

XML

(eXtensible Markup Language)

Introduction:

- XML stands for eXtensible Markup Language and it is used for storing and transferring data.
- XML doesn't depend on the platform and the software (programming language). i.e. we can able to write a program in any language on any platform (Operating System) to send, receive or store data using XML.
- XML is a simple document with the data, which can be used to store and transfer data between any systems irrespective of their hardware and software compatibilities.
- It is so much easier to read the data from XML and display it on a GUI (graphical user interface) using markup language like HTML.

HTML Vs XML

HTML(Hyper Text Markup Language)	XML(eXtensible Markup Language)
HTML is not a case sensitive language	XML is case sensitive language
HTML is mainly concerned with the presentation of data	XML is mainly used for storing and transporting the data
HTML is static	XML is dynamic
In HTML the closing tag is optional	In XML the closing tag is mandatory
HTML uses predefined tags such as , , etc.	XML uses the user-defined tags that we create while writing the XML document.
HTML does not preserve white space.	XML preserves white space.
HTML Attribute Values Quoting is optional.	XML Attribute Values Must Always be Quoted.

Simple XML document example

A XML document structure looks like this:

<root>

<child>

<subchild>....</subchild>

</child>

<child>

<subchild>....</subchild>

</child>

</root>

- **<?xml version="1.0" encoding="UTF-8"?>** is called XML Prolog.
- It is optional, however when we include it in the XML document, it should always be the first line of the document. XML Prolog defines the XML version and the encoding used in the XML document.

- The XML code can be written on a simple notepad and should be saved as “filename.xml”.

Students.xml

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

```
<students>
```

```
  <student>
```

```
    <num>527</num>
```

```
    <name>Ravi</name>
```

```
    <age>20</age>
```

```
  </student>
```

```
  <student>
```

```
    <num>572</num>
```

```
    <name>Dev</name>
```

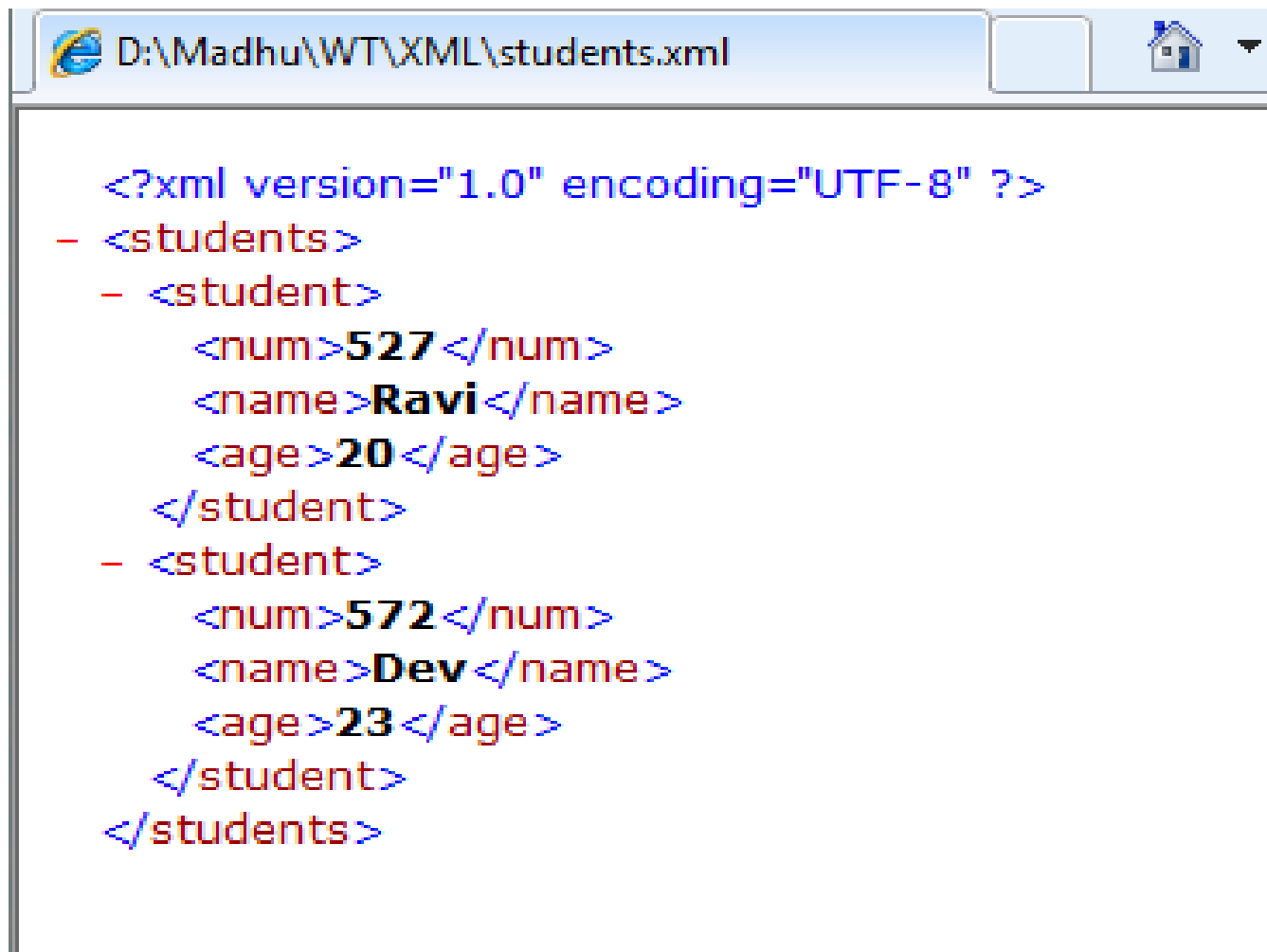
```
    <age>23</age>
```

```
  </student>
```

```
</students>
```

- In the above XML document we have the details of the few students. Here <students> is the root element, <student> is the child element and num,name and age are sub-child elements.

Output:

A screenshot of a text editor window. The title bar shows the file path 'D:\Madhu\WT\XML\students.xml'. The editor contains XML code with two student entries. The first entry has num=527, name=Ravi, and age=20. The second entry has num=572, name=Dev, and age=23. The code is color-coded: opening and closing tags are blue, attribute names are red, and values are black.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <students>
- <student>
    <num>527</num>
    <name>Ravi</name>
    <age>20</age>
  </student>
- <student>
    <num>572</num>
    <name>Dev</name>
    <age>23</age>
  </student>
</students>
```

XML Syntax

Root Element is mandatory in XML

- XML document must have a root element. A root element can have child elements and sub-child elements.
- For example: In the following XML document, <message> is the root element and <to>, <from>, <subject> and <text> are child elements.

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

```
<message>
```

```
  <to>Durga</to>
```

```
  <from>Madhu</from>
```

```
  <subject>Message from teacher to Student</subject>
```

```
  <text>You have an exam tomorrow at 10:00 AM</text>
```

```
</message>
```

XML is case sensitive

XML is a case sensitive language.

For example:

`<from>madhu</from>` → This is valid

`<from>Prasad</FROM>` → This is invalid

- All letters of closing tag is in capital while all letters of opening tag is in small, this is an example of invalid XML.

XML Prolog

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

- This line is called the XML Prolog. It is an optional line, however it should be the first line when you mention it. It specifies the XML version and the encoding used in the XML document.

XML Syntax

Contd...

Elements should not overlap

- All the elements in XML should be properly nested and they should not overlap.

`<class><teacher>Madhu</class></teacher>` --> Wrong (Not nested properly)

`<class><teacher>Durga</teacher></class>` --> Correct (Correctly nested)

XML elements must have a closing tag

- All XML documents must have a closing tag.

`<text category = message>hello</text>` --> correct

`<text category = message>hello` --> wrong

Comments in XML

- This is how a comment should look like in XML document.

`<!-- This is just a comment -->`

XML Attributes

- XML elements can have attributes. By the use of attributes we can add the additional information about the element.
- XML attributes are a way to add additional data to the XML element. Attributes contain data in form of name & value pairs.
- XML attributes are used to enhance the properties of the elements.

Note: XML attributes must always be quoted. We can use single or double quote.

Example:

```
<book category="computers">
```

```
<price>1000rs</price>
```

```
<publisher>Tata McGraw Hill</publisher>
```

```
</book>
```

XML attributes vs XML Sub Elements:

- The Data can be stored in attributes or in child elements. But there are some limitations in using attributes, over child elements.

Same information can be represented in two ways:

1st Way:

```
<book category="computers">  
<price>1000rs</price>  
<publisher>Tata McGraw Hill</publisher>  
</book>
```

2nd Way:

```
<book>  
<category>computers </category>  
<price>1000rs</price>  
<publisher>Tata McGraw Hill</publisher>  
</book>
```

XML Attributes

Contd...

- In the first way category is used as an attribute and in the second way category is used as an element.
- Both examples provide the same information but it is good practice to avoid attribute in XML and use elements instead of attributes. Because
- Attributes cannot contain multiple values but child elements can have multiple values.
- Attributes cannot contain tree structure but child element can.
- Elements are easy to be handled by the programming language compared to the attributes.