

Designing & Authoring XML Documents

Documents for Data Interchange

3.1: A Simple XML Document

```
<name>Xavier Ampul III</name>
```

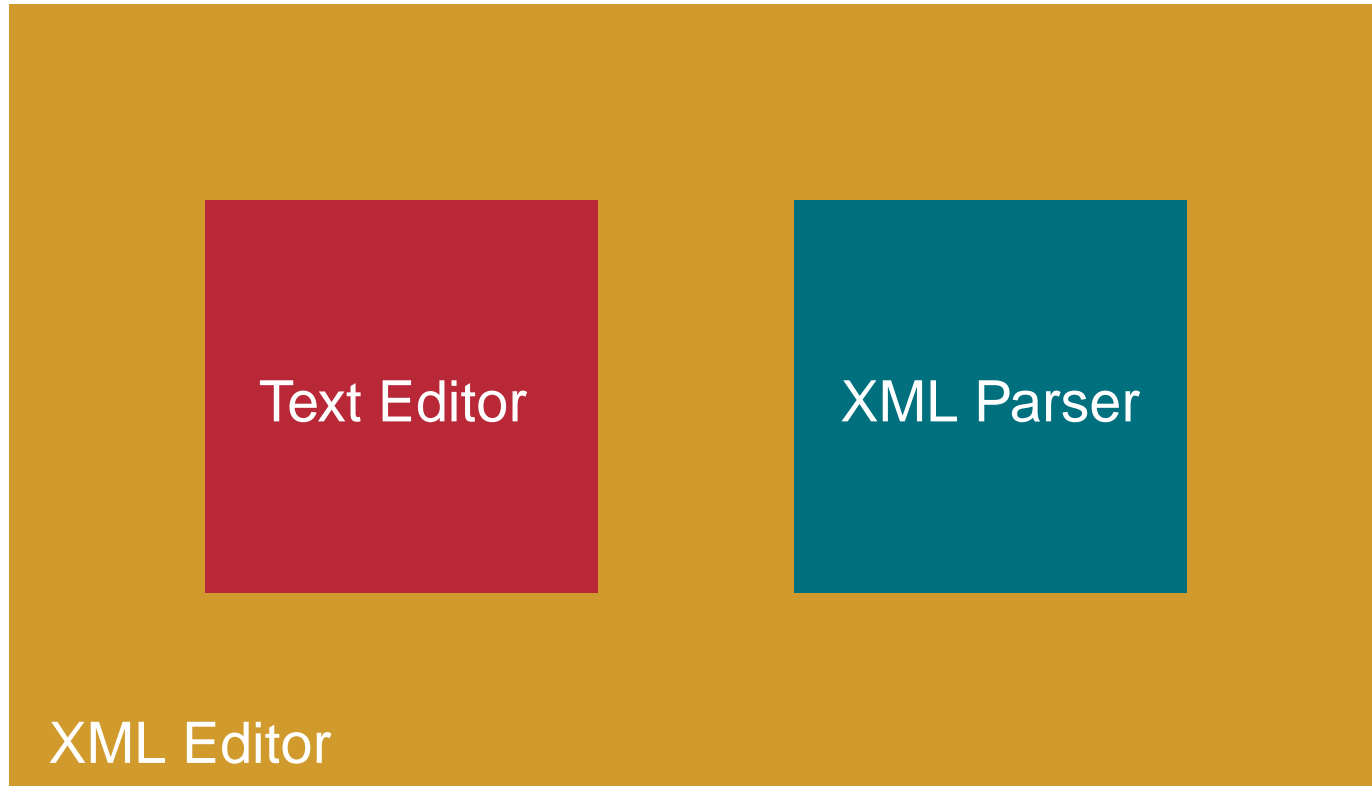
About Example 3.1

- XML documents are made up entirely of text.
- The basic unit of an XML document is an **element**.
- Elements begin with a start tag: `<name>`
- Elements end with a matching end tag: `</name>`
- **Element content** occurs between the start and end tags.
- The simplest form of element content is text.
- XML documents that follow the syntax rules are **well-formed**.

About Example 3.1 (last)

- XML documents that break the syntax rules are **malformed**.

XML Authoring Tools



Plain Text Editors

- Wordpad
- Any other **unicode**-enabled text editor

Some XML Parsers

- Xerces-J (Apache)
- MSXML (Microsoft)
- Crimson (Apache)

Advantages of XML Editors

- Automatic end tag insertion
- Syntax coloring
- Well-Formedness checking
- Validity checking (when schema available)
- Entry helpers (when schema available)
- WYSIWYG editing for non-technical authors

Some XML Editors

- XMLSpy (Altova)
- Oxygen (SyncROsoft)
- XMetaL (Blast Radius)
- Arbortext Editor (PTC)
- Eclipse
- Visual Studio .NET

Using XMLSpy to Check Well-Formedness

- Click on the yellow checkmark icon on the toolbar, or
- Press F7.

3.2: Using Child Elements

```
<name>  
  <firstname>Xavier</firstname>  
  <lastname>Amput</lastname>  
  <suffix>III</suffix>  
</name>
```

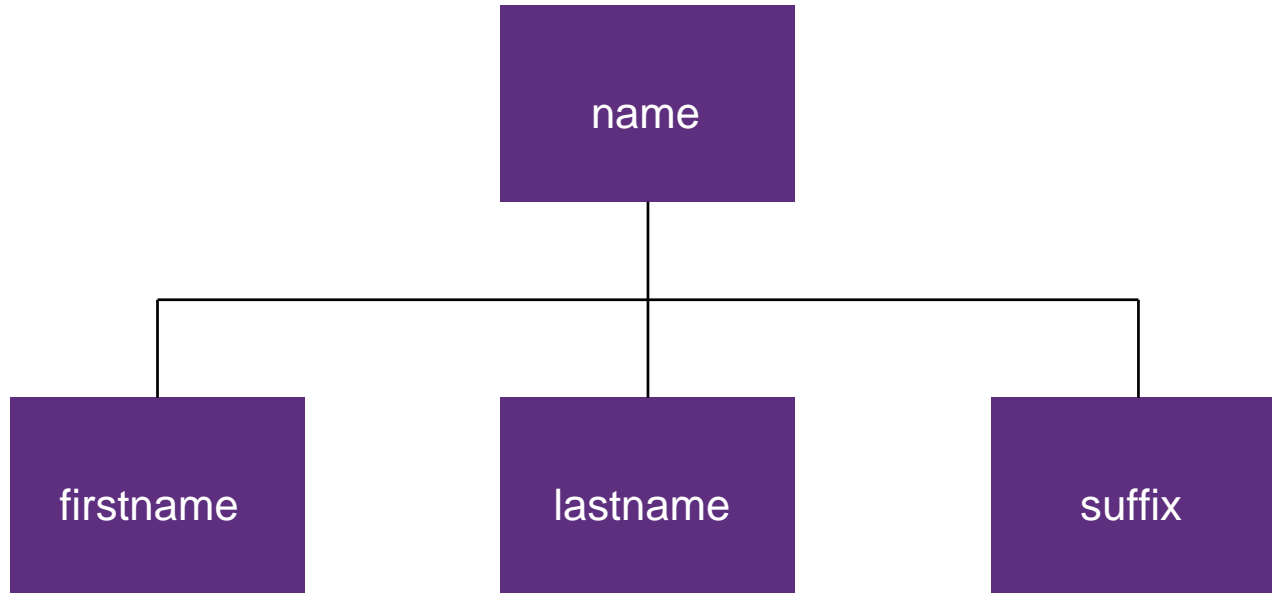
About Example 3.2

- Elements may contain other elements.
- Element containment is best illustrated by the **tree model**.
- Containee elements are called **children**.
- Container elements are called **parents**.
- Adjacent elements at the same level of the hierarchy are called **siblings**.
- A child must be completely contained within its parent.
- Elements may be nested as deeply as needed.

About Example 3.2 (last)

- Indenting child elements helps humans understand document structure.

Tree Model of Example 3.2



3.3: Creating the <employee> Element

```
<employee>  
  <name>  
    <firstname>Xavier</firstname>  
    <lastname>Ampul</lastname>  
    <suffix>III</suffix>  
  </name>  
  <address>  
    <street>1234 Main Street</street>  
    <street>Apartment 401</street>  
    <city>Chicago</city>  
    <state>IL</state>  
    <zip>60610</zip>  
  </address>  
</employee>
```


About Example 3.3

- The element that encloses the entire document is called the **root element** or the **document element**.
- An XML document must have exactly one root element.
- Repeating elements like `<street>` are given the same element name.
- Repeating elements may occur a variable number of times.
- The order of repeating elements is always preserved.

3.4: A More Complete <employee> Element

```
<?xml version="1.0" encoding="utf-8"?>
<employee>
  <name>
    <firstname>Xavier</firstname>
    <lastname>Amput</lastname>
    <suffix>III</suffix>
  </name>
  <address>
    <street>1234 Main Street</street>
    <street>Apartment 401</street>
    <city>Chicago</city>
    <state>IL</state>
    <zip>60610</zip>
  </address>
  <phone>555-555-5555</phone>
</employee>
```

About Example 3.4

- XML documents may begin with an **XML declaration**.
- Most documents specify `version="1.0"`.
- Most english language documents specify `encoding="utf-8"`.
- XML documents should use a **unicode** encoding scheme.

3.5: Creating the <employeeelist> Element

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

```
<employeeelist>
```

```
  <employee>
```

```
    <name>
```

```
      <firstname>Xavier</firstname>
```

```
      <lastname>Ampul</lastname>
```

```
      <suffix>III</suffix>
```

```
    </name>
```

```
    <address>
```

```
      <street>1234 Main Street</street>
```

```
      <street>Apartment 401</street>
```

```
      <city>Chicago</city>
```

```
      <state>IL</state>
```

```
      <zip>60610</zip>
```

```
    </address>
```

```
    <phone>555-555-5555</phone>
```

```
  </employee>
```

```
  <employee>
```

3.5: Creating the <employeeelist> Element (cont.)

```
<name>
  <firstname>Frances</firstname>
  <middleinit>R</middleinit>
  <lastname>Smith</lastname>
</name>
<address>
  <street>559 Primary Avenue</street>
  <city>Evanston</city>
  <state>IL</state>
  <zip>60201</zip>
</address>
<phone>555-555-5590</phone>
<phone>555-555-5591</phone>
</employee>
```

```
<employee>
  <name>
    <firstname>Raymond</firstname>
    <lastname>Jones</lastname>
  </name>
```

3.5: Creating the <employeeelist> Element (last)

```
<address>  
  <street>987 Center Street</street>  
  <city>Chicago</city>  
  <state>IL</state>  
  <zip>60610</zip>  
</address>  
<phone>555-555-0987</phone>  
</employee>  
  
</employeeelist>
```

About Example 3.5

- Repeating elements like `<employee>` must always be contained by a parent.
- **Optional** elements like `<middleinit>` and `<suffix>` are omitted when they are not needed.
- Repeating elements like `<phone>` may occur as many times as needed.

3.6: Using Attributes

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

  <employee>
    <name>
      <firstname>Xavier</firstname>
      <lastname>Ampul</lastname>
      <suffix>III</suffix>
    </name>
    <address>
      <street>1234 Main Street</street>
      <street>Apartment 401</street>
      <city>Chicago</city>
      <state>IL</state>
      <zip>60610</zip>
    </address>
    <phone type="home">555-555-5555</phone>
  </employee>

  <employee>
```


3.6: Using Attributes (cont.)

```
<name>
  <firstname>Frances</firstname>
  <middleinit>R</middleinit>
  <lastname>Smith</lastname>
</name>
<address>
  <street>559 Primary Avenue</street>
  <city>Evanston</city>
  <state>IL</state>
  <zip>60201</zip>
</address>
<phone type="home">555-555-5590</phone>
<phone type="mobile">555-555-5591</phone>
</employee>
```

```
<employee>
  <name>
    <firstname>Raymond</firstname>
    <lastname>Jones</lastname>
  </name>
```

3.6: Using Attributes (last)

```
<address>  
  <street>987 Center Street</street>  
  <city>Chicago</city>  
  <state>IL</state>  
  <zip>60610</zip>  
</address>  
<phone type="mobile">555-555-0987</phone>  
</employee>  
  
</employeelist>
```

About Example 3.6

- **Attributes** are another way for an element to carry content.
- Attributes are made up of **name/value pairs**.
- Attributes appear on the start tag and must be separated by at least one space.
- Attribute values must be quoted.
- An element may not have more than one attribute with the same name.
- Attribute values may not contain markup.

3.7: <employeeelist> Final Touches

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

```
<employeeelist>
```

```
  <!-- The contents of this document are  
        confidential. Distribution is restricted  
        to Xampul Employees and other  
        previously authorized persons.    -->
```

```
<listdate>April 4, 2006</listdate>
```

```
<listregion>Central</listregion>
```

```
<employee>
```

```
  <name>
```

```
    <firstname>Xavier</firstname>
```

```
    <lastname>Ampul</lastname>
```

```
    <suffix>III</suffix>
```

```
  </name>
```

```
  <address>
```

```
    <street>1234 Main Street</street>
```

```
    <street>Apartment 401</street>
```

3.7: <employee> Final Touches (cont.)

```
<city>Chicago</city>  
<state>IL</state>  
<zip>60610</zip>  
</address>  
<phone type="home">555-555-5555</phone>  
</employee>
```

```
<employee>  
  <name>  
    <firstname>Frances</firstname>  
    <middleinit>R</middleinit>  
    <lastname>Smith</lastname>  
  </name>  
  <address>  
    <street>559 Primary Avenue</street>  
    <city>Evanston</city>  
    <state>IL</state>  
    <zip>60201</zip>  
  </address>  
  <phone type="home">555-555-5590</phone>
```

3.7: <employee> Final Touches (last)

```
<phone type="mobile">555-555-5591</phone>  
</employee>
```

```
<employee>  
  <name>  
    <firstname>Raymond</firstname>  
    <lastname>Jones</lastname>  
  </name>  
  <address>  
    <street>987 Center Street</street>  
    <city>Chicago</city>  
    <state>IL</state>  
    <zip>60610</zip>  
  </address>  
  <phone type="mobile">555-555-0987</phone>  
</employee>
```

```
</employee>
```

About Example 3.7

- **XML comments** begin with: `<! --`
- Comments end with: `-->`
- Comments may span multiple lines.
- A comment may not contain the string: `--`
- A comment may not end with: `--->`
- Comments may appear anywhere within normal document text.
- Comments may not appear inside a tag or inside another comment.