0301402 INTRODUCTION TO XML

UNIT	MODULES	WEIGHTAGE
1	Introduction to XML	20 %
2	Document Type Definition (DTD)	20 %
3	XML Namespace	20 %
4	XML Schema	20 %
5	Extensible StyleSheet Language (XSL)	20 %

UNIT -3 XML Namespace

- Need for XML Namespace
- Namespace – The Big Data
- Declaring Namespaces
- Namespace Scoping
- Documents with Multiple Namespaces
- Elements with No Namespace
- Attributes andNamespace
- Namespace Processing
- Use of Namespace-Example
- Problems with Namespace

Need of XML Namespace

- Element and Attribute Names can be Ambiguous
- Demo1.xml
- **Need a way to address the problem**
  - Adopt industry standard document formats and naming conventions.
  - Use verbose element names:
    - i.e BookTitle, courtesyTitle etc.
  - Use some name qualifier that is already established as qunique.
    - i.e a domain name qualified URI

Namespace : The Big Idea

- Each element name and attribute name could be expressed as

  Uniform Resource Identifier (URI) + name

  i.e - <http://www.library.com/books:title>
- **Two problem for the above formate:**
  - It is not well-formed XML under the 1.0 specification.
  - It is a lot of typing.

Namespace : The Big Idea

- If it were possible to create a synonym for the URI and replace occurrences of the URI with that synonym,
- The amount of typing would be reduced and, if handled correctly, the result would be compatible with XML 1.0
- i.e specify
- books=”http://www.library.com/books” and code the elements as

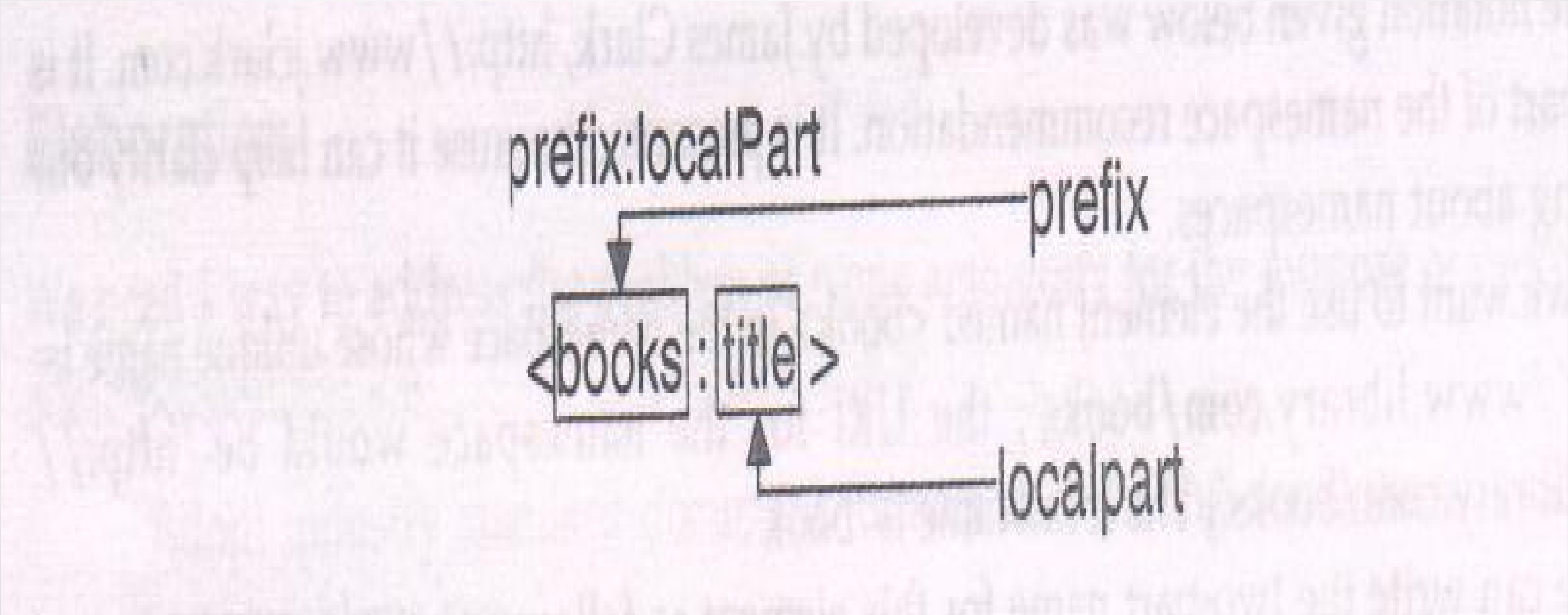
  <books:title>
- **This concept forms the basis of the XML Namespace specification.**
- **These URI qualifiers are called NameSpaces.**

XML Namespace

- If URIs are different, they represent different Namespace.
- The Namespace specification deals with the mechanics of associating a **URI qualifier with element and attribute names to create two part name** that are unique and free of ambiguity.
- The Namesapce specificaion refers to these two part names as **Qualified Names** or **Qnames**
- URIs are not checked by the processor to ensure that they exist.

Qualified Names (QNames)

- Qnames are used in place of element and attribute names.
- Qname have a prefix and local part – they look like this:



Declaring Namespaces

- When using prefixes in XML, **a namespace for the prefix must be defined.**
- The namespace **can be defined by an xmlns attribute** in the start tag of an element.
- The namespace declaration has the **following syntax.**

xmlns:prefix="URI".

<prefix : elementName xmlns:prefix=”URI” />

- DEMO_2.xml
- DEMO_3.xml

Namespace Scoping

- Every elementor attribute name that is in the namespace has the appropriate namespace prefix in front of it.
- The namespace of an element depends on:
  - **The namespace prefix used.**
  - **The declaration of a default namspace.**
- When a namespace prefix is declared, it remains in scope for:
  - **Attributes of the element where it is declared.**
  - **Child elements of the element where it is declared.**
  - **Unless the prefix is redefined on a nested element.**

Default Namespace

- Defining a default namespace for an element saves us from using prefixes in all the child elements.
- It has the following syntax:

```
<elementName xmlns=”URI” />
```

- *i.e*

```
<table xmlns="http://www.w3schools.com/furniture">
    <name>African Coffee Table</name>
    <width>80</width>
    <length>120</length>
</table>
```

Documents with Multiple Namespace

- Namespaces is to allow documents to use names from multiple namespaces without interfering with each other.

- DEMO_4.xml

Element with no Namespace

- DEMO_5.xml
- If a name has no prefix and there is no default namespace, then the name is not in any namespace.

Attributes and Namespaces

- There are two interacting rules that affect attributes and namespaces:
  - Attributes are not affected by a default namespace declaration.
  - Attributes on a single element must be unique.
- DEMO_6.xml

Use of Namespaces

- DEMO_7.xml

Problem with Namespaces

- Namespace recommendation came after XML1.0, so it is not considered in the specification.
- **This means there are places where namespaces and DTD-bound XML 1.0 don't intergrate well.**
- **DTD don't integrate well.**
- **Testing the equality of namespaces is not handled by the parser.**
- **There's no easy way to test the equality of two namespace except to get the two namespace.**

Best Practices

- **When to use namespace**
  - When the data requires uniqueness for application processing.
  - When the need to combine a schema with other grammers is necessary.
- Performance implications
  - Namespace processing slows down the parser and/or increases memory use.
- Dont use relative URIs for namespace identifiers
- Pick the default namespace carefully

Best Practices

- Don't declare more than one prefix for a namespace URL
- Be careful with attributes when using namespace.

UNIT 3 COMPLETED

- Assignment Submission
  - Theory : 27 / 12 / 2021
  - Practical : 27 / 12 / 2021
- CEC Submission
  - Theory : 27 / 12 / 2021
  - Practical : 27 / 12 / 2021