

### XML Entity

[almost same as symbolic constants in C]

**An entity is a symbolic representation of information.** With symbolic representation of information, a lot of text, such as, "Hello my name is Tizag.com and I am an artificial intelligence that teaches the general public how to program in web languages for free" can be represented by an entity symbol.

### Creating an XML Entity

An entity must be created in the Document Type Definition (DTD). When you know where to place the entity, the rest is easy. Here is the syntax for creating your own XML entities.

```
<!ENTITY entityName "The text you want to appear when the entity is used">
```

Below we have created an entity for the default introduction we want to include on all of our documents.

```
<!ENTITY intro "Hello my name is Tizag.com and I am an artificial intelligence that teaches the general public how to program in web languages for free">
```

### Using Your Entity

After the entity has been created in the DTD it can then be referenced. An example email XML document that uses such an entity would look like:

```
<!ENTITY intro "Hello my name is Tizag.com and I am an artificial intelligence that teaches the general public how to program in web languages for free">
```

```
<email>
```

```
  <to>A. Nony Mouse</to>
```

```
  <body>&intro;</body>
```

```
</email>
```

### Entity Syntax

You may have used entities in the HTML. The format of an entity in XML is an ampersand(&), followed by the name of the symbol, and concluded with a semicolon.

Generic Entity - &name;

Entities are divided into two group built in entity and user defined entity

HTML is another markup language that supports entities. Below are some example entities for built in entities and the information they represent.

&lt; = <

&gt; = >

&amp; = &

**&quot; = "**

Entities are great for many situations. Especially if you...

- use something a lot. If you have a default introduction, signature, or something else that is commonly used, you should use an entity.
- change something often. If you have a relatively static document that has one or two pieces of information, that are used throughout the document, that change frequently replace them with entities. You only need to change the value of the entity to change hundreds or maybe even thousands of references that are in your XML document.
- are using complex ascii characters that don't occur on your keyboard: © and ® are easy when you use entities.

**Comments in XML**

A comment is used to leave a note or to temporarily edit out a portion of XML code. Although XML is supposed to be self-describing data, you may still come across some instances where an XML comment might be necessary. For many different reasons, sometimes you might want to temporarily remove some XML code from your XML document. XML Comments let you do this easily, as this example below shows.

XML Code:

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<!-- Students grades are updated bi-monthly -->
<class_list>
  <student>
    <name>Robert</name>
    <grade>A+</grade>
  </student>
<!--
  <student>
    <name>Lenard</name>
    <grade>A-</grade>
  </student>
-->
</class_list>
```

In this xml file contains only one students details

**CDATA**

The parser will parse all text in an XML document. The parser will ignore only text inside a CDATA section.

**Parsed Data**

When an XML element is parsed, the text between the XML tags is also parsed:

```
<message>This text is also parsed</message>
```

The parser does this because XML elements can contain other elements, as in this example, where the <name> element contains two other elements (first and last):

```
<name><first>Bill</first><last>Gates</last></name>
```

and the parser will break it up into sub-elements like this:

```
<name>
  <first>Bill</first>
  <last>Gates</last>
</name>
```

**Illegal XML characters have to be replaced by entity references.**

If you place a character like "<" inside an XML element, it will generate an error because the parser interprets it as the start of a new element. You cannot write something like this:

```
<message>if salary < 1000 then</message>
```

To avoid this, you have to replace the "<" character with an entity reference, like this:

```
<message>if salary &lt; 1000 then</message>
```

There are 5 predefined entity references in XML:

&lt;	<	less than
&gt;	>	greater than

&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

**Note:** Only the characters "<" and "&" are strictly illegal in XML. Apostrophes, quotation marks and greater than signs are legal, but it is a good habit to replace them.

### CDATA

**Everything inside a CDATA section is ignored by the parser.**

If your text contains a lot of "<" or "&" characters - as program code often does - the XML element can be defined as a CDATA section. The less than sign (<) would automatically interpreted as being the start of an XML tag. With the CDATA section in place, this won't happen.

A CDATA section starts with "<![CDATA[" and ends with "]]>":

```
<script>
<![CDATA[
function matchwo(a,b)
{
if (a < b && a < 0) then
  { return 1 }
else
  { return 0 }
}
]]>
</script>
```

In the example above, everything inside the CDATA section is ignored by the parser.

Notes on CDATA sections:

A CDATA section cannot contain the string "]]>", therefore, nested CDATA sections are not allowed.

Also make sure there are no spaces or line breaks inside the "]]>" string.

If you wanted to save the following HTML code within an XML block, you would find that parser complains that the img tag is not closed.

```
<skill>
  <name> visual basis</name>
  <htmlText>
    <div>
      
    </div>
  </htmlText>
</skill>
```

The HTML could be safely protected in the preceding sample by enclosing in a CDATA section, as the following example demonstrates:

```
<skill>
  <name> visual basis</name>
  <htmlText><![CDATA[
    <div>
      
    </div>]]>
  </htmlText>
</skill>
```

CDATA sections preserve white space as well as XML special characters, so using CDATA makes sense when you want to be able to save script text and HTML text within an XML document, as in the above example. Because contain large number of symbols like <, > and ”.

\*\*\*Comments is just like removing part of file from the XML document. Portion inside the CDATA section in the XML document is part of the document\*\*\*\*

## XML

- 1) what is entity? Write how the entity help the programmers to create XML files
- 2) Explain the syntax for creating entity and using entity
- 3) Write the difference between CDATA and comment
- 4) What are the advantages of using CDATA in XML document
- 5) Write the importance of comment tag.(<!-- and -->)