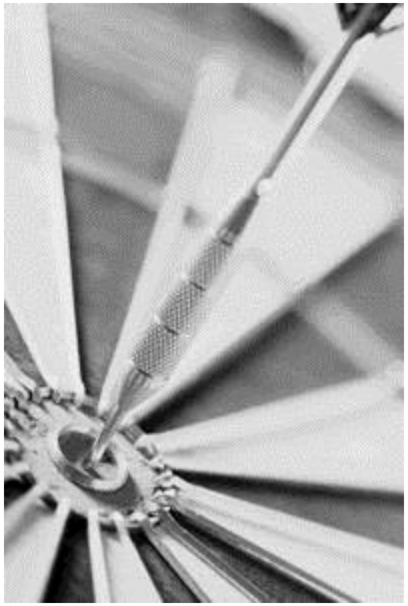
Objectives

- Upon completing this module, you will be able to:
  - Explain the concept and use of XML
  - Describe the XML tree structure
  - Identify and adhere to XML syntax rules
  - Use XML elements and attributes
  - Describe how XML documents are validated
  - Use XML schemas for XML document validation



Agenda

- XML Overview
- XML Tree Structure
- XML Syntax Rules
- XML Elements
- XML Attributes
- Validating XML Documents

XML Overview (1 of 2)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- XML:
  - Is a markup language for documents containing structured information.
  - Is used to describe, store, and transport data.
  - Is pure information wrapped in user-defined tags.
  - Was created so that richly structured documents could be used over the internet.

XML Overview (2 of 2)

- Design Goals for XML:
  - XML shall be straightforwardly usable over the Internet.
  - XML shall support a wide variety of applications.
  - XML shall be compatible with SGML(Standard Generalized Markup Language).
  - It shall be easy to write programs that process XML documents.
  - XML documents should be human-legible and reasonably clear.
  - The XML design should be prepared quickly.
  - The design of XML shall be formal and concise.
  - XML documents shall be easy to create.

XML Tree Structure (1 of 3)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- Elements in XML documents form a logical tree structure.
- XML tree structure illustrates the hierarchy and locality of the elements in a XML document.
- XML tree structure can help in showing which elements are the descendants and ancestors of each element.

XML Tree Structure (2 of 3)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

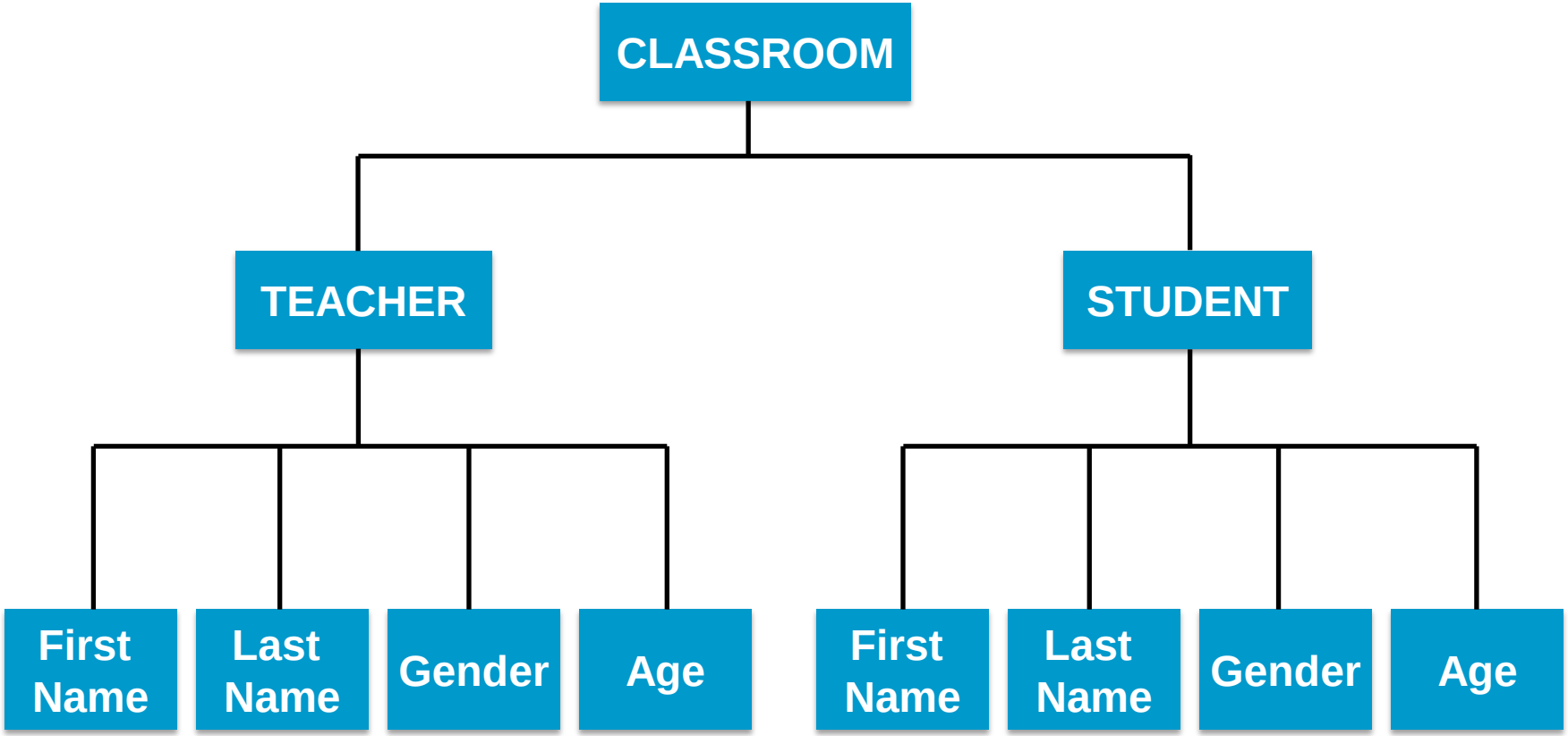
- Sample code:

```
<?xml version="1.0" ?>
<classroom>
<teacher>
     <first_name>Victoria</first_name>
     <last_name>Brooke</last_name>
     <gender>Female</gender>
     <age>30</age>
</teacher>
<student>
     <first_name>Michael</first_name>
     <last_name>Rogers</last_name>
     <gender>Male</gender>
     <age>18</age>
</student>
</classroom>
```

XML Tree Structure (3 of 3)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- XML Tree of previous code:



XML Syntax Rules (1 of 5)

- All XML documents should begin with an XML declaration. The XML declaration is a processing instruction that identifies the document as being XML.

Example :

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

- There are no predefined tags in XML, users have to make their own tags.

- XML documents must have exactly one root element, also known as the document element.

- Basic syntax for XML elements:

Syntax :

```
<element_name>element value</element_name>
```

Examples :

```
<first_name>Jason</first_name>
<book_title>My Favorite Book</book_title>
```

XML Syntax Rules (2 of 5)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- All XML elements must have a corresponding closing tag.

Invalid : `<some_tag>some value….`

Valid : `<some_tag>some value placed here</some_tag>`

- XML tags are case sensitive.

Invalid : `<Item_Name>7 Tonner Rice</item_name>`

Valid : `<item_name>7 Tonner Rice</item_name>`

XML Syntax Rules (3 of 5)

2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f48696768207

- XML elements must be properly nested

Invalid :

```
<parent_element>
        <child_element>Some value here</parent_element>
        <child_element2>Some value here   </child_element>
</child_element2>
```

Valid :

```
<parent_element>
        <child_element>Some value here</child_element>
        <child_element2>Another value here</child_element2>
</parent_element>
```

XML Syntax Rules (4 of 5)

- XML attribute values must be placed within quotes

Invalid :

```
<some_tag attribute1=attributeValue>….</some_tag>
```

Valid :

```
<some_tag attribute1="attributeValue">….</some_tag>
```

- Make use of entity references for special characters

<	<	Less than symbol
>	>	Greater than symbol
&	&	Ampersand
'	‘	Apostrophe
"	“	Quotation mark

XML Syntax Rules (5 of 5)

2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f48696768207

- Example using entity references:

```
<?xml version="1.0" ?>
<quote>
   <author>Sam Ewing </author>
   <paragraph> &quot; Success has a simple formula: do your
       best, and people may like it.  &quot; </paragraph>
    </quote>
```

- The value of <paragraph> when accessed will be:

  “Success has a simple formula: do your best, and people may like

XML Elements

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- Elements are used to classify data in an XML document to make the data understandable.
- Elements can have any name desired and are usually descriptive of the data they hold.
- Elements can contain other elements, usually to include more details.

XML Elements (Cont.)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657

- Elements can contain attributes that also allow additional information.
- Elements are defined by its opening and closing tags.

```
<myElement> This sentence is found inside the element </myElement>
- myElement-
```

XML Attributes

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- XML Attributes provide additional information to the element they belong to.
- Attributes are commonly used for identification purposes, in such cases, there are more than one element of the same type.

XML Attributes (Cont.)

2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f48696768207

- XML Attribute Sample:

```
<?xml version="1.0" ?>
  <class_list  year="3"  section="B">
      <student  id="B-0001">
            <first_name>Anna</first_name>
            <last_name>Sanders</last_name>
            <gender>Female</gender>
      </student>
      <student  id="B-0002">
            <first_name>John</first_name>
            <last_name>dela Cruz</last_name>
            <gender>Male</gender>
      </student>
  </class_list>
```

Validating XML Documents (1 of 17)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- Validating XML documents is done through a DTD (Document Type Definition) or an XSD (XML Schema Definition).
- There are different types of XML documents in terms of validity:

Well-Formed XML Documents	Refer to XML documents that fully comply to the syntax rules.
Valid XML Documents	Refer to XML documents that are well-formed and comply to a DTD/XSD.

Validating XML Documents (2 of 17)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f48696768207

- A DTD defines the structure of an XML document with a list of legal elements and attributes.
- XML Schema Definition (XSD) is the XML-based alternative of DTDs, having the same purpose of DTDs but more powerful and extensible.
- DTD is the older standard. It is most probable that XSD will replace DTD for validating XML Documents.

Validating XML Documents (3 of 17)

2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f48696768207

- XML Schema (XSD):
  - Is extensible to accept future additions.
  - Is more powerful than its predecessor.
  - Is written similarly to XML as XSD makes use of XML syntax, hence most of the rules of XML apply to XSD.
  - Has data types and namespaces, unlike DTDs.
  - Includes the following data types:
    - String
    - Date
    - Numeric
    - Many others

Validating XML Documents (4 of 17)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- The <schema> element is the root element for XSD.

Syntax :

```
<xs:schema>
…
…
…
</xs:schema>
```

- The <schema> element can have attributes, including the default namespace to be used.

Validating XML Documents (5 of 17)

- Attributes of elements are defined within the elements where the attributes belong.

Syntax :

```
<xs:attribute name=“name_of_attribute" type=“data_type"/>
```

Example :

```
<xs:element name=“dog”>
     <xs:complexType>
          <xs:simpleContent>
               <xs:extension base=“xs:string”>
                    <xs:attribute name=“breed" type=“xs:string"/>
               </xs:extension>
          </xs:simpleContent>
     </xs:complexType>
</xs:element>
```

Validating XML Documents (6 of 17)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- Elements in schemas define the structure and properties of elements in XML documents.
- Elements in schemas are divided into two types:
  - Simple Elements
  - Complex Elements

Validating XML Documents (7 of 17)

- Simple elements refer to elements containing “only text” and cannot contain other elements or attributes.

In XML :

```
<first_name>Michael Angelo</first_name>
```

Syntax :

```
<xs:element name=“my_name" type=“the_datatype"/>
```

Example :

```
<xs:element name=“first_name” type=“xs:string”/>
```

Validating XML Documents (8 of 17)

2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070657266f726d616e63652c2044656c6976657265642c2f486967682070

- Simple elements can have default or fixed values.

Syntax:

```
<xs:element name="my_name" type="the_datatype" default="default_value"/>
<xs:element name="my_name" type="the_datatype" fixed="fixed_value"/>
```

Examples:

```
<xs:element name="current_year" type="xs:integer" default="2008"/>
<xs:element name="legal_age" type="xs:integer" fixed="18"/>
```

Validating XML Documents (9 of 17)

- Complex elements refer to elements that can have attributes, such as:
  - Elements that contain only text (with attributes).
  - Elements that are empty.
  - Elements containing other elements.
  - Elements that contain both text and other elements.

Validating XML Documents (10 of 17)

2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f48696768207

- Elements that contain “only text” but have attributes are considered complex elements.

XML:

```
<dog dog_tag_number=“100012”>Spike</dog>
```

XSD:

```
<xs:element name=“dog">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name=“dog_tag_number" type="xs:integer" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Validating XML Documents (11 of 17)

2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f48696768207

- Elements that are empty refer to elements that hold no text but may contain attributes.

XML:

```
<computer serial_ID="103A-0212-00A7-101B" />
```

XSD:

```
<xs:element name="computer">
    <xs:complexType>
        <xs:attribute name="serial_ID" type="xs:string"/>
    </xs:complexType>
</xs:element>
```

Validating XML Documents (12 of 17)

- Elements that serve only to contain other elements are considered complex elements.

XML:

```
<car>
  <color>red</color>
  <wheels>4</wheels>
</car>
```

XSD :

```
<xs:element name=“car">
   <xs:complexType>
         <xs:sequence>
             <xs:element name=“color" type="xs:string"/>
             <xs:element name=“wheels" type="xs:integer"/>
         </xs:sequence>
   </xs:complexType>
</xs:element>
```

Validating XML Documents (13 of 17)

- Elements that contain both text and other elements are considered complex elements

XML:

```
<letter> Dear Mrs.
  <name>Erika Daniels</name>. Your child,
  <child_name>Michael</child_name>, has done something at
            school. Please come to the prinicipal’s office
            anytime tomorrow,
  <schedule>2008-08-21</schedule>.
</letter>
```

XSD :

```
<xs:element name="letter">
     <xs:complexType mixed="true">
            <xs:sequence>
                <xs:element name="name" type="xs:string"/>
                <xs:element name="child_name" type="xs:string"/>
                <xs:element name="schedule" type="xs:date"/>
            </xs:sequence>
   </xs:complexType>
</xs:element>
```

Key Points

- XML is a markup language for documents containing structured information, it is used to describe, store, and transport data.
- XML tree structure illustrates the hierarchy and locality of the elements in a XML document it helps show which elements are the descendants and ancestors of each element.
- All XML documents should begin with an XML declaration. The XML declaration is a processing instruction that identifies the document as being XML.
- Elements are used to classify data in an XML document to make the data understandable.
- Attributes are information that are often not part of the data but are used in manipulating the data the element holds.
- Validating XML documents is done through a DTD or an XSD.

References

2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070

- W3Schools – XML Tutorial
  http://www.w3schools.com/xml/default.asp