**XSD (XML Schema Definition)**

**XSD Introduction**

**What is an XML Schema?**

An XML Schema describes the structure of an XML document. The XML Schema language is also referred to as XML Schema Definition (XSD).

The purpose of an XML Schema is to define the legal building blocks of an XML document:

* the elements and attributes that can appear in a document
* the number of (and order of) child elements
* data types for elements and attributes
* default and fixed values for elements and attributes

**Why Learn XML Schema?**

In the XML world, hundreds of standardized XML formats are in daily use.

Many of these XML standards are defined by XML Schemas.

XML Schema is an XML-based (and more powerful) alternative to DTD.

XML Schema Supports Data Types.

**How to use XSD?**

XML documents can have a reference to a DTD or to an XML Schema. Let’s have a simple XML file note.xml

**note.xml**

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

1. **A DTD file**.

The following example is a DTD file called "note.dtd" that defines the elements of the XML document above ("note.xml"):

1. <!ELEMENT note (to, from, heading, body)>  
2. <!ELEMENT to (#PCDATA)>  
3. <!ELEMENT from (#PCDATA)>  
4. <!ELEMENT heading (#PCDATA)>  
5. <!ELEMENT body (#PCDATA)> The first line defines the note element to have four child elements: "to, from, heading, body". Line 2-5 defines the to, from, heading, and body elements to be of type "#PCDATA".

A reference to a DTD.

<?xml version="1.0"?>  
  
<!DOCTYPE note SYSTEM “note.dtd”>  
  
<note>  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>

1. An XML Schema

The following example is an XML Schema file called "note.xsd" that defines the elements of the XML document above ("note.xml"):

<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="https://www.w3schools.com"  
xmlns="https://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>

The note element is a **complex type** because it contains other elements. The other elements (to, from, heading, body) are **simple types** because they do not contain other elements. You will learn more about simple and complex types in the following chapters.

A reference to an XML Schema

<?xml version="1.0"?>  
  
<note  
xmlns="https://www.w3schools.com"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="https://www.w3schools.com/xml note.xsd">  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>

XSD without target namespace:

<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="plant">

<!-- XSD validation rules go inside this xs:elemet tag -->

<xs:complexType>

<!-- Elements inside the sequence must fallow the specified order-->

<xs:sequence>

<xs:element name="genus">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:maxLength value="60"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<!-- minOccurs is to mention "species is optional" -->

<xs:element name="species" type="xs:string" minOccurs="0" />

<xs:element name="cultivar" type="xs:string" minOccurs="0"/>

<xs:element name="commanName" type="xs:string" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

XML depending on above XSD which is having no target name space.

<?xml version="1.0" encoding="UTF-8"?>

<plant xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="plant.xsd">

<genus>ABCD</genus>

<!-- <species>xyz</species> -->

<cultivar>sdf</cultivar>

<commanName>asdfer</commanName>

</plant>

**noNamespaceSchemaLocation**

The **noNamespaceSchemaLocation** attribute references **an XML Schema document that does not have a target namespace.**

<xsi:noNamespaceSchemaLocation="anyURI" >

XSD with target namespace:

<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

**targetNamespace**="https://www.skg.com" **elementFormDefault**="qualified">

<xs:element name="plant">

<!-- XSD validation rules go inside this xs:elemet tag -->

<xs:complexType>

<!-- Elements inside the sequence must fallow the specified order-->

<xs:sequence>

<xs:element name="genus">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:maxLength value="60"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<!-- minOccurs is to mention "species is optional" -->

<xs:element name="species" type="xs:string" minOccurs="0" />

<xs:element name="cultivar" type="xs:string" minOccurs="0"/>

<xs:element name="commanName" type="xs:string" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**targetNamespace**

The value of targetNamespace is simply a unique identifier, typically a company may use their URL followed by something descriptive to qualify it. Placing the targetNamespace attribute at the top of your XSD schema means that all entities defined in it are part of this namespace.

**elementFormDefault**

By specifying elementFormDefault="qualified" we enforce namespace declaration to be used in documents validated with this schema.

<?xml version="1.0" encoding="UTF-8"?>

<skg:plant xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:skg="https://www.skg.com">

<skg:genus>ABCD</skg:genus>

<!-- <species>xyz</species> -->

<skg:cultivar>sdf</skg:cultivar>

<skg:commanName>asdfer</skg:commanName>

</skg:plant>