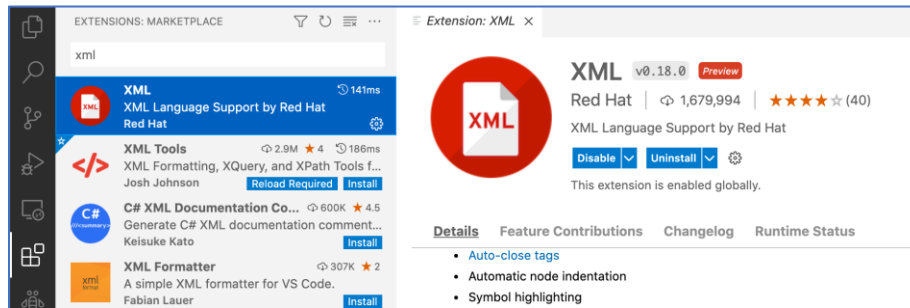


INT305 Lab 1 XML

Software: Visual Studio Code

<https://code.visualstudio.com>

Extensions: XML, XML Language Support by Read Hat



Simple XML document example

```
<?xml version="1.0" encoding="UTF-8"?>
<book>
  <name>A Song of Ice and Fire</name>
  <author>George R. R. Martin</author>
  <language>English</language>
  <genre>Epic fantasy</genre>
</book>
```

Line 1: XML Prolog

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

It is optional and should always be the first line of the document. It defines the XML version and encoding used in XML document

Line 2: `<book>` is the root of this XML document.

A XML document should always have one root element and at the end of the document this root element needs a closing tag `</book>`.

Line 3-5: The child elements of the root element `<book>`

The tags `<name>`, `<author>`, `<language>` and `<genre>` are the child elements.

Example 2: Students Data

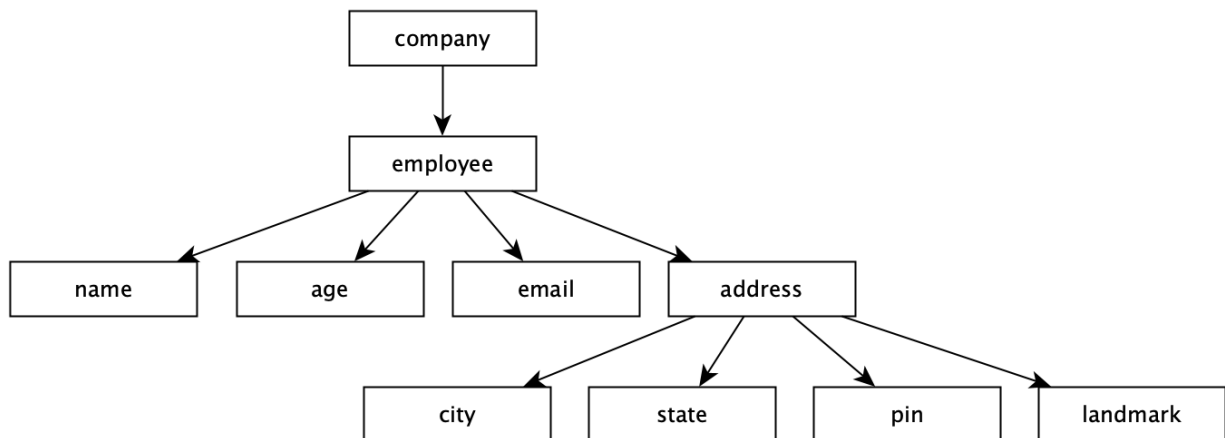
```
<?xml version="1.0" encoding="UTF-8" ?>
<students>
  <student>
    <name>Rick Grimes</name>
    <age>35</age>
    <subject>Maths</subject>
    <gender>Male</gender>
  </student>
  <student>
    <name>Daryl Dixon </name>
    <age>33</age>
    <subject>Science</subject>
    <gender>Male</gender>
  </student>
  <student>
    <name>Maggie</name>
    <age>36</age>
    <subject>Arts</subject>
    <gender>Female</gender>
  </student>
</students>
```

XML Tree Structure

XML document has a tree structure, where the root element is at the top and the child elements are connected to the root element, the same way, how leaves are connected to tree through branches.

Example 3: Company Data

```
<?xml version="1.0" encoding="UTF-8"?>
<company>
  <employee>
    <name>Negan</name>
    <age>40</age>
    <email>imnegan@twd.com</email>
    <address>
      <city>Noida</city>
      <state>Uttar Pradesh</state>
      <pin>201301</pin>
      <landmark>Near hill top</landmark>
    </address>
  </employee>
</company>
```



XML Syntax or Rules

1. XML Prolog is optional and should be the first line
2. Root Element is mandatory and must have only one in XML
3. XML is case sensitive
4. Elements should not overlap
5. Attributes are defined in opening tags and contain name and value pairs.
6. Attribute names are case sensitive and should not be in quotation marks.
7. Attribute values should be in single or double quotation.
8. An element with no content (Empty element) must use an empty tag or self-closing tag such as `<available></available>` → `<available />`
9. White-spaces are preserved in XML
10. Use `<!-- This is just a comment -->` for a comment in XML document

Task 1: Check the following XML document whether is well-formed or not. If it is not well-formed, give a reason and correct it. (Do it by yourself, do not use editor)

```
<library>
  <book id=2003-Blanken>
    <title>Intelligent XML Search</title>
    <editor>Henk A. Blanken et al.</editor>
    <publisher>Springer Verlag</PUBLISHER>
  </book>
  <journal id="TODS_1_2003" publ_year="2003">
    <title>ACM Transactions on Databases</title>
    <volume>24<number>1</number></volume>
  </journal>
  <misc id="Dipl2002-12" id="TR2002-01-04">
    <author>U. Known</author>
    <title>The Diploma Thesis that didn't appear <year>2002</year>
    <pages>0</pages>
  </misc>
</library>
```

Write your answer below:

1. `<book id="2003-Blanken">`
ตรง value ต้องใส่ “ ” ด้วย
2. `<publisher>Springer Verlag</publisher>`
tag ปิดควรใช้เป็น lowercase ตาม tag เปิด เพราะ XML เป็น case sensitive
3. `<misc id="Dipl2002-12">`
1 attribute มีได้แค่ value เดียว
4. `<title>The Diploma Thesis that didn't appear
<year>2002</year>
</title>`
ต้องมี tag ปิดของ title ด้วย

XML Naming Rules

1. Tag names are case sensitive and cannot contain space.
2. Tag name must begin with letters, "_" or ":"
3. Tag name may contain letters, digits, ".", "_", and ":" (Avoid Colons (":") are reserved for namespaces)
4. Tag name should be descriptive, short, and simple name such as `<person>` `<firstname>` `<book>` `<title>` not like this: `<the_title_of_book>`

Naming Styles

| | |
|---------------------------------|-------------|
| <code><firstname></code> | Lower case |
| <code><FIRSTNAME></code> | Upper case |
| <code><first_name></code> | Snake case |
| <code><FirstName></code> | Pascal case |
| <code><firstName></code> | Camel case |

***If you choose a naming style, it is good to be consistent!

XML Attributes

XML attributes are a way to add additional data to XML element. Attributes contains data in form of name & value pairs.

Syntax

`<element attr1="value1" attr2="value2"> ... </element>`

Rules for XML attributes

1. Attributes are name and value pairs. The attribute name should not be in quotes, however the attribute value must always in quotes (single or double).
2. XML element can have more than one attributes.
3. The attribute name cannot contain duplicate in the element.

Task 2: Which of the following Tags names are valid, which are not? Explain your answer (Do it by yourself, do not use editor)

| Tag Name | Valid or Not valid and Explain |
|---|---|
| <code><lecture></code> | Valid |
| <code><student and teacher></code> | Not Valid เพราะชื่อ element ต้องไม่เว้นวรรค |
| <code><student&teacher></code> | Not Valid เพราะชื่อ element ไม่ควรมีสัญลักษณ์ & |
| <code><sm0rebr0d></code> | Valid |
| <code><DibertComicStrip></code> | Valid |
| <code><XMLisFun></code> | Valid |
| <code><4u></code> | Not Valid เพราะชื่อ element ขึ้นต้นด้วยตัวเลขไม่ได้ |
| <code><student.and.teatcher></code> | Valid |
| <code><></code> | Not Valid เพราะเป็น tag เปล่า |

Task 3: Check the following XML attributes in an element whether are valid or not valid. If it is not valid, give a reason and correct it. (Do it by yourself, do not use editor)

3.1. `<room id=100 >`

`<price>200</price>`

`</room>`

3.2. `<message category="important" category="public">Welcome to XML</message>`

3.3. `<temperature "fahrenheit">100</temperature>`

3.4. `<person name="Scott" surname="Hanselman" age="34" />`

3.5. `<car brand="Ford Mustang">`

`<generation>6</generation>`

`<release>December 5,2013</release>`

`</car>`

Write your answer below:

3.1. Not Valid เพราะ value ของ id ต้องมี “” `<room id="100" >`

3.2. Not Valid เพราะ value ของ category ควรมีแค่ value เดียว `<message category="important">Welcome to XML</message>`

3.3. Not Valid เพราะไม่ได้ประกาศชื่อ attribute `<temperature type="fahrenheit">100</temperature>`

3.4. Valid

3.5. Valid

XML Attributes vs. XML Elements

Let have a look at the following two examples:

Here sex is an attribute

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Here sex is an element

```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

In both examples, the XML document contain the same data. However, you should always prefer XML element over XML attribute.

The following table shows the advantages of using XML elements over XML attributes:

| | Attributes | Elements |
|----------------------------|-------------------|-----------------|
| Multiple values | Not allowed | Allowed |
| Expandable | Not allowed | Allowed |
| Show in XML tree structure | Not show | Show |
| Test against a DTD | Difficult | Easy |
| Handle by the programming | Difficult | Easy |
| Ordering | Not matter | Do matter |

Reference:

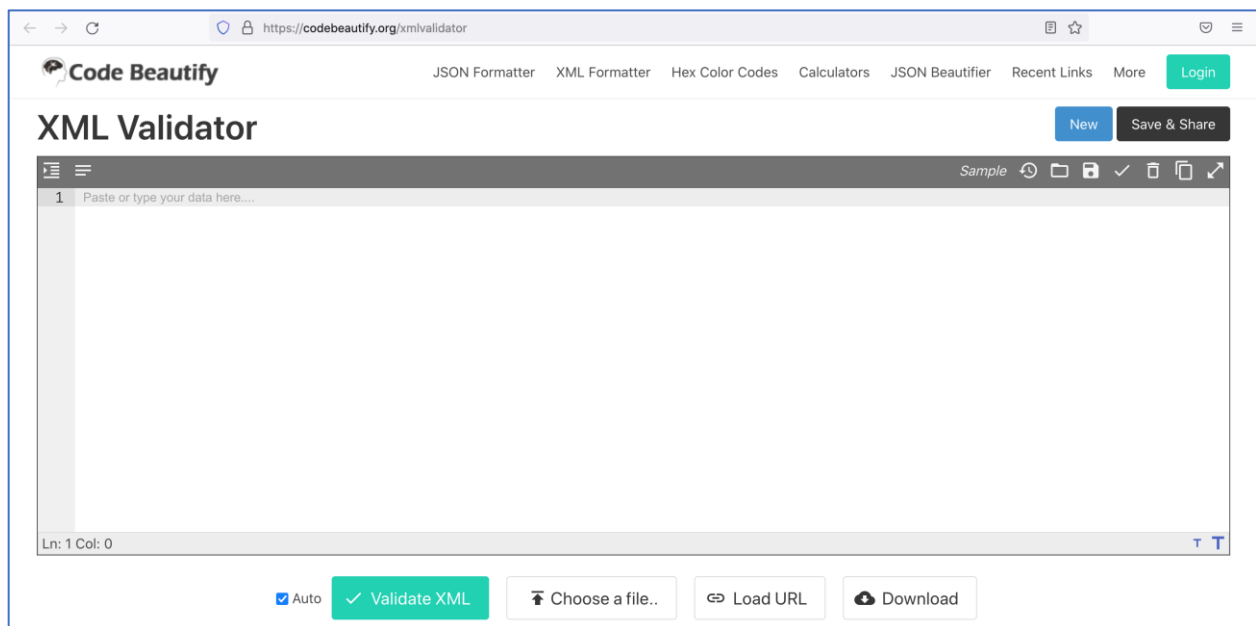
- https://www.w3schools.com/xml/xml_dtd_el_vs_attr.asp
- <https://stackoverflow.com/questions/1096797/should-i-use-elements-or-attributes-in-xml>
- https://nedbatchelder.com/blog/200412/elements_vs_attributes.html

Task 4: Write XML document (Use Visual studio code and verify it in the URL below)

Create a well-form XML document to store students' data. Each student's data should contain as follows:

- Student id
- Firstname
- Last name
- Gender
- Email
- University
 - Name
 - Faculty
 - Major

List at least two students. Save the file and copy the XML document and validate the XML document at <https://codebeautify.org/xmlvalidator>



Reference:

- https://docstore.mik.ua/orelly/xml/xmlnut/ch02_04.htm
- https://www.w3schools.com/xml/xml_elements.asp
- <https://beginnersbook.com/2018/10/xml-tutorial-learn-xml/>
- <https://stackoverflow.com/questions/29082189/why-use-an-empty-element-in-xml>
- <https://www.guru99.com/xml-tutorials.html>
- <https://www.oracle.com/technical-resources/articles/wang-whitespace.html>

Write your answer below:

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
  <student id = "1">
    <id>63130500134</id>
    <name>
      <firstname>Arisa</firstname>
      <lastname>Audomkhachonkitti</lastname>
    </name>
    <gender>female</gender>
    <email>arisa.ice@mail.kmutt.ac.th</email>
    <university>
      <name>KMUTT</name>
      <faculty>SIT</faculty>
      <major>IT</major>
    </university>
  </student>
  <student id = "2">
    <id>63130500086</id>
    <name>
      <firstname>Pichaya</firstname>
      <lastname>Pairin</lastname>
    </name>
    <gender>male</gender>
    <email>pichaya.game@mail.kmutt.ac.th</email>
    <university>
      <name>KMUTT</name>
      <faculty>SIT</faculty>
      <major>IT</major>
    </university>
  </student>
</students>
```

XML Validator

[Add to Fav](#) [New](#) [Save & Share](#)

Sample

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

2 <students>

3 <student id = "1">

4 <id>63130500134</id>

5 <name>

6 <firstname>Arisa</firstname>

7 <lastname>Audomkhachonkitti</lastname>

8 </name>

9 <gender>female</gender>

10 <email>arisa.ice@mail.kmutt.ac.th</email>

11 <university>

12 <name>KMUTT</name>

13 <faculty>SIT</faculty>

14 <major>IT</major>

15 </university>

16 </student>

17 <student id = "2">

18 <id>63130500086</id>

19 <name>

20 <firstname>Pichaya</firstname>

Ln: 23 Col: 29 size: 976 B

☒ Auto

☒ Validate XML

Valid XML

