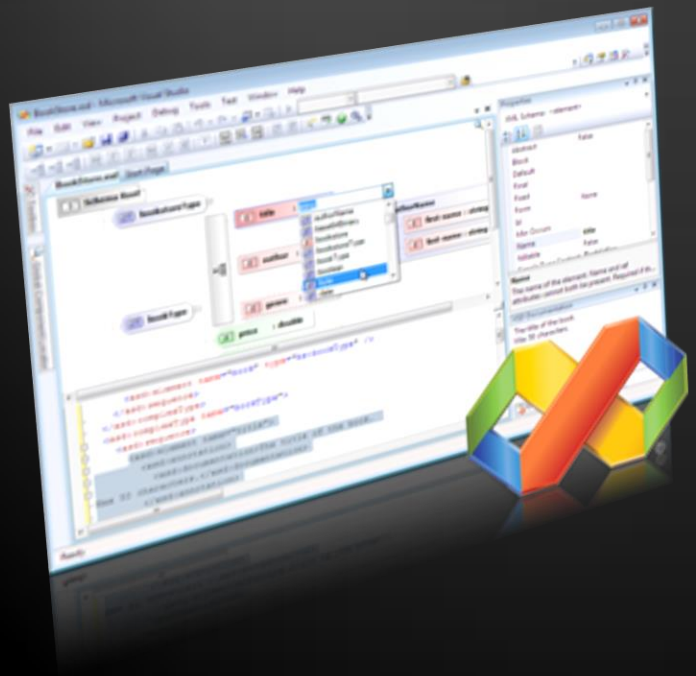
Telerik Academy

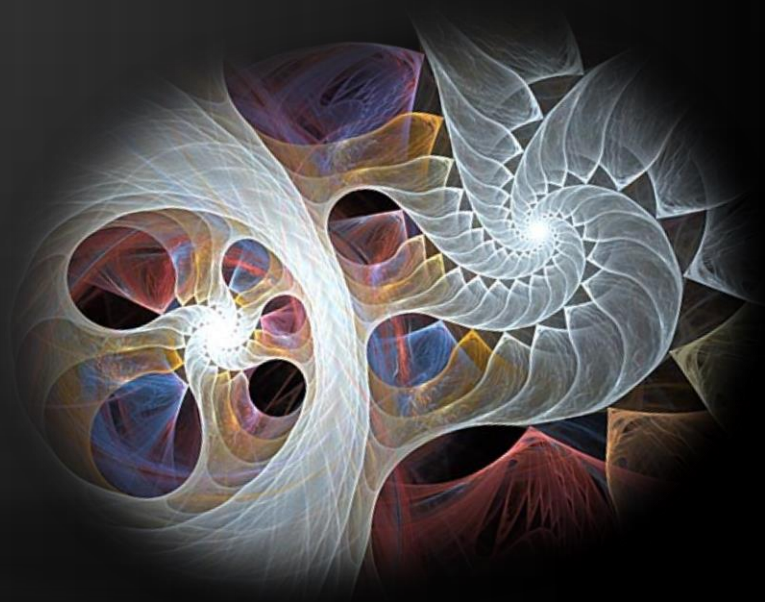
XSD Schemas – Example

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="https://telerikacademy.com">
    <xs:element name="library">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="book"
                    maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="name"
                type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="book">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="title"/>
                <xs:element ref="author"/>
                <xs:element ref="isbn"/>
            </xs:sequence>
```

XSD Schemas – Example (2)

```
        </xs:complexType>
    </xs:element>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="author" type="xs:string"/>
    <xs:element name="isbn" type="xs:string"/>
</xs:schema>
```





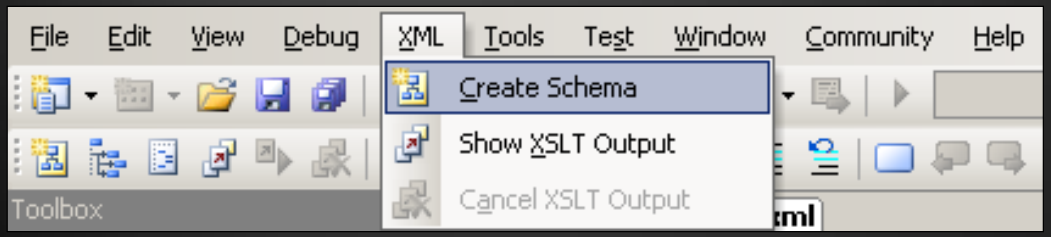
Reference DTD and XSD

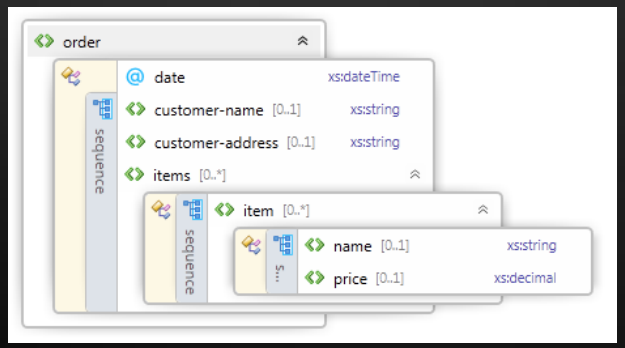
```
<?xml version="1.0"?>
<!DOCTYPE library SYSTEM "[file path]">
<library>
  <book>
    <title>Professional C# 4.0 and .NET 4</title>
    <author>Christian Nagel</author>
    <isbn>978-0-470-50225-9</isbn>
  </book>
</library>
```

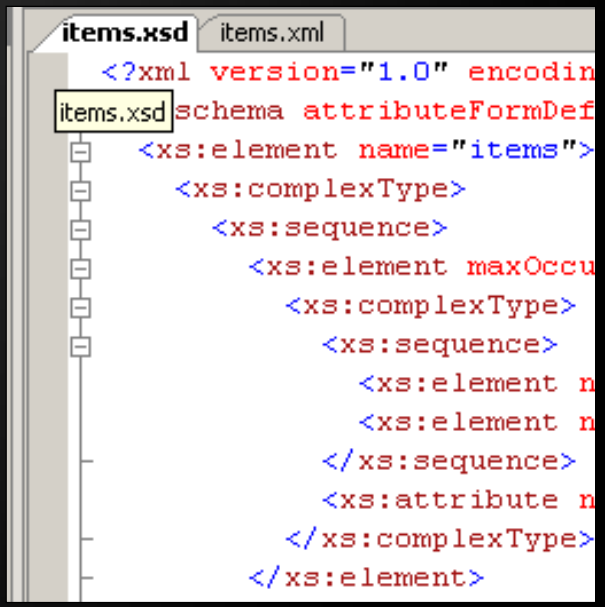
```
<?xml version="1.0"?>
<library xmlns="[namespace]"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="[namespace path]">
  <book>
    <title>Professional C# 4.0 and .NET 4</title>
    <author>Christian Nagel</author>
    <isbn>978-0-470-50225-9</isbn>
  </book>
</library>
```

Visual Studio Schema Editor

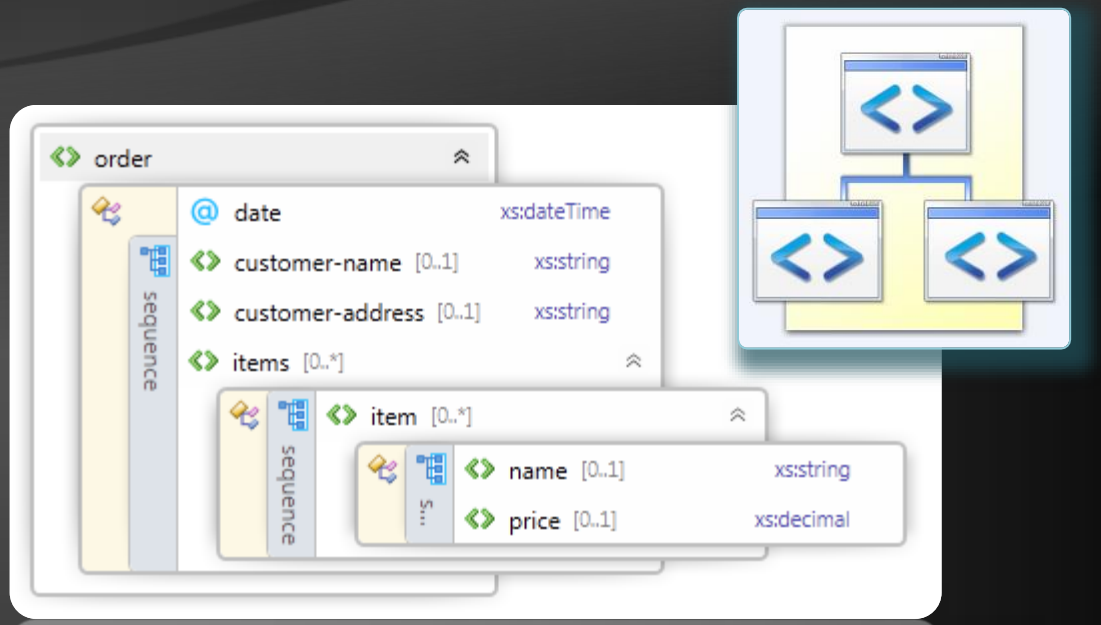
- Visual Studio can generate XSD Schemas from the structure of a given XML document
- VS has powerful XSD Schema Editor
  - Visually edit schemas







Telerik Academy

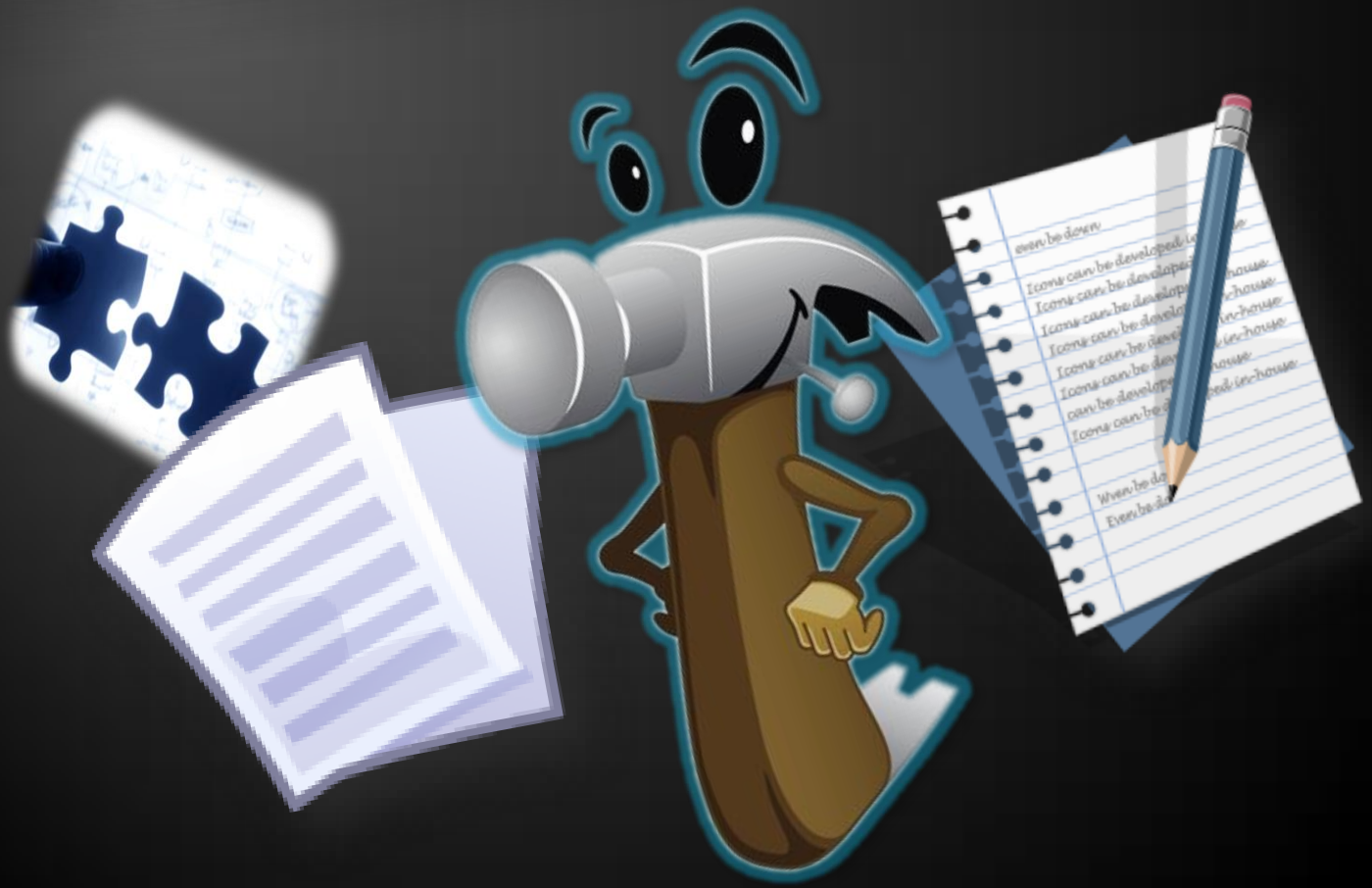


Working with the XSD Editor in Visual Studio

Live Demo

Telerik Academy

XML Parsers



XML Parsers

- XML parsers are programming libraries that make the work with XML easier
- They serve for:
  - Extracting data from XML documents
  - Modifying existing XML documents
  - Building new XML documents
  - Validating XML documents by given schema
  - Transforming XML documents

Telerik Academy

XML Working Models

- They have several working models:
  - DOM (Document Object Model)
    - Represents XML documents as a tree in the memory
    - Allows processing and modifying the document
  - SAX (Simple API for XML Processing)
    - Reads XML documents consequently tag by tag
    - Event-driven API
    - Allows analyzing the read portions at each step
  - StAX (Streaming API for XML)
    - Like SAX but works in "pull" mode

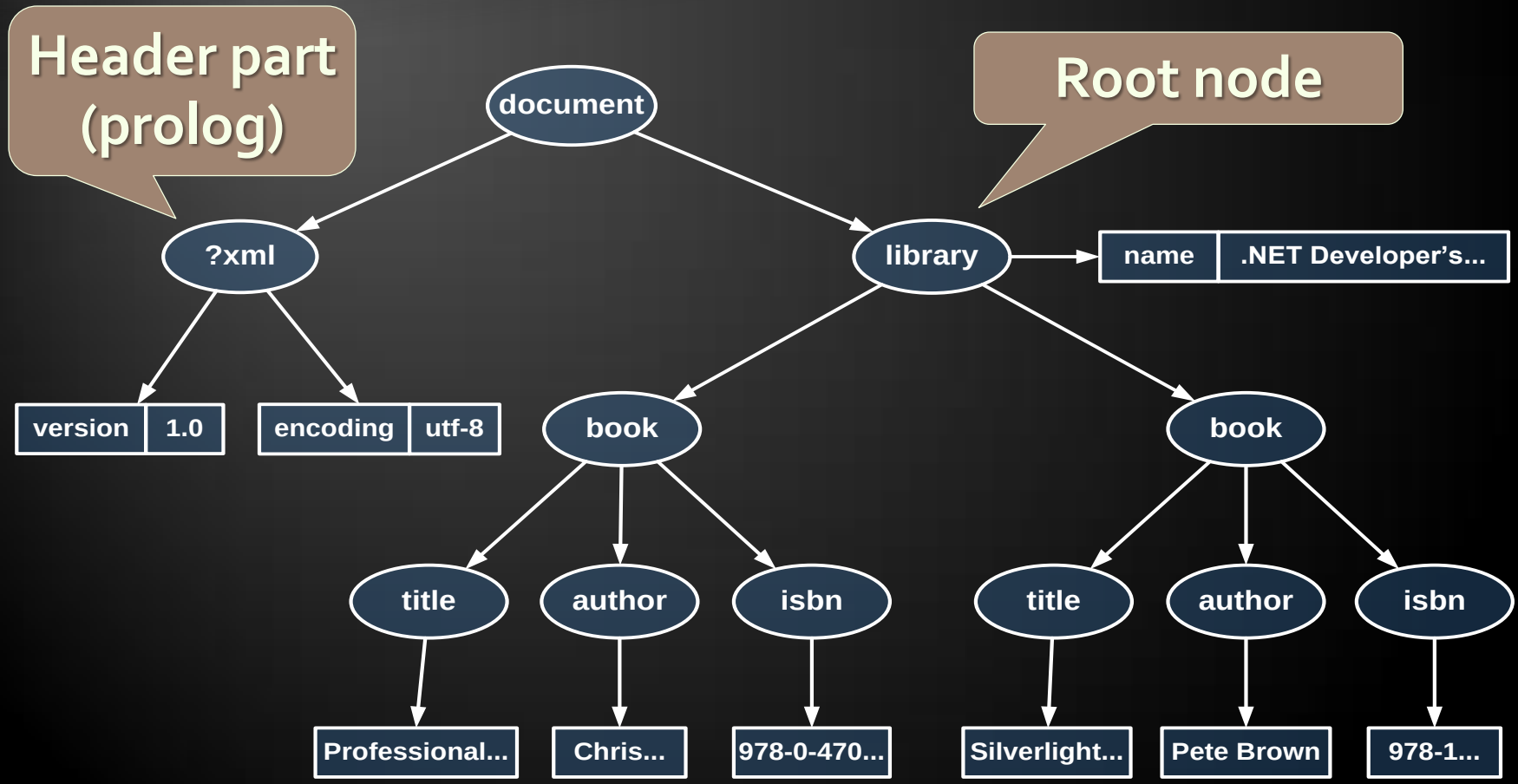
The DOM Parser

- Given the following XML document:

```
<?xml version="1.0"?>
<library name=".NET Developer's Library">
  <book>
    <title>Professional C# 4.0 and .NET 4</title>
    <author>Christian Nagel</author>
    <isbn>978-0-470-50225-9</isbn>
  </book>
  <book>
    <title>Silverlight in Action</title>
    <author>Pete Brown</author>
    <isbn>978-1-935182-37-5</isbn>
  </book>
</library>
```

The DOM Parser (2)

- This document is represented in the in the memory as a DOM tree in the following way:



The SAX Parsers

- The SAX parsers:
  - Iterate the document consequently tag by tag
  - Invoke callback functions (events) when particular node is reached
  - Stream-like access – going backwards or jumping ahead is not allowed
  - Require many times less resources (memory and loading time)
  - Work well over streams

The StAX Parser

- StAX is like SAX but
  - "Pull"-based parser
  - Not event driven (not callback based)
  - Developer manually say "go to next element" and analyze it
- SAX vs. StAX
  - SAX reads the documents and invokes callbacks like "node found", "attribute found", etc.
  - In StAX parsers the read is invoked by the developer synchronously

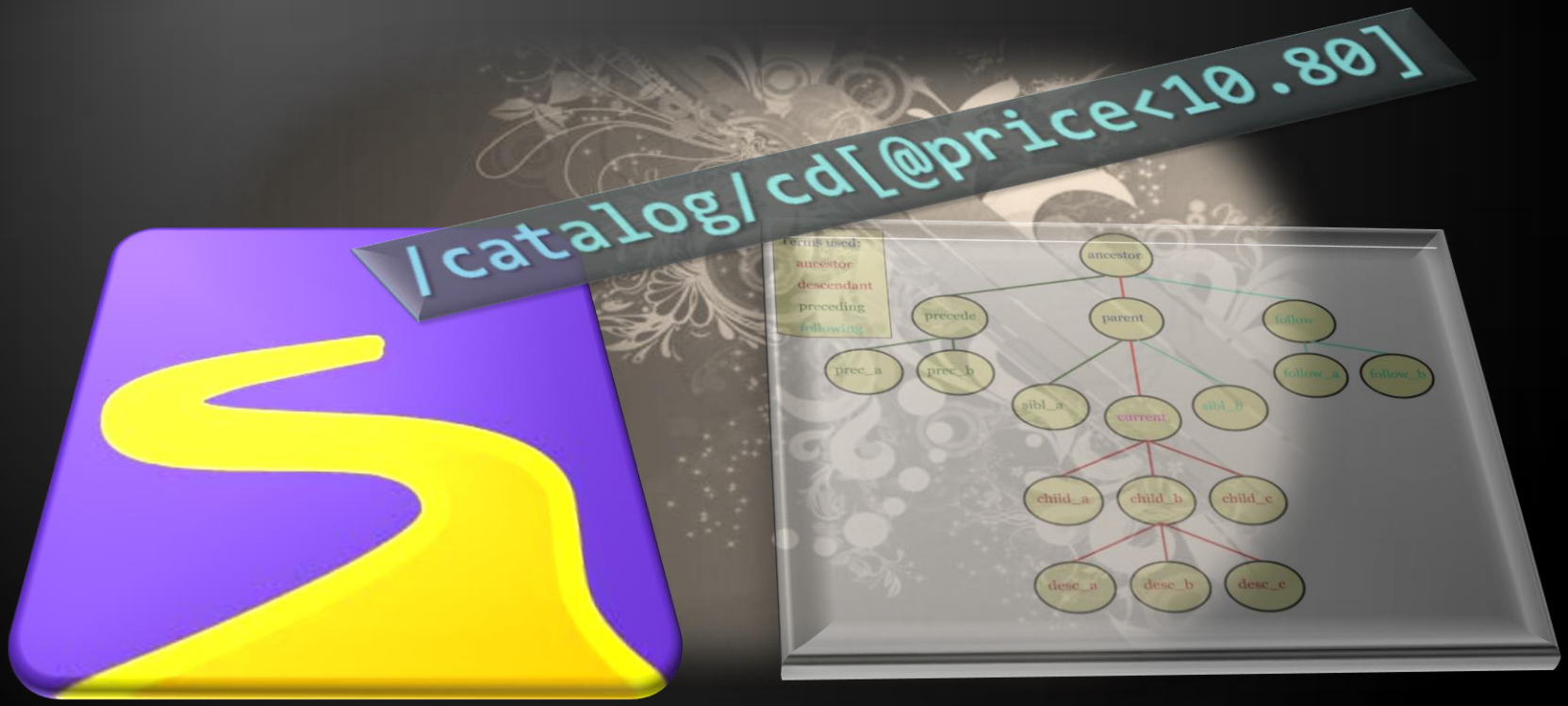
Telerik Academy

When to Use DOM and When to Use SAX / StAX?

- Use DOM processing model when:
  - Process not big documents (e.g. less than 10 MB)
  - There is a need of flexibility
  - To modify XML documents
- Use SAX / StAX processing model when:
  - Process big documents
  - The performance is important
  - There is no need to change the document

Telerik Academy

XPath



XPath

- XPath (XML Path Language) is a language for addressing parts of XML documents
- XPath expressions contains description of paths to nodes and filter criteria
- Example for XPath expressions:

```
/library/book[isbn='1-930110-19-7']
```

```
/catalog/cd[@price<10.80]
```

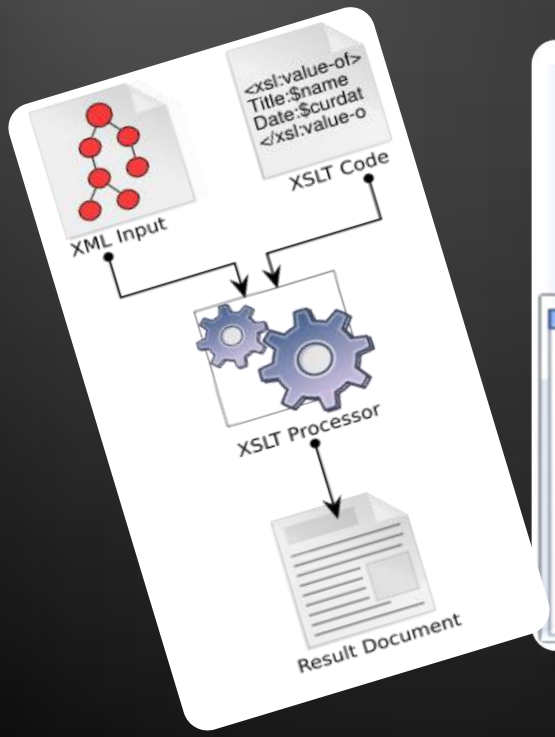
```
/book/chapter[3]/paragraph[last()]
```

XPath Expressions

- / – addresses the root element derived
- /someNode – addresses all nodes with name "someNode", straight inheritors to the root
- /books/book – addresses all nodes "book", straight inheritors to the node "books"
- /books/book[price<10]/author – addresses all authors (/books/book/author), whose books have price < 10
- /items/item[@type="food"] – addresses all nodes with name item (/items/item/) , which have attribute "type" with value "food"

Telerik Academy

XSL Transformations



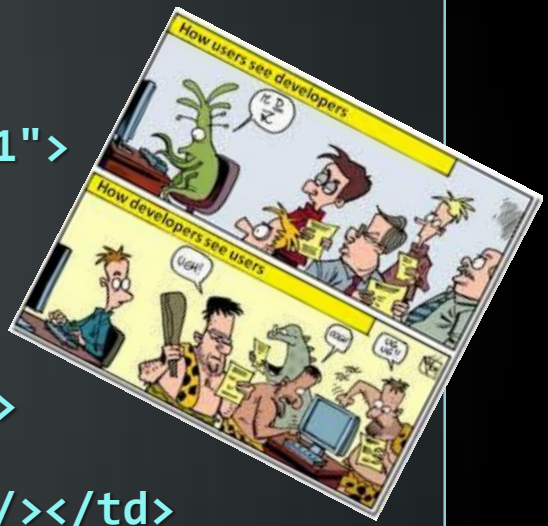
Telerik Academy

XSL Transformations

- XSL transformations (XSLT) allows to
  - Transform of one XML document to other XML document with different structure
  - Convert between different XML formats
- XSLT depends on XPath
  - XPath is used to match parts from the input document and replace them with another XML
- In particular XSLT can be used for transforming XML documents to XHTML

XSL Stylesheet – Example

```
<?xml version="1.0" encoding="windows-1251"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
    <body>
      <h1>My library</h1>
      <table bgcolor="#E0E0E0" cellspacing="1">
        <tr bgcolor="#EEEEEE">
          <td><b>Title</b></td>
          <td><b>Author</b></td>
        </tr>
        <xsl:for-each select="/library/book">
          <tr bgcolor="white">
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="author"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```



The XSL Language

- <xsl:template match="XPath expr."> … </xsl:template> – replaces the pointed with XPath expression part of the document with the construction body
- <xsl:for-each select="XPath expr."> … </xsl:for-each> – replaces each node, pointed by the XPath expression with construction body
- <xsl:value-of select="XPath expr."/> – extracts the value of the given XPath expression (the first occurrence)
- XSL patterns are valid XML documents!

Telerik Academy

XML Basics

Questions?

http://academy.telerik.com

Exercises

1. What does the XML language represents? What does it used for?
2. Create XML document `students.xml`, which contains structured description of students. For each student you should enter information for his name, sex, birth date, phone, email, course, specialty, faculty number and a list of taken exams (exam name, tutor, score).
3. What does the namespaces represents in the XML documents? What are they used for?

Exercises (2)

4. Explore http://en.wikipedia.org/wiki/Uniform_Resource_Identifier to learn more about URI, URN and URL definitions.
5. Add default namespace for the students "urn:students".
6. Create XSD Schema for the `students.xml` document. Add new elements in the schema: information for enrollment (date and exam score) and teacher's endorsements.
7. Write a XSL stylesheet to visualize the students as HTML. Test it in your favourite browser.

Free Trainings @ Telerik Academy

- "Web Design with HTML 5, CSS 3 and JavaScript" course @ Telerik Academy
  - html5course.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com