DTD - ATTRIBUTES

In this chapter we will discuss about DTD Attributes. Attribute gives more information about an element or more precisely it defines a property of an element. An XML attribute is always in the form of a name-value pair. An element can have any number of unique attributes.

Attribute declaration is very much similar to element declarations in many ways except one; instead of declaring allowable content for elements, you declare a list of allowable attributes for each element. These lists are called ATTLIST declaration.

Syntax

Basic syntax of DTD attributes declaration is as follows:

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

In the above syntax

- The DTD attributes start with <!ATTLIST keyword if the element contains the attribute.
- **element-name** specifies the name of the element to which the attribute applies.
- **attribute-name** specifies the name of the attribute which is included with the element-name.
- **attribute-type** defines the type of attributes. We will discuss more on this in the following sections.
- **attribute-value** takes a fixed value that the attributes must define. We will discuss more on this in the following sections.

Example

Below is a simple example for attribute declaration in DTD:

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA #REQUIRED>
 ]>
<address>
   <name >Tanmay Patil</name>
</address>
```

Let us go through the above code:

- Begin with the XML declaration with the following statement:

  ```
  <?xml version = "1.0"?>
  ```

- Immediately following the XML header is the document type declaration, commonly referred to as the DOCTYPE:

  ```
  <!DOCTYPE address [
  ```

  The DOCTYPE informs the parser that a DTD is associated with this XML document. The DOCTYPE declaration has an exclamation mark ! at the start of the element name.

- Following is the body of DTD. Here we have declared element and attribute:

```
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
```

- Attribute *id* for the element *name* is defined as:

```
<!ATTLIST name id CDATA #REQUIRED>
```

Here attribute type is *CDATA* and its value is *#REQUIRED*.

Rules of Attribute Declaration

- All attributes used in an XML document must be declared in the Document Type Definition *DTD* using an Attribute-List Declaration
- Attributes may only appear in start or empty tags.
- The keyword ATTLIST must be in upper case
- No duplicate attribute names will be allowed within the attribute list for a given element.

Attribute Types

When declaring attributes, you can specify how the processor should handle the data that appears in the value. We can categorize attribute types in three main categories:

- String type
- Tokenized types
- Enumerated types

Following table provides a summary of the different attribute types:

Type	Description
CDATA	CDATA is character data *textandnotmarkup*. It is a *String Attribute Type*.
ID	It is a unique identifier of the attribute. It should not appear more than once. It is a *Tokenized Attribute Type*.
IDREF	It is used to reference an ID of another element. It is used to establish connections between elements. It is a *Tokenized Attribute Type*.
IDREFS	It is used to reference multiple ID's. It is a *Tokenized Attribute Type*.
ENTITY	It represents an external entity in the document. It is a *Tokenized Attribute Type*.
ENTITIES	It represents a list of external entities in the document. It is a *Tokenized Attribute Type*.
NMTOKEN	It is similar to CDATA and the attribute value consists of a valid XML name. It is a *Tokenized Attribute Type*.
NMTOKENS	It is similar to CDATA and the attribute value consists a list of valid XML name. It is a *Tokenized Attribute Type*.
NOTATION	An element will be referenced to a notation declared in the DTD document. It is an *Enumerated Attribute Type*.
Enumeration	It allows defining a specific list of values where one of the values must match. It is an *Enumerated Attribute Type*.

Attribute Value Declaration

Within each attribute declaration, you must specify how the value will appear in the document. You can specify if an attribute:

- can have a default value
- can have a fixed value
- is required
- is implied

Default Values

It contains the default value. The values can be enclosed in single quotes' or double quotes "

Syntax

Following is the syntax of value:

```
<!ATTLIST element-name attribute-name attribute-type "default-value">
```

where *default-value* is the attribute value defined.

Example

Following is a simple example of attribute declaration with default value:

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA "0">
 ]>
<address>
   <name >
      Tanmay Patil
   </name>
</address>
```

In this example we have *name* element with attribute *id* whose default value is *0*. The default value is been enclosed within the double quotes.

FIXED Values

#FIXED keyword followed by the fixed value is used when you want to specify that the attribute value is constant and cannot be changed. A common use of fixed attributes is specifying version numbers.

Syntax

Following is the syntax of fixed values:

```
<!ATTLIST element-name attribute-name attribute-type #FIXED "value" >
```

where #FIXED is an attribute value defined.

Example

Following is a simple example of attribute declaration with FIXED value:

```
<?xml version="1.0"?>
<!DOCTYPE address [
  <!ELEMENT address (company)*>
  <!ELEMENT company (#PCDATA)>
  <!ATTLIST company name NMTOKEN #FIXED "tutorialspoint">
```

```
]>
<address>
  <company name="tutorialspoint">we are a free online teaching faculty</company>
</address>
```

In this example we have used the keyword #FIXED where it indicates that the value "tutorialspoint" is the only value for the attribute *name* of element <company>. If we try to change the attribute value then it gives an error.

Following is an invalid DTD:

```
<?xml version="1.0"?>
<!DOCTYPE address [
  <!ELEMENT address (company)*>
  <!ELEMENT company (#PCDATA)>
  <!ATTLIST company name NMTOKEN #FIXED "tutorialspoint">
]>
<address>
  <company name="abc">we are a free online teaching faculty</company>
</address>
```

REQUIRED values

Whenever you want specify that an attribute is required, use #REQUIRED keyword.

Syntax

Following is the syntax of #REQUIRED:

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

where #REQUIRED is an attribute type defined.

Example

Following is a simple example of DTD attribute declaration with #REQUIRED keyword:

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA #REQUIRED>
 ]>
<address>
   <name >
      Tanmay Patil
   </name>
</address>
```

In this example we have used #REQUIRED keyword to specify that the attribute *id* must be provided for the element-name *name*

IMPLIED Values

When declaring attributes you must always specify a value declaration. If the attribute you are declaring has no default value, has no fixed value, and is not required, then you must declare that the attribute as *implied*. Keyword #IMPLIED is used to specify an attribute as *implied*.

Syntax

Following is the syntax of #IMPLIED:

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

where #IMPLIED is an attribute type defined.

Example

Following is a simple example of #IMPLIED

```
<?xml version = "1.0"?>
<!DOCTYPE address [
<!ELEMENT address ( name )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name id CDATA #IMPLIED>
 ]>
<address>
   <name />
</address>
```

In this example we have used the keyword #IMPLIED as we do not want to specify any attributes to be included in element *name*. It is optional.