**XML Schema**

**Language Activity**

XML Schemas are much more powerful than DTDs.

## XML Schemas Support Data Types

One of the greatest strengths of XML Schemas are their support for data types.

With support for data types:

* It is easier to describe allowable document content
* It is easier to validate the correctness of data
* It is easier to work with data from a database
* It is easier to define data facets (restrictions on data)
* It is easier to define data patterns (data formats)
* It is easier to convert data between different data types

## XML Schemas use XML Syntax

Another great strength of XML Schemas is that they are written in XML.

Some benefits of XML Schemas being written in XML include:

* You don't have to learn a new language
* You can use your XML editor to edit your Schema files
* You can use your XML parser to parse your Schema files
* You can manipulate your Schema with the XML DOM
* You can transform your Schema with XSLT

## XML Schemas Secure Data Communication

When sending data from a sender to a receiver, it is essential that both parts have the same "expectations" about the content.

With XML Schemas, the sender can describe the data in a way that the receiver will understand.

A date like: "03-11-2004" will, in some countries, be interpreted as 3.November and in other countries as 11.March.

However, an XML element with a data type like this:

<date type="date">2004-03-11</date>

ensures a mutual understanding of the content, because the XML data type "date" requires the format "YYYY-MM-DD".

## XML Schemas are Extensible

XML Schemas are extensible, because they are written in XML.

With an extensible Schema definition you can:

* Reuse your Schema in other Schemas
* Create your own data types derived from the standard types
* Reference multiple schemas in the same document

## Well-Formed is not Enough

A well-formed XML document is a document that conforms to the XML syntax rules, like:

* it must begin with the XML declaration
* it must have one unique root element
* start-tags must have matching end-tags
* elements are case sensitive
* all elements must be closed
* all elements must be properly nested
* all attribute values must be quoted
* entities must be used for special characters

Even if documents are well-formed they can still contain errors, and those errors can have serious consequences.

Think of the following situation: you order 5 gross of laser printers, instead of 5 laser printers. With XML Schemas, most of these errors can be caught by your validating software.

**Questions**

1. What is the main advantage of XML-Schema compared with DTD?

2. Could you imagine a situation where you decide to validate with XML-Schema instead of DTD?

3. Do you think that XML-Schema became the major XML definition language for loading data into databases?Why?

4. Why are XML Schema extensible?

5. What is the definition of a well-formed XML document?

6. How do XML Schemas correct the designer´s errors?

**XML Schema**

**Language Activity**

XML Schemas are much more powerful than DTDs.

## XML Schemas Support Data Types

One of the greatest strengths of XML Schemas are their support for data types.

With support for data types:

* It is easier to describe allowable document content
* It is easier to validate the correctness of data
* It is easier to work with data from a database
* It is easier to define data facets (restrictions on data)
* It is easier to define data patterns (data formats)
* It is easier to convert data between different data types

## XML Schemas use XML Syntax

Another great strength of XML Schemas is that they are written in XML.

Some benefits of XML Schemas being written in XML include:

* You don't have to learn a new language
* You can use your XML editor to edit your Schema files
* You can use your XML parser to parse your Schema files
* You can manipulate your Schema with the XML DOM
* You can transform your Schema with XSLT

## XML Schemas Secure Data Communication

When sending data from a sender to a receiver, it is essential that both parts have the same "expectations" about the content.

With XML Schemas, the sender can describe the data in a way that the receiver will understand.

A date like: "03-11-2004" will, in some countries, be interpreted as 3.November and in other countries as 11.March.

However, an XML element with a data type like this:

<date type="date">2004-03-11</date>

ensures a mutual understanding of the content, because the XML data type "date" requires the format "YYYY-MM-DD".

## XML Schemas are Extensible

XML Schemas are extensible, because they are written in XML.

With an extensible Schema definition you can:

* Reuse your Schema in other Schemas
* Create your own data types derived from the standard types
* Reference multiple schemas in the same document

## Well-Formed is not Enough

A well-formed XML document is a document that conforms to the XML syntax rules, like:

* it must begin with the XML declaration
* it must have one unique root element
* start-tags must have matching end-tags
* elements are case sensitive
* all elements must be closed
* all elements must be properly nested
* all attribute values must be quoted
* entities must be used for special characters

Even if documents are well-formed they can still contain errors, and those errors can have serious consequences.

Think of the following situation: you order 5 gross of laser printers, instead of 5 laser printers. With XML Schemas, most of these errors can be caught by your validating software.

**Questions**

1. What is the main advantage of XML-Schema compared with DTD?

2. Could you imagine a situation where you decide to validate with XML-Schema instead of DTD?

3. Do you think that XML-Schema became the major XML definition language for loading data into databases?Why?

4. Why are XML Schema extensible?

5. What is the definition of a well-formed XML document?

6. How do XML Schemas correct the designer´s errors?