SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)
DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJ19ITL801 **DATE:** 25-01-2024

COURSE NAME: Semantic Web Technology Laboratory **CLASS:** Final Year B.Tech I1-1

NAME: Rishikesh Sharma **SAP ID:** 60003200102

EXPERIMENT NO. 1

CO/LO: Apply Semantic web technologies to real world applications.

AIM / OBJECTIVE:

Parsing the XML dataset and comparing Different XML Serialization Formats and Their Impact on Data Size and Processing Time.

DESCRIPTION OF EXPERIMENT:

XML (Extensible Markup Language) is a standard for storing and exchanging data that is widely used for exchanging data on the web. XML data can be stored in different serialization formats, each with its own advantages and disadvantages. The most used serialization formats are XML, Compact XML (CXML), and Binary XML (BXML).

INPUT DATA / DATASET:

Dataset is an XML dataset of countries.

Exp01.py | output.xml | country_data.xml

country_data.xml

```xml
<?xml version="1.0"?>
<data>
    <country name="Liechtenstein">
        <rank updated="yes">2</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank updated="yes">5</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank updated="yes">69</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

PROCEDURE / ALGORITHM:

1. Open Notepad and create XML data
2. Save the XML data as a .XML file, .CXML file and a .BXML file.
3. Open Google Colab and upload your XML file.

TECHNOLOGY STACK USED:

Google Colab

SOURCE CODE (OPTIONAL):

XML Parsing:

Parsing XML using Element Tree XML API in Python.

1. Get the root element

```
[5]  import xml.etree.ElementTree as ET

[14] tree = ET.parse('/content/sample_data/country_data.xml')
     root = tree.getroot()
```

2. Get the first child of root element

```
root[0].tag
'country'
```

3. Get the attribute for the child of the root element.

```
[32] root[0].attrib
{'name': 'Liechtenstein'}
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

4. Print all the tags and attributes that are present within the child tag

```
for child in root[0]:
  print(child.tag, child.attrib)
```

```
rank {}
year {}
gdppc {}
neighbor {'name': 'Austria', 'direction': 'E'}
neighbor {'name': 'Switzerland', 'direction': 'W'}
```

5. Print the text present within these tags

```
[36] for child in root[0]:
       print(child.tag, child.text, child.attrib)
```

```
rank 1 {}
year 2008 {}
gdppc 141100 {}
neighbor None {'name': 'Austria', 'direction': 'E'}
neighbor None {'name': 'Switzerland', 'direction': 'W'}
```

6. Iterate through all the tags with a specific name.

```
for neighbor in root.iter('neighbor'):
  print(neighbor.attrib)
```

```
{'name': 'Austria', 'direction': 'E'}
{'name': 'Switzerland', 'direction': 'W'}
{'name': 'Malaysia', 'direction': 'N'}
{'name': 'Costa Rica', 'direction': 'W'}
{'name': 'Colombia', 'direction': 'E'}
```

7. Add and remove elements from the xml file

```
[38] for rank in root.iter('rank'):
       new_rank = int(rank.text) + 1
       rank.text = str(new_rank)
       rank.set('updated', 'yes')

tree.write('output.xml')
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

```
<?xml version="1.0"?>
- <data>
  - <country name="Liechtenstein">
      <rank updated="yes">2</rank>
      <year>2008</year>
      <gdppc>141100</gdppc>
      <neighbor name="Austria" direction="E"/>
      <neighbor name="Switzerland" direction="W"/>
    </country>
  - <country name="Singapore">
      <rank updated="yes">5</rank>
      <year>2011</year>
      <gdppc>59900</gdppc>
      <neighbor name="Malaysia" direction="N"/>
    </country>
  - <country name="Panama">
      <rank updated="yes">69</rank>
      <year>2011</year>
      <gdppc>13600</gdppc>
      <neighbor name="Costa Rica" direction="W"/>
      <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

```
[40] for country in root.findall('country'):
       rank = int(country.find('rank').text)
       if rank > 50:
         root.remove(country)

     tree.write('output1.xml')
```

Output:

```
<data>
    <country name="Liechtenstein">
        <rank updated="yes">3</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E" />
        <neighbor name="Switzerland" direction="W" />
    </country>
    <country name="Singapore">
        <rank updated="yes">6</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N" />
    </country>
</data>
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Comparing Different XML Serialization Formats and Their Impact on Data Size and Processing

Time

1. Choose an XML data sample to use for the experiment. This can be any XML data that is suitable for your purpose, such as a simple XML file, an XML file with many elements, or an XML file with nested elements.
2. Store the XML data sample in three different serialization formats: XML, CXML, and BXML.
3. Measure the size of each file and note it down. Parse each file using an XML parser and measure the time taken to parse it.

OBSERVATIONS / DISCUSSION OF RESULT:

XML:

- **Size:** 687 bytes
- **Time to parse:** 0.005s

CXML:

- **Size:** 687 bytes
- **Time to parse:** 0.397s

BXML:

- **Size:** 687 bytes
- **Time to parse:**1.423.

The file size for all is the same though the parsing time is best in this order C**XML>XML>BXML**

CONCLUSION:

Parsing is fastest when the file is in XML format. We used XML files and parsed over it using the Element Tree API.

REFERENCE:

[1] XML specification: https://www.w3.org/TR/REC-xml/

[2] Compact XML (CXML) specification: https://www.w3.org/TR/cxml/

[3] Binary XML (BXML) specification: https://www.w3.org/TR/binaryxml/

[4] Python XML Parser Tutorial | Read and Write XML in Python | Python Training | Edureka – YouTube