

Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's [policies and procedures](#).

Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

© Trinity College Dublin 2020

CSU22041: Information Management I

Parts of an XML Document

... an **art of making information accessible**.

2020-2021

Gaye Stephens gaye.stephens@tcd.ie

XML Example

```
<?xml version='1.0' encoding='ISO-8859-1' standalone='no' ?>
<doc type="book" isbn="1-56592-796-9" xml:lang="en">
  <title>A Guide to XML</title>
  <author>Norman Walsh</author>
  <chapter>
    <title>What Do XML Documents Look Like?</title>
    <paragraph>If you are ...</paragraph>
    <item>
      <paragraph>The document begins ...</paragraph>
    </item>
    <item>
      <paragraph>Empty elements have ...</paragraph>
      <paragraph>In a very ..</paragraph>
    </item>
    <section>...</section>
    ...
  </chapter>
  <chapter>...</chapter>
</doc>
```



Parts of XML documents

Physical parts of XML documents

- Elements
- Attributes
- Entities
- Character Data Sections
- Processing Instructions
- Comments
- XML Declaration or Prolog
- Document Type Declaration



Elements

- Define logical structure and sections of XML documents
- Different content types:
 - Attribute Content
 - Text content
 - Element content
 - Empty.
- Each element must be completely enclosed by another element, except for the root
- Any Element name must start with a letter or underscore but after that can include also digits, fullstops, hyphens.
- Don't start with colon due to namespaces
- Don't include spaces

```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price />
  </book>
</bookstore>
```

Modified from https://www.w3schools.com/xml/xml_elements.asp



Attributes

- Provides additional information about an element
- Attributes are contained within the **start-tag**
- Consists of a name and associated value separated by an equals sign
- **The attribute value must always be enclosed by quotes**
- The order of attributes is insignificant

```
<?xml version="1.0" ?>

<doc type="book"
      isbn="0-201-71050-1">

  <title>Java Gently</title>
  <author>Judy Bishop</author>
  <chapter>
    <paragraph type="abstract">
      In this book ...
    </paragraph>
  </chapter>

</doc>
```



Entities

- Storage units for repeated text

```
<?xml version="1.0"
      standalone="no" ?>

<!DOCTYPE copyright [
  <!ELEMENT copyright >
    <!-- ENTITY c "-//W3C//TEXT
    copyright//EN"
    "http://www.w3.org/xmlspec/c
    opyright.xml">
  ]>

<copyright>&c;</copyright>
```



Entities

- **built-in Entities** are also used for special characters that cannot be typed directly in an XML document, like "<" and "&".
- XML contains a number of 'built-in' entities
 - "
 - '
 - <
 - >
 - &

```
<bullet>
  XML contains a number
  of &apos;built-in&apos;
  entities
  <list>
    <item>&amp;quot;</item>
    <item>&amp;apos;</item>
    <item>&amp;lt;</item>
    <item>&amp;gt;</item>
    <item>&amp;amp;</item>
  </list>
</bullet>
```



Character Data Sections

- Data which is to be parsed is called PCDATA
- An XML parser will not treat the contents of a CDATA section as markup
- Entity references are not resolved in CDATA sections
- Useful for including source code in XML

```
<![CDATA[
    You don't need to escape
    special characters in CDATA
    sections, such as <, >, &, ,
    ' and ".
]]>

<![CDATA[<<< STOP now >>>]]>
```



Processing Instructions- 5 examples

- Pass additional information to application (e.g. parser)
- Application-specific instructions
- Consists of a PI Target and PI Value
 - Syntax
`<?PITarget PIContent?>`
- Processed by applications that recognise the PI Target

```
<?xml-stylesheet  
  type='text/css'  
  href='style.css'?>
```

```
<?xml-stylesheet  
  type='text/xsl'  
  href='style.xsl'?>
```

```
<?myapp filename='test.txt'?>
```

```
<?mso-application  
  progid="Word.Document"?>
```

```
<?mso-application  
  progid="Excel.Sheet"?>
```

Comments 2 examples

- Used to comment XML documents
- Not considered to be part of an XML document
- An XML parser is not required to pass comments to higher-level applications

```
<!-- one-line comment -->
```

```
<!--  
This  
is a  
multi-line comment  
-->
```

XML Declaration or Prolog- 4 Examples

- Placed at the start of an XML document
- Informs XML software of
 - the version of XML the document conforms to
 - the character encoding scheme used in the document
 - whether or not a set of external declarations affect the interpretation of this document
 - Whether external declarations are public or private

https://www.w3schools.com/xml/xml_dtd_intro.asp

```
<?xml version="1.0" ?>
```

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

```
<?xml  
  version="1.0"  
  encoding="UTF-8"  
  standalone="yes" ?>  
<!DOCTYPE person [  
  <!ELEMENT person (name, adult,  
    nationality)>  
]>
```

```
<?xml version="1.0">  
<!DOCTYPE person SYSTEM  
  'person.dtd'>
```



XML Example

```
<?xml version='1.0' encoding='ISO-8859-1' standalone='no' ?>
<doc type="book" isbn="1-56592-796-9" xml:lang="en">
  <title>A Guide to XML</title>
  <author>Norman Walsh</author>
  <chapter>
    <title>What Do XML Documents Look Like?</title>
    <paragraph>If you are ...</paragraph>
    <item>
      <paragraph>The document begins ...</paragraph>
    </item>
    <item>
      <paragraph>Empty elements have ...</paragraph>
      <paragraph>In a very ..</paragraph>
    </item>
    <section>...</section>
    ...
  </chapter>
  <chapter>...</chapter>
</doc>
```



Well formed XML

- XML Declaration required
- At least one element
 - Exactly one root element
- Empty elements are written in one of two ways:
 - Closing tag (e.g. "<Name></Name>")
 - Special start tag (e.g. "<Name />")
- For non-empty elements, closing tags are required
- Attribute values must always be quoted
- Start tag must match closing tag (name & case)
- Correct nesting of elements
 - Example **incorrect nesting** and **incorrect case**

```
<full_name>  
<first_name>  
John </Full_name>  
</first_name>
```



Exercise- check out exercises on Blackboard

- Spot the 6 deliberate errors in the XML opposite

```
<?xml version="1.0"!>
<!DOCTYPE catalog SYSTEM "books.dtd">
<catalog>
  <book id='bk101' type='softback'>
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish-date>
    <description>An in-depth look at creating applications with XML.
  </book></description>
  <book id='bk102' >
    <author nationality=irish>Jenkins, Fred</Author>
    <title>XML Technology Guide</title>
    <price>50.00</price>
    <publish date>2000-10-01</publish date>
    <description>An in-depth look at using XML
technologies.</description>
    <stocked_by>Easons</stocked_by>
    <stocked_by>Amazon</stocked_by>
  </book type='hardback'>
</catalog>
```



**That's All
Folks
Thank You
for
Listening**

