XML

XML stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data.

There are three important characteristics of XML that make it useful in a variety of systems and solutions:

- **XML is extensible**: XML allows you to create your own self-descriptive tags, or language, that suits your application.
- **XML carries the data, does not present it:** XML allows you to store the data irrespective of how it will be presented.
- **XML is a public standard:** XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML Usage

A short list of XML usage says it all:

- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to exchange the information between organizations and systems.
- XML can be used for offloading and reloading of databases.
- XML can be used to store and arrange the data, which can customize your data handling needs.
- XML can easily be merged with style sheets to create almost any desired output.
- Virtually, any type of data can be expressed as an XML document.

XML Declaration

The XML document can optionally have an XML declaration. It is written as below:
<?xml version="1.0" encoding="UTF-8"?>
Where version is the XML version and encoding specifies the character encoding used in the document here UTF-8 stands for one byte encoding for English letters and symbol.

Syntax Rules for XML declaration

- The XML declaration is case sensitive and must begin with "<?xml>" where "xml" is written in lower-case.
- If document contains XML declaration, then it strictly needs to be the first statement of the XML document.
- The XML declaration strictly needs be the first statement in the XML document.
- An HTTP protocol can override the value of encoding that you put in the XML declaration.

Tags and Elements

An XML file is structured by several XML-elements, also called XML-nodes or XML-tags. XML-elements' names are enclosed by triangular brackets < > as shown below:
<element>

Syntax Rules for Tags and Elements

Element Syntax: Each XML-element needs to be closed either with start or with end elements as shown below:

<element>....</element>

or in simple-cases, just this way:

<element/>

Nesting of elements: An XML-element can contain multiple XML-elements as its children, but the children elements must not overlap. i.e., an end tag of an element must have the same name as that of the most recent unmatched start tag.
Following example shows incorrect nested tags:

<?xml version="1.0"?>
<contact-info>
<company>Tutorials
<contact-info>
</company>

Following example shows correct nested tags:
<?xml version="1.0"?>
<contact-info>
<company>Tutorials</company>
<contact-info>

Root element: An XML document can have only one root element. For example, following is not a correct XML document, because both the x and y elements occur at the top level without a root element:
<x>...</x>
<y>...</y>
The following example shows a correctly formed XML document:
<root>
<x>...</x>
<y>...</y>
</root>
Case sensitivity: The names of XML-elements are case-sensitive. That means the name of the start and the end elements need to be exactly in the same case.
For example, <contact-info> is different from <Contact-Info>.

Attributes

An attribute specifies a single property for the element, using a name/value pair. An XML-element can have one or more attributes. For example:
<IMG src="/images/lu.png">University of Lucknow
Here, img is the attribute name and /images/lu.png is attribute value.

Syntax Rules for XML Attributes

- Attribute names in XML (unlike HTML) are case sensitive. That is, IMG and img are considered two different XML attributes.
- Same attribute cannot have two values in a syntax. The following example shows incorrect syntax because the attribute b is specified twice:
  ....

Example1

Code:

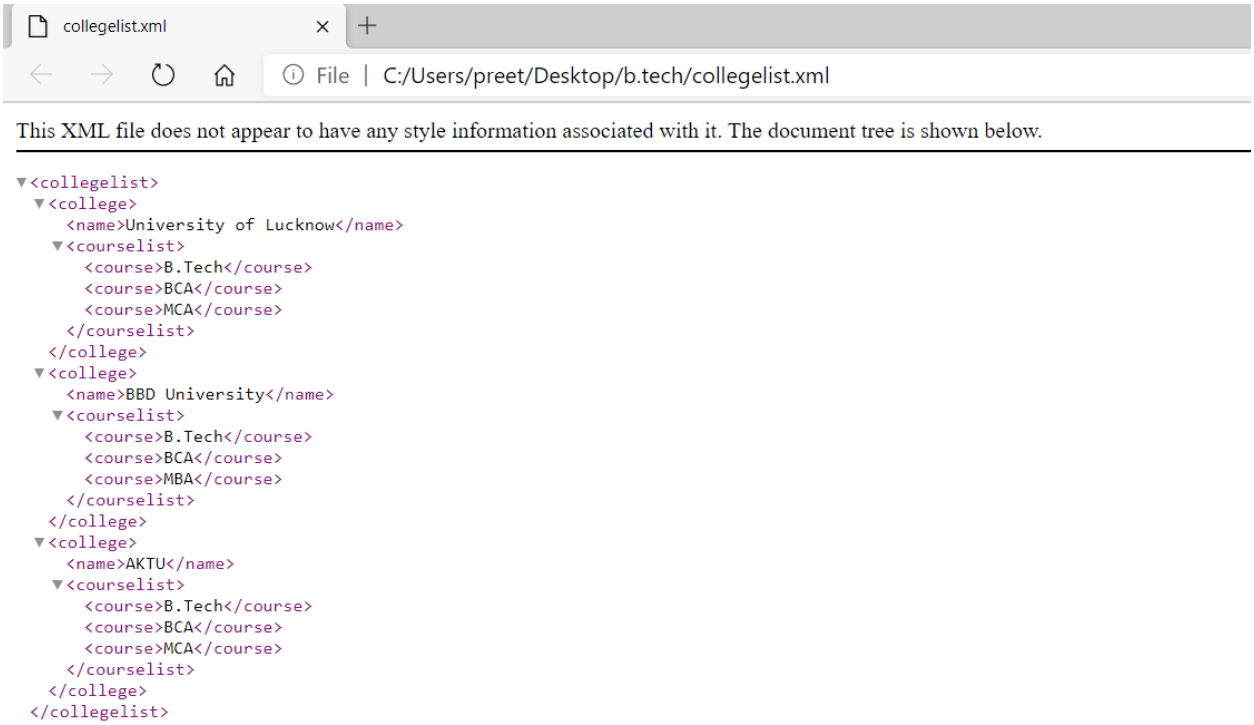
```
<?xml version="1.0" encoding="UTF-8"?>
<collegelist>
 <college>
  <name>University of Lucknow</name>
  <courselist>
   <course>B.Tech</course>
   <course>BCA</course>
   <course>MCA</course>
  </courselist>

 </college>
 <college>
  <name>BBD University</name>
  <courselist>
   <course>B.Tech</course>
   <course>BCA</course>
   <course>MBA</course>
  </courselist>

 </college>
 <college>
  <name>AKTU</name>
  <courselist>
   <course>B.Tech</course>
   <course>BCA</course>
   <course>MCA</course>
  </courselist>

 </college>
</collegelist>
```

Output:



XML can easily be merged with style sheets to create almost any desired

XML Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="C:\Users\preet\Desktop\b.tech\xml-css.css"?>
<collegelist>
 <college>
  <name>University of Lucknow</name>
  <courselist>
   <course>B.Tech</course>
   <course>BCA</course>
   <course>MCA</course>
```

```
   </courselist>

  </college>
  <college>
   <name>BBD University</name>
   <courselist>
     <course>B.Tech</course>
     <course>BCA</course>
     <course>MBA</course>
   </courselist>

  </college>
  <college>
   <name>AKTU</name>
   <courselist>
     <course>B.Tech</course>
     <course>BCA</course>
     <course>MCA</course>
   </courselist>

  </college>
</collegelist>
```

CSS-code:

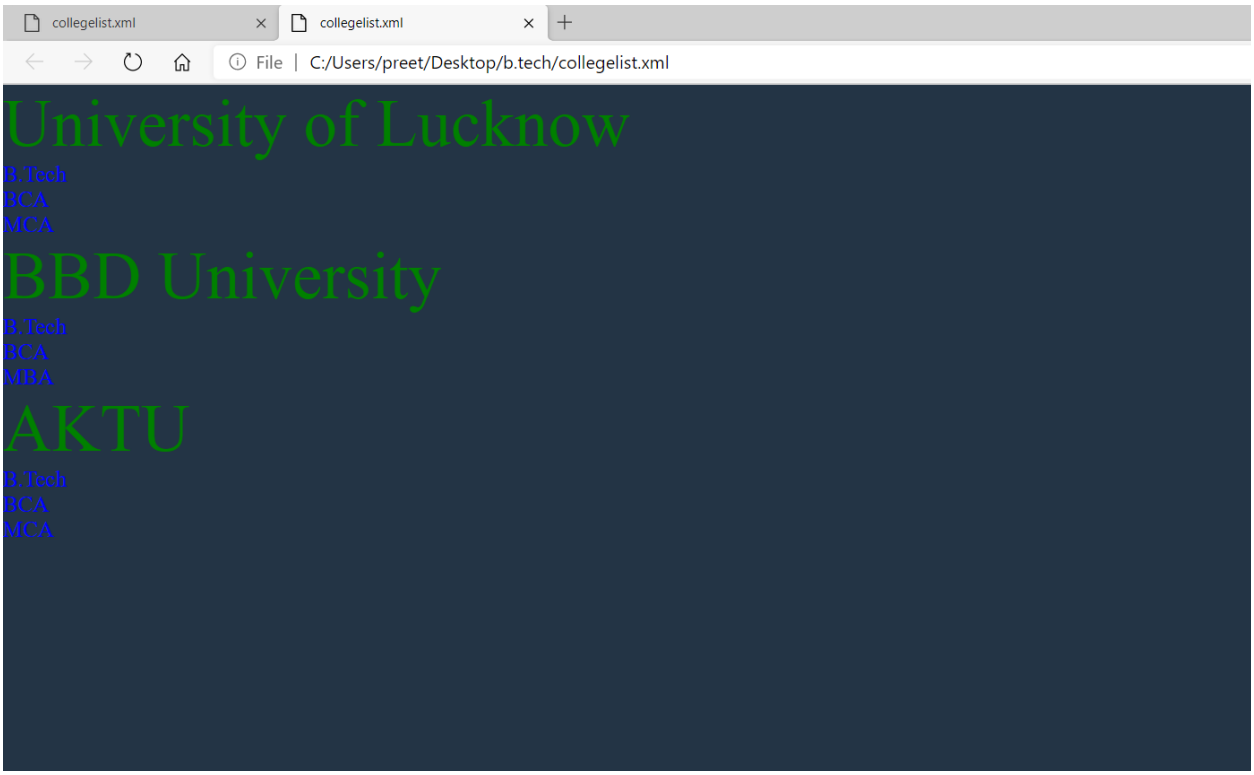
```
collegelist {
   background-color: #233445;
   width: 100%;
}
```

```
 college{
display: block  ;
color: yellow




}
name {
   display: block  ;
   color: green;
   font-size: 60px;
}
course{
display: block  ;
color: blue;
   font-size: 20px;
}
```

Output:


collegelist.xml
collegelist.xml
File | C:/Users/preet/Desktop/b.tech/collegelist.xml
University of Lucknow
B.Tech
BCA
MCA
BBD University
B.Tech
BCA
MBA
AKTU
B.Tech
BCA
MCA