

XML eXtensible Markup Language

XML is often used for distributing data over the Internet.

XML DTD: XML Document Type Definition

Aim: independent groups of people can agree on a standard DTD for interchanging data.

Structure of XML DTD:

Elements: **main building blocks** <message>some text</message>

Attributes: **extra information about elements.**

PCDATA: **is text that WILL be parsed by a parser**

XML DTD

DTD example:

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
```

!DOCTYPE note defines that the root element of this document is note

!ELEMENT note defines that the note element must contain four elements:
"to,from,heading,body"

!ELEMENT to defines the to element to be of type "#PCDATA"

!ELEMENT from defines the from element to be of type "#PCDATA"

!ELEMENT heading defines the heading element to be of type "#PCDATA"

!ELEMENT body defines the body element to be of type "#PCDATA"

XML DTD

DTD example:

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

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

Validate the XML DTD using Notepad++ or <http://xmlvalidator.new-studio.org/>

XML DTD

In a DTD, attributes are declared with an ATTLIST declaration.

The **attribute-value** can be one of the following:

#REQUIRED

The attribute is required

#IMPLIED

The attribute is optional

#FIXED *value*

The attribute value is fixed

<!ATTLIST element-name attribute-name attribute-type #REQUIRED>

Example:

<!ELEMENT to (#PCDATA)>

<!ATTLIST to type CDATA #REQUIRED >

Valid XML:

<to type="admin" />

Invalid XML:

<to />

XML DTD

In a DTD, attributes are declared with an ATTLIST declaration.

The **attribute-value** can be one of the following:

#REQUIRED

The attribute is required

#IMPLIED

The attribute is optional

#FIXED *value*

The attribute value is fixed

<!ATTLIST element-name attribute-name attribute-type #IMPLIED>

Example:

<!ELEMENT from (#PCDATA)>

<!ATTLIST from location CDATA #IMPLIED >

Valid XML:

<from location="NSW" />

Valid XML:

<from />

XML DTD

In a DTD, attributes are declared with an ATTLIST declaration.

The **attribute-value** can be one of the following:

#REQUIRED

The attribute is required

#IMPLIED

The attribute is optional

#FIXED *value*

The attribute value is fixed

<!ATTLIST element-name attribute-name attribute-type #FIXED>

Example:

<!ELEMENT from (#PCDATA)>

<!ATTLIST from location CDATA #FIXED "UTS" >

Valid XML:

<from company="UTS" />

Invalid XML:

<from company="Google" />

XML Schema

An XML Schema describes the structure of an XML document.

The <schema> element is the root element of every XML Schema.

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

```
</xs:schema>
```

<schema> element and may contain some attributes such as xmlns:xs

XML Schema

A complex element is an XML element that contains other elements and/or attributes.

File is saved as **.xsd**

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

```
<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>
```


XML Schema

xs:string, xs:decimal, xs:integer, xs:Boolean, xs:date, xs:time

XSD attributes:

```
<xs:element name="to">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute type="xs:string" name="type" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

XML Schema

Required:

```
<xs:attribute type="xs:string" name="type" use="required"/>
```

Optional:

```
<xs:attribute type="xs:string" name="type" />
```

Fixed:

```
<xs:attribute type="xs:string" name="type" fixed="0000" />
```