

UNIT IV XML DATABASES 9

Structured, Semi structured, and Unstructured Data – XML Hierarchical Data Model – XML Documents – Document Type Definition – XML Schema – XML Documents and Databases – XML Querying – XPath – XQuery

XML DTD:

DTD stands for Document Type Definition. It defines the legal building blocks of an XML document. It is used to define document structure with a list of legal elements and attributes.

Purpose of DTD:

Its main purpose is to define the structure of an XML document. It contains a list of legal elements with the help of them.

Example:

```
<?xml version="1.0"?>
  <!DOCTYPE employee SYSTEM "employee.dtd">
  <employee>
    <firstname>vimal</firstname>
    <lastname>jaiswal</lastname>
    <email>vimal@javatpoint.com</email>
  </employee>
```

Description of DTD:

- **<!DOCTYPE employee :** It defines that the root element of the document is employee.
- **<!ELEMENT employee:** It defines that the employee element contains 3 elements "firstname, lastname and email".
- **<!ELEMENT firstname:** It defines that the firstname element is #PCDATA typed. (parse-able data type).
- **<!ELEMENT lastname:** It defines that the lastname element is #PCDATA typed. (parse-able data type).
- **<!ELEMENT email:** It defines that the email element is #PCDATA typed. (parse-able data type).

XML DTD

A DTD defines the legal elements of an XML document. In simple words we can say that a DTD defines the document structure with a list of legal elements and attributes. XML schema is a XML based alternative to DTD. Actually DTD and XML schema both are used to form a well formed XML document. We should avoid errors in XML documents because they will stop the XML programs.

XML schema

It is defined as an XML language. It uses namespaces to allow for reuses of existing definitions. It supports a large number of built in data types and definition of derived data types

Valid and well-formed XML document with External DTD

Let's take an example of well-formed and valid XML document. It follows all the rules of DTD.

employee.xml

```
<?xml version="1.0"?>
  <!DOCTYPE employee SYSTEM "employee.dtd">
  <employee>
    <firstname>vimal</firstname>
    <lastname>jaiswal</lastname>
    <email>vimal@javatpoint.com</email>
  </employee>
```

In the above example, the DOCTYPE declaration refers to an external DTD file. The content of the file is shown in below paragraph.

employee.dtd

```
<!ELEMENT employee (firstname,lastname,email)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

Valid and well-formed XML document with Internal DTD

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes" ?>
<!DOCTYPE address [
<!ELEMENT address (name,company,phone)>
<!ELEMENT name (#PCDATA)>
```

```
<!ELEMENT company (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
]>
<address>
<name>Tanmay Patil</name>
<company>TutorialsPoint</company>
<phone>(011) 123-4567</phone>
</address>
```

Description of DTD

<!DOCTYPE employee : It defines that the root element of the document is employee.

<!ELEMENT employee: It defines that the employee element contains 3 elements "firstname, lastname and email".

<!ELEMENT firstname: It defines that the firstname element is #PCDATA typed. (parse-able data type).

<!ELEMENT lastname: It defines that the lastname element is #PCDATA typed. (parse-able data type).

<!ELEMENT email: It defines that the email element is #PCDATA typed. (parse-able data type).

XML CSS

Purpose of CSS in XML

CSS (Cascading Style Sheets) can be used to add style and display information to an XML document. It can format the whole XML document.

How to link XML file with CSS

To link XML files with CSS, you should use the following syntax:

```
<?xml-stylesheet type="text/css" href="cssemployee.css"?>
```

XML CSS Example

cssemployee.css

```
employee
{
    background-color: pink;
}
firstname,lastname,email
{
    font-size:25px;
```

```
display:block;
color: blue;
margin-left: 50px;
}
```

employee.dtd

```
<!ELEMENT employee (firstname,lastname,email)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

employee.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="cssemployee.css"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
<firstname>vimal</firstname>
<lastname>jaiswal</lastname>
<email>vimal@javatpoint.com</email>
</employee>
```

CDATA vs PCDATA

CDATA

CDATA: (Unparsed Character data): CDATA contains the text which is not parsed further in an XML document. Tags inside the CDATA text are not treated as markup and entities will not be expanded.

Let's take an example for CDATA:

```
<?xml version="1.0"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
<![CDATA[
<firstname>vimal</firstname>
<lastname>jaiswal</lastname>
<email>vimal@javatpoint.com</email>
```

```
]]>
</employee>
```

In the above CDATA example, CDATA is used just after the element employee to make the data/text unparsed, so it will give the value of employee:

```
<firstname>vimal</firstname>
<lastname>jaiswal</lastname>
<email>vimal@javatpoint.com</email>
```

PCDATA

PCDATA: (Parsed Character Data): XML parsers are used to parse all the text in an XML document. PCDATA stands for Parsed Character data. PCDATA is the text that will be parsed by a parser. Tags inside the PCDATA will be treated as markup and entities will be expanded. In other words you can say that a parsed character data means the XML parser examines the data and ensures that it doesn't contain an entity if it contains that will be replaced.

Let's take an example:

```
<?xml version="1.0"?>
  <!DOCTYPE employee SYSTEM "employee.dtd">
  <employee>
    <firstname>vimal</firstname>
    <lastname>jaiswal</lastname>
    <email>vimal@javatpoint.com</email>
  </employee>
```

In the above example, the employee element contains 3 more elements 'firstname', 'lastname', and 'email', so it parses further to get the data/text of firstname, lastname and email to give the value of employee as:

```
vimaljaiswal vimal@javatpoint.com
```
