# 1, What is XML

- XML = eXtensible Markup Language

- In XML we use tags <> or marks

= and you can use your own tags ,… tags are not limited and not predefined => it can be **extended**

* XML is used to store and transport data
* XML is a W3C Recommendation(World Wide Web Consortium)
* A software and hardware independent way of storing, transporting, and sharing data
* Both Human and machine can read

-XML Validation

Syntax|Structure

Is well formed(satisfy all the basic syntax condition) behaves according to predefined DTD or XML schema

* DTD – Document Type Definition
* XSD – XML Schema Definition

Example code

|  |
| --- |
| <friendsList>      <friend>          <name>Alex</name>      </friend>      <friend>          <name>Mary</name>      </friend>      <friend>          <name>Rachel</name>      </friend>      <bestFriend>          <name>Anna</name>      </bestFriend>  </friendsList> |

it satisfies all the basic syntax and rules and conditions and then we can also check our xml against some predefined schema . a dtd that is document type definition or can be a xst called xml schema. definition in most of the cases the creators or the developers of the xml will provide you this schema and you will have to check your xml against this schema. it will contain the structure of the xml how it should be structured. what should be the text, what is mandatory, what is not mandatory, what should be the data type inside the text. all this will be here so kind of a grammar and content check is done here. most of the cases today we use xst now in case you do not have a xst or a dtd or any schema. we can also do the reverse that is we can use our xml and can generate a schema out of it.

# 2,Learn XML Syntax From Scratch

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <friendsList>  <friend>  <name>Alex</name>  </friend>  <friend>  <name>Mary</name>  </friend>  <friend>  <name>Rachel</name>  </friend>  <bestFriend>  <name>Anna</name>  </bestFriend>  </friendsList> |

**Root** Must contain a root element that is parent of all elements

|  |
| --- |
| <friendsList> |

**XML Prolog** is optional and is the first line

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

**Tags**  All elements must have a closing tag

XML tags are case sensitive

|  |
| --- |
| <bestFriend>  <name>Anna</name>  </bestFriend> |

Notes:

the XML prolog does not have a closing tag

The prolog is not a part of the XML message

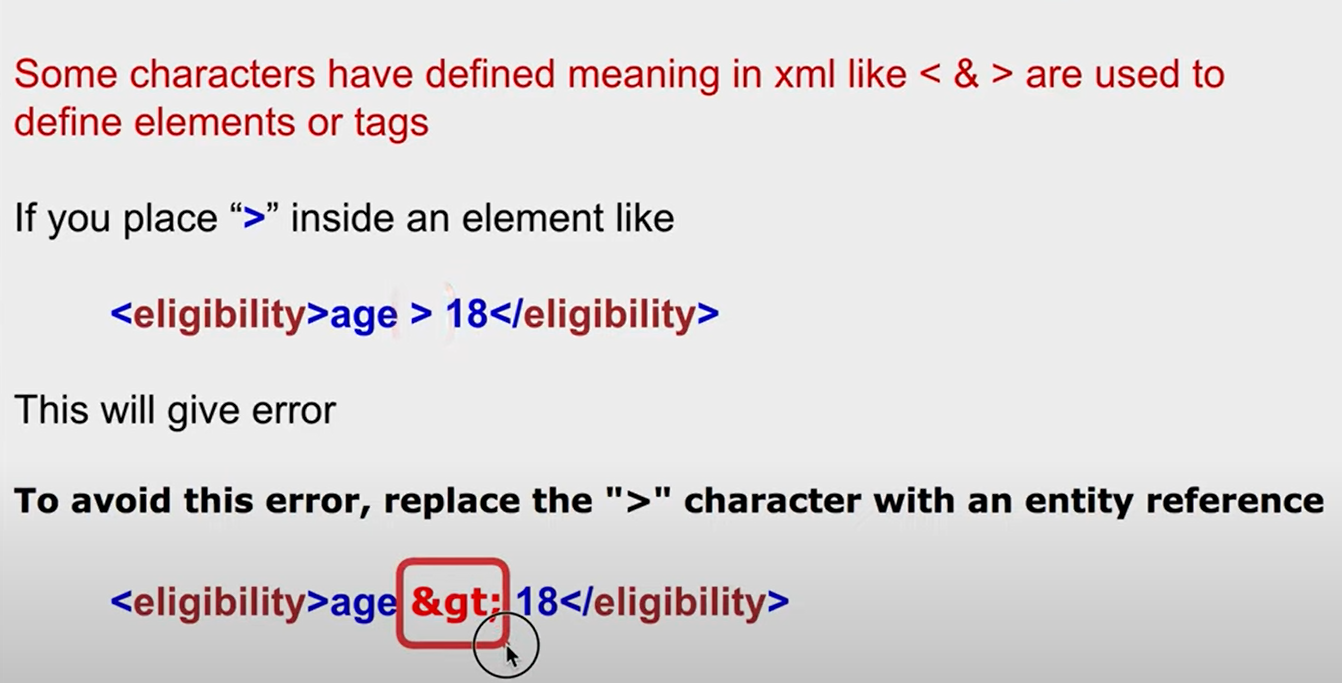
**Attributes**  must be quoted

|  |
| --- |
| <friendsList>      <friend gender=”male”>          <name>Alex</name>  <age>25</age>      </friend>      </friendsList> |

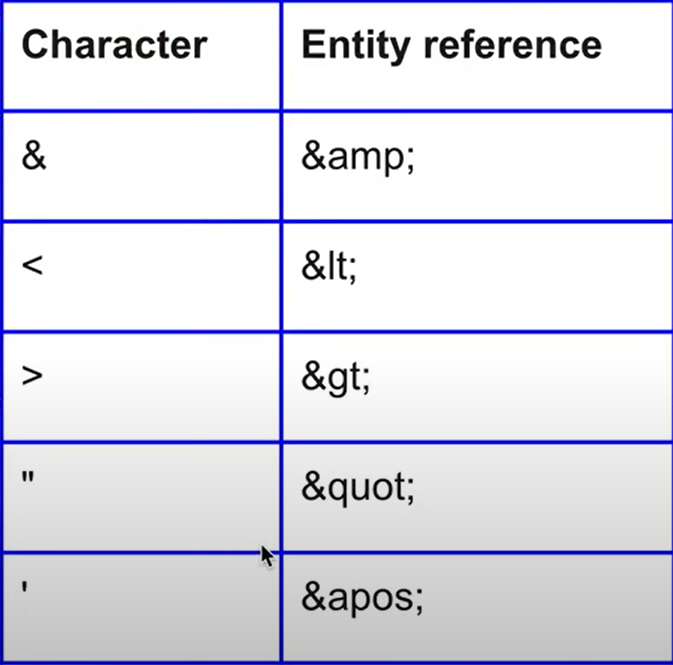
**Nesting** XML Elements must be nest properly

|  |
| --- |
| <root>      <child>          <subchild>..</subchild>      </child>  </root> |

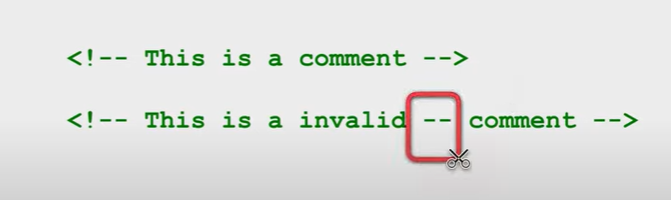
**Entity Reference**

****

The chart

****

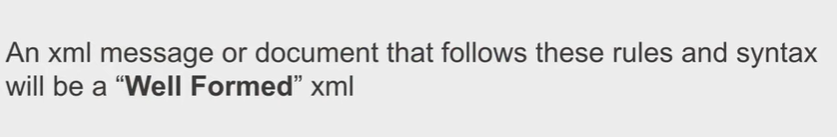
**Comment**

****

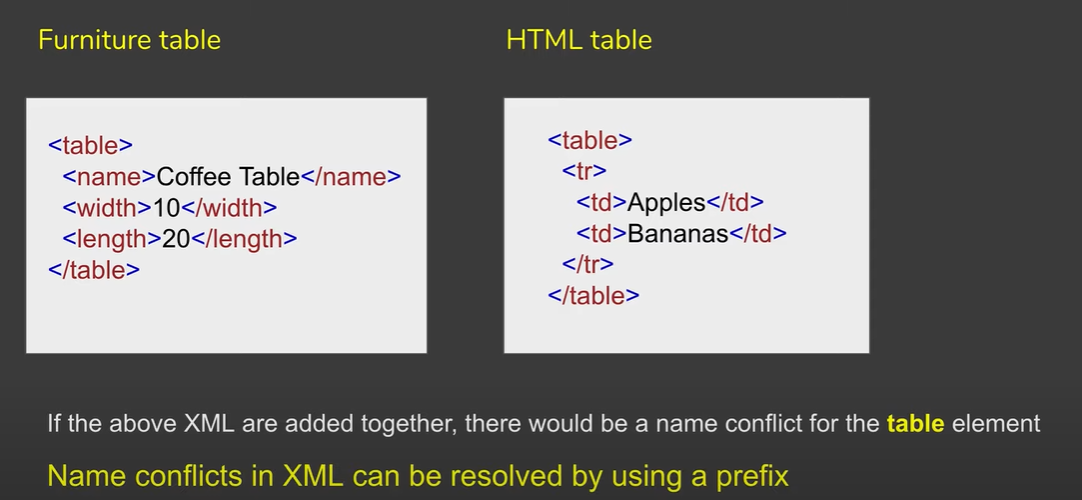
**White spaces**

****

=>>



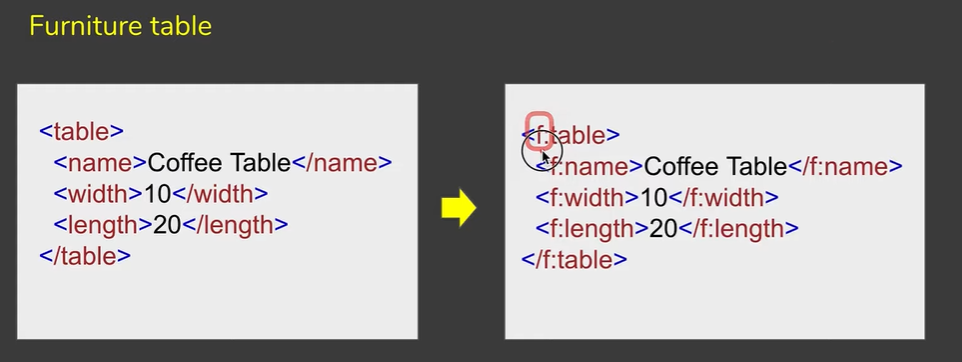
# 3,XML Namespace

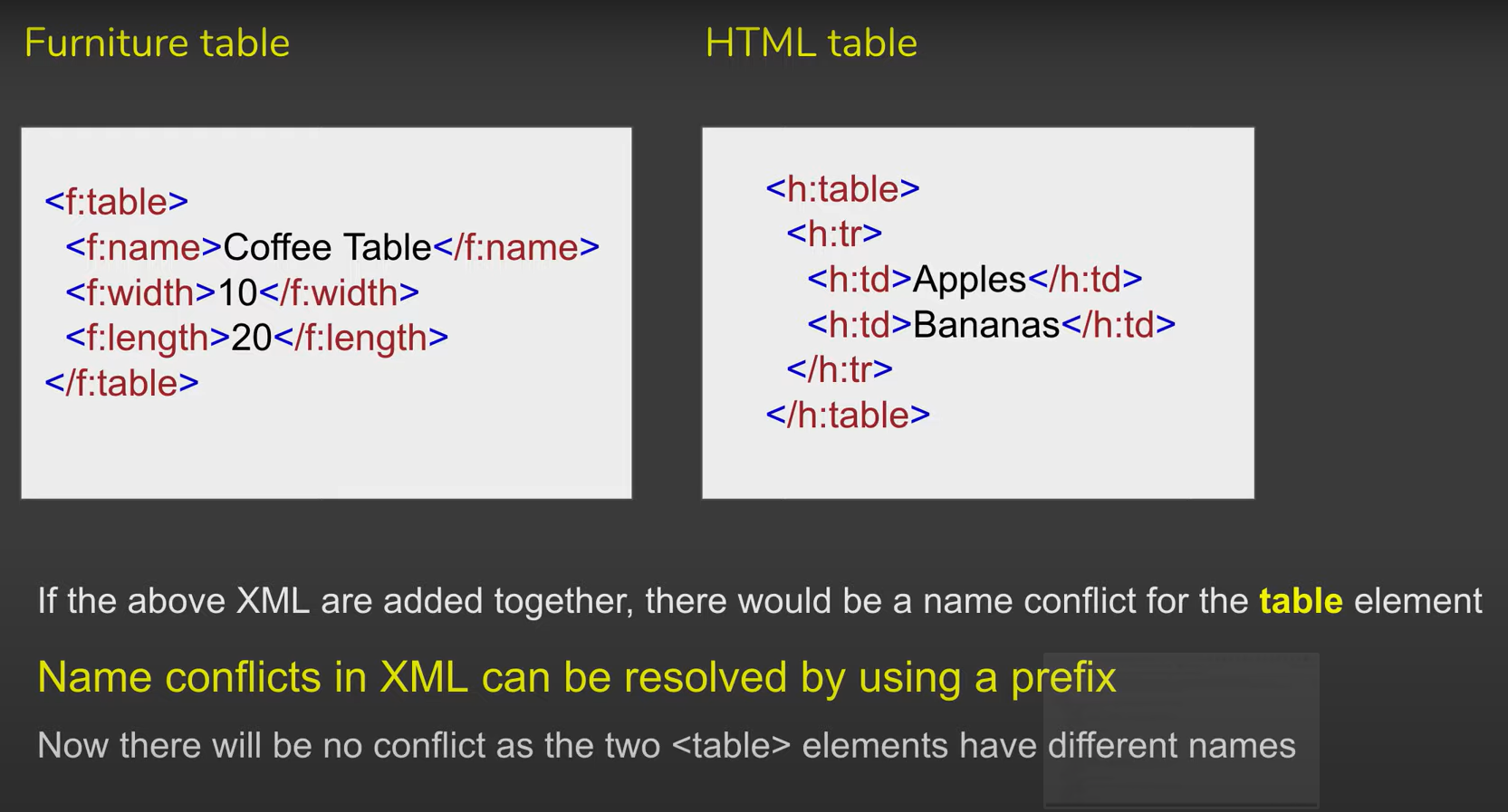


if the above XML are added together, there would be a name conflict for the table element.

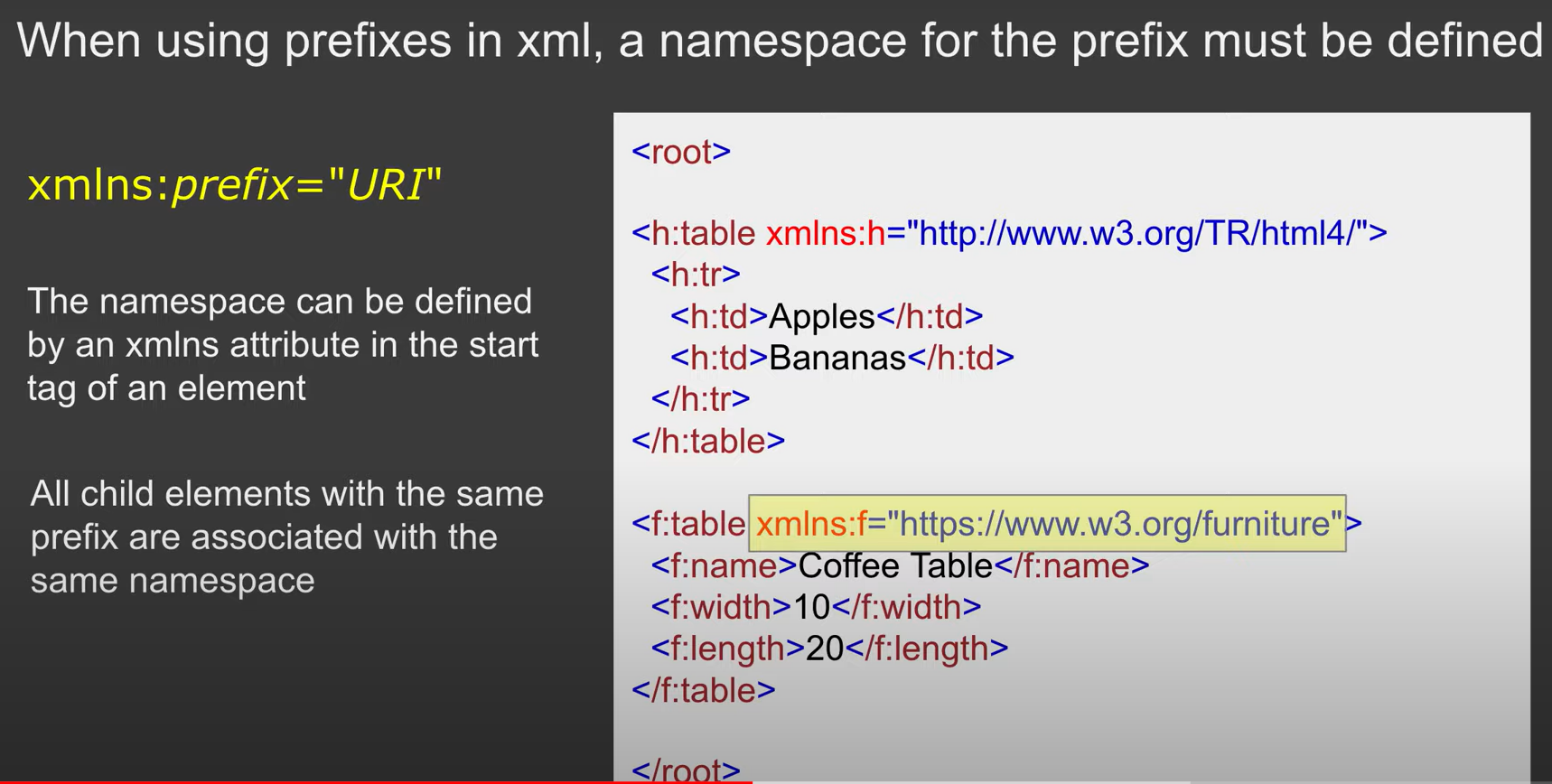
Name conflicts in XML can be resolved by using a prefix.

**Prefix**





The **xmlns** attribute

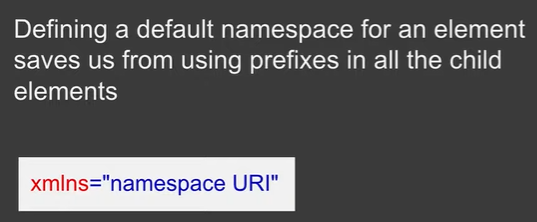


|  |  |
| --- | --- |
|  |  |

**Default namespace**

Defining a default namespace for an element saves us from using prefixes in all the child elements

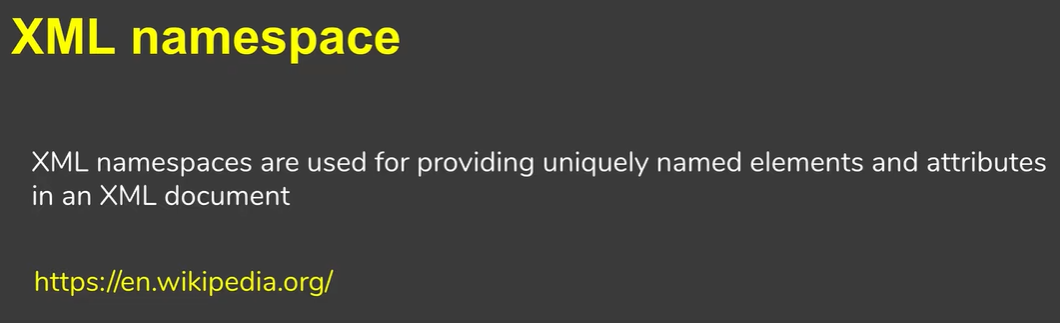
(Việc xác định không gian tên mặc định cho một phần tử giúp chúng ta không sử dụng tiền tố trong tất cả các phần tử con)



Example: child element to be associated



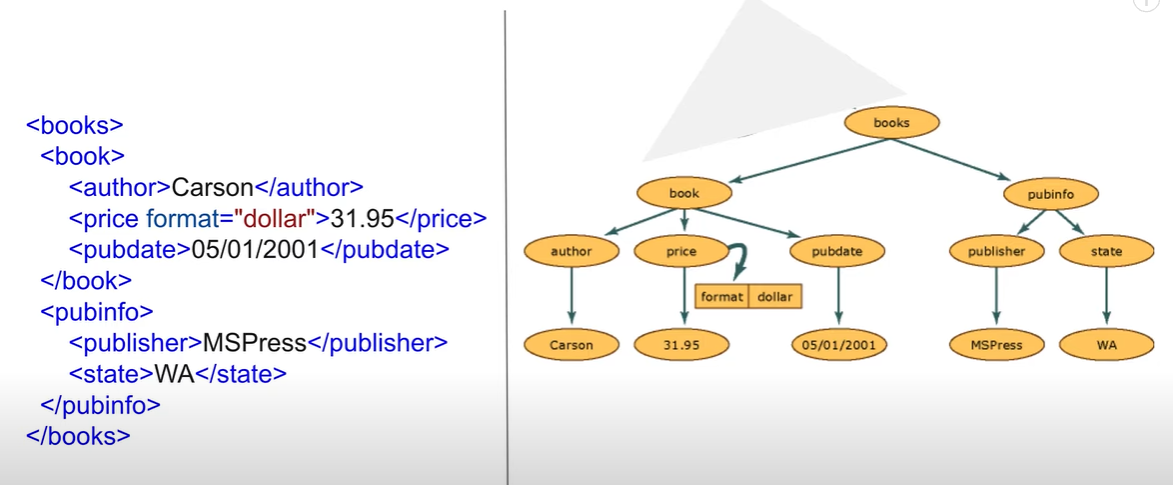
XML namespace are used for providing uniquely named elements and attributes in an XML document



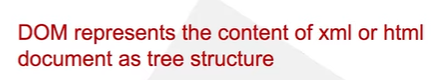
|  |  |
| --- | --- |
| Prefixes can be used in XML to resolve elements name conflicts  When using prefixes in xml, a namespace for the prefix must be defined  xmlns:prefix="URI"  The namespace can be defined by an xmlns attribute in the start tag of an element  All child elements with the same prefix are associated with the same namespace  Namespaces can also be declared in the XML root element  Default Namespaces  Defining a default namespace for an element saves us from using prefixes in all the child elements  xmlns="namespace URI"  XML namespaces are used for providing uniquely named elements and attributes in an XML document | Tiền tố có thể được sử dụng trong XML để giải quyết xung đột tên phần tử  Khi sử dụng tiền tố trong xml, một không gian tên cho tiền tố phải được xác định  xmlns: prefix = "URI"  Không gian tên có thể được xác định bởi thuộc tính xmlns trong thẻ bắt đầu của một phần tử  Tất cả các phần tử con có cùng tiền tố được liên kết với cùng một không gian tên  Không gian tên cũng có thể được khai báo trong phần tử gốc XML  Không gian tên mặc định  Việc xác định không gian tên mặc định cho một phần tử giúp chúng ta không sử dụng tiền tố trong tất cả các phần tử con  xmlns = "URI không gian tên"  Không gian tên XML được sử dụng để cung cấp các phần tử và thuộc tính được đặt tên duy nhất trong một tài liệu XML |

# 4,What is DOM | Document Object Model

Example and Tree representation



**Definition**

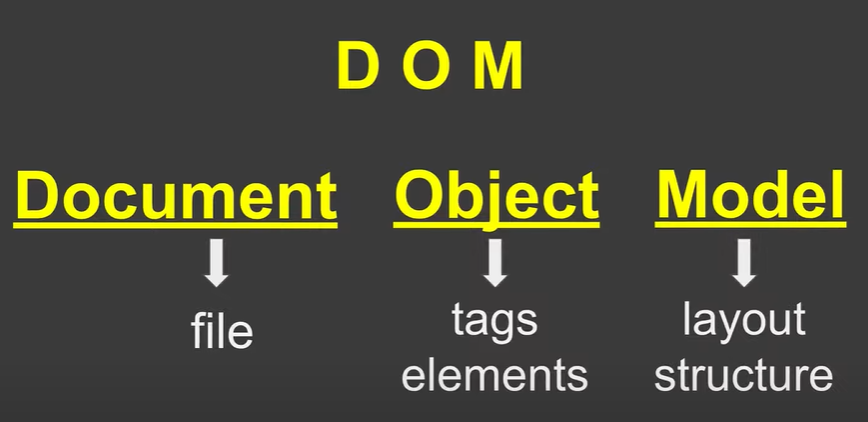
****

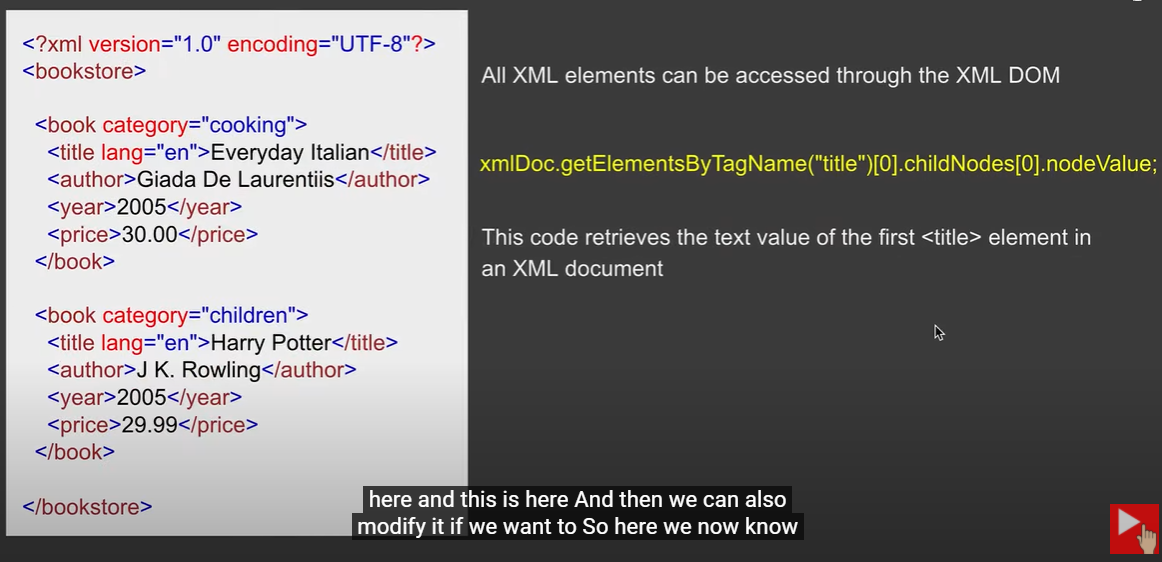
DOM: =>Represents the content of xml or html document as tree structure (Biểu diễn nội dung của tài liệu xml hoặc html dưới dạng cấu trúc cây)

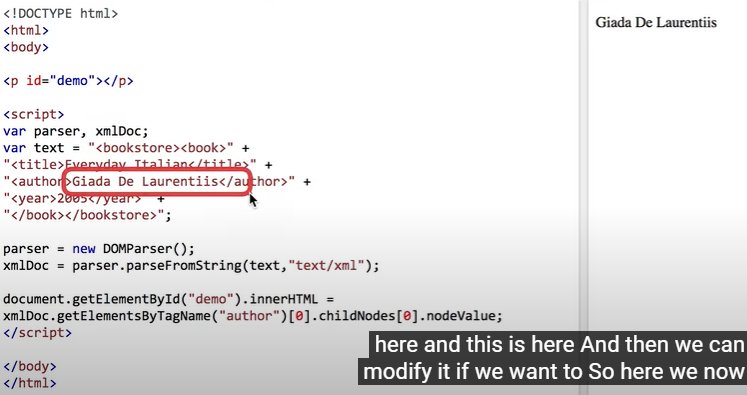


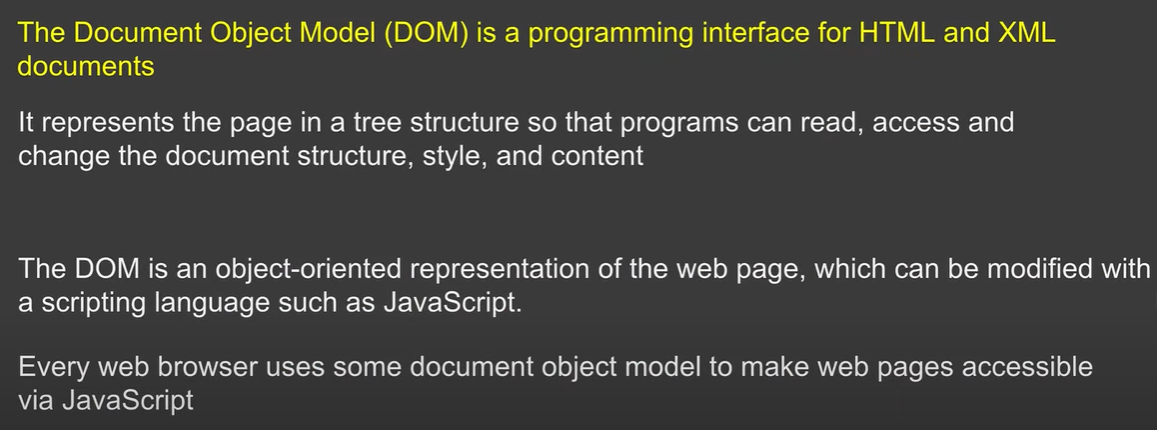
DOM: Can easily read, access, update the contents of the document (Có thể dễ dàng đọc, truy cập, cập nhật nội dung của tài liệu)

Is a programming interface (API)





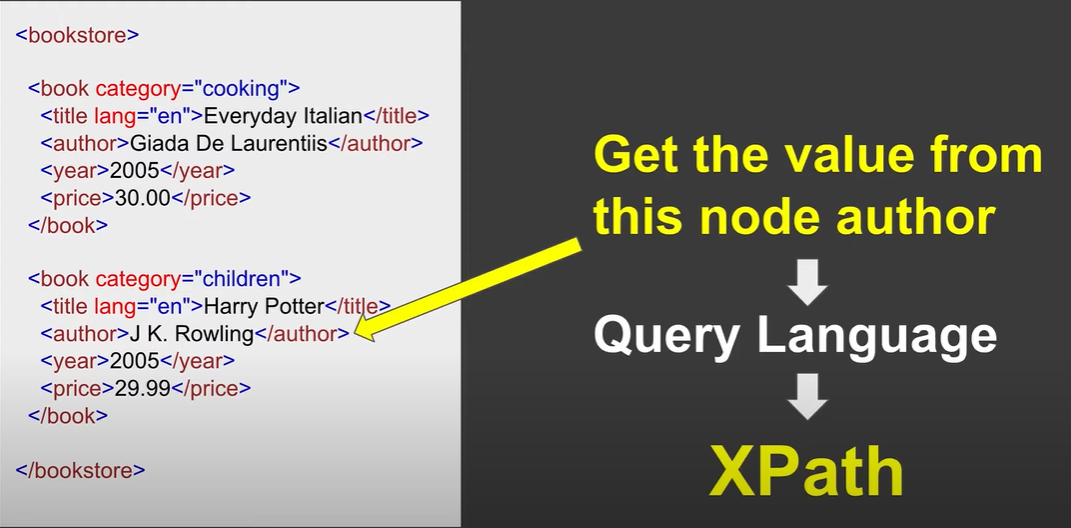
****

****

|  |  |
| --- | --- |
| Represents the content of xml or html document as tree structure  Can easily read, access, update the contents of the document  Is a programming interface (API)  All XML elements can be accessed through the XML DOM  xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;  This code retrieves the text value of the first title element in an XML document  The Document Object Model (DOM) is a programming interface for HTML and XML documents  It represents the page in a tree structure so that programs can read, access and change the document structure, style, and content  The DOM is an object-oriented representation of the web page, which can be modified with a scripting language such as JavaScript.  Every web browser uses some document object model to make web pages accessible via JavaScript | Biểu diễn nội dung của tài liệu xml hoặc html dưới dạng cấu trúc cây  Có thể dễ dàng đọc, truy cập, cập nhật nội dung của tài liệu  Là một giao diện lập trình (API)  Tất cả các phần tử XML có thể được truy cập thông qua DOM XML  xmlDoc.getElementsByTagName ("title") [0] .childNodes [0] .nodeValue;  Mã này truy xuất giá trị văn bản của phần tử tiêu đề đầu tiên trong tài liệu XML  Mô hình Đối tượng Tài liệu (DOM) là một giao diện lập trình cho các tài liệu HTML và XML  Nó đại diện cho trang trong cấu trúc cây để các chương trình có thể đọc, truy cập và thay đổi cấu trúc, kiểu và nội dung tài liệu  DOM là một đại diện hướng đối tượng của trang web, có thể được sửa đổi bằng ngôn ngữ kịch bản như JavaScript.  Mọi trình duyệt web đều sử dụng một số mô hình đối tượng tài liệu để làm cho các trang web có thể truy cập được qua JavaScript |

# 5, What is XPath | How to create XPath

**Example**

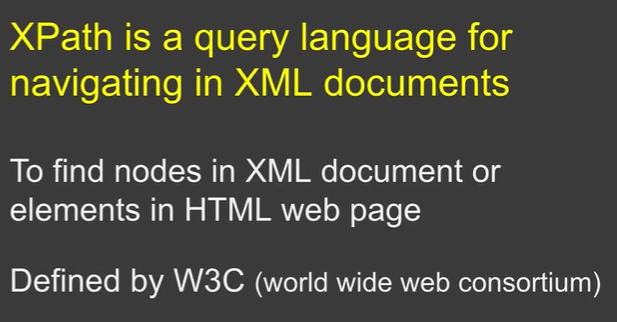
****

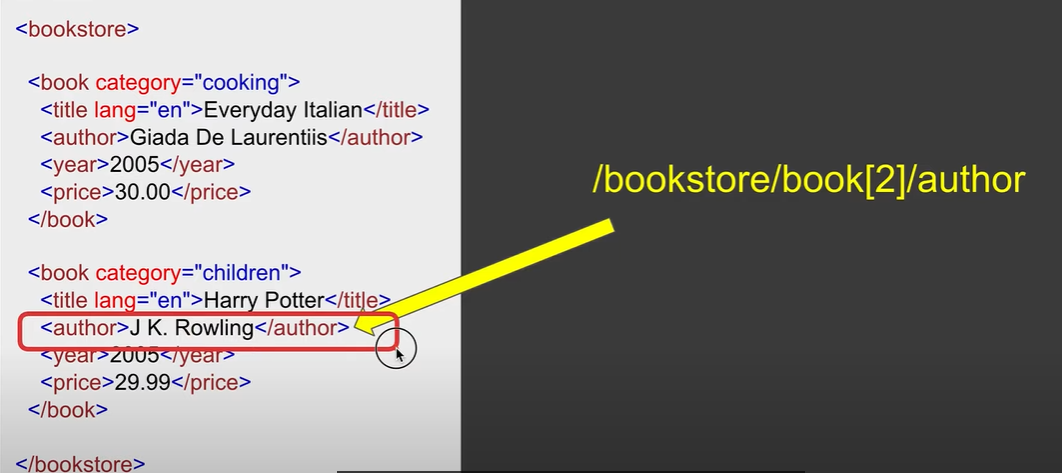
**XPath** is a query language for navigating in XML documents

(XPath là một ngôn ngữ truy vấn để điều hướng trong các tài liệu XML)

**Use**: To find nodes in XML document or elements in HTML web page

XPath is defined by W3C ( World Wide Web Consortium)



****

**Code**

|  |
| --- |
| <bookstore>      <book category="cooking">          <title lang="en"> Everyday Italian </title>          <author> Giada De Laurentiis </author>          <year>2005</year>          <price>30.00</price>      </book>      <book category="children">          <title lang="en"> Harry Potter </title>          <author> J K.Rowling </author>          <year>2005</year>          <price>29.99</price>      </book>  </bookstore> |

* **Để truy cập vào author J K.Rowling**

XPath: /bookstore/book[2]/author <=  **Absolute XPath**

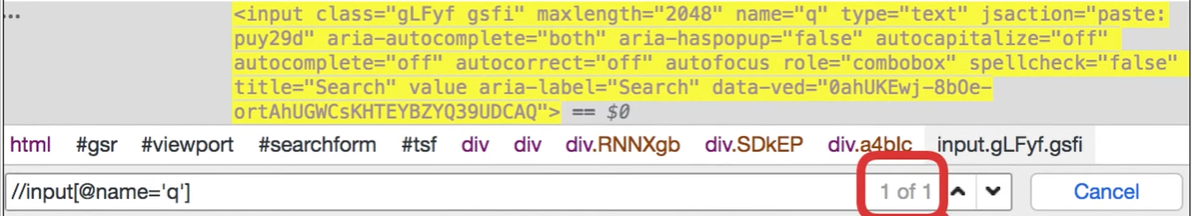
FULL XPath

|  |
| --- |
| /bookstore  /bookstore/book  /bookstore/book/@category  /bookstore/book/author  /bookstore/book/price  /bookstore/book/title  /bookstore/book/title/@lang  /bookstore/book/year  /bookstore/book[2]  /bookstore/book[2]/@category  /bookstore/book[2]/author  /bookstore/book[2]/price  /bookstore/book[2]/title  /bookstore/book[2]/title/@lang  /bookstore/book[2]/year |

**Absolute XPath:** /bookstore/book[2]/author

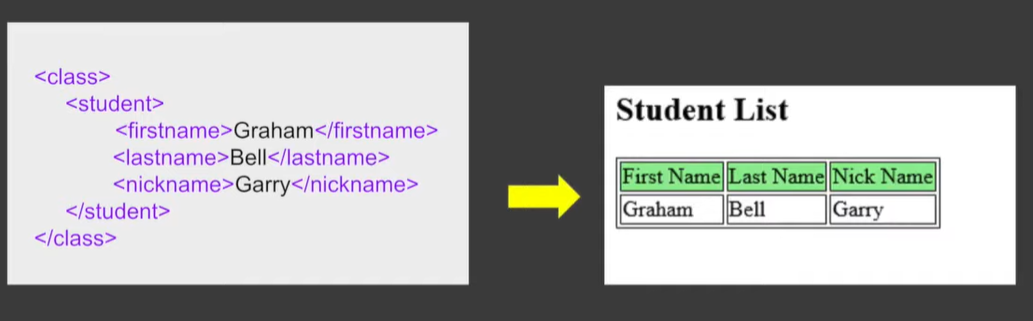
**Relative XPath:** //\*/book[2]/author

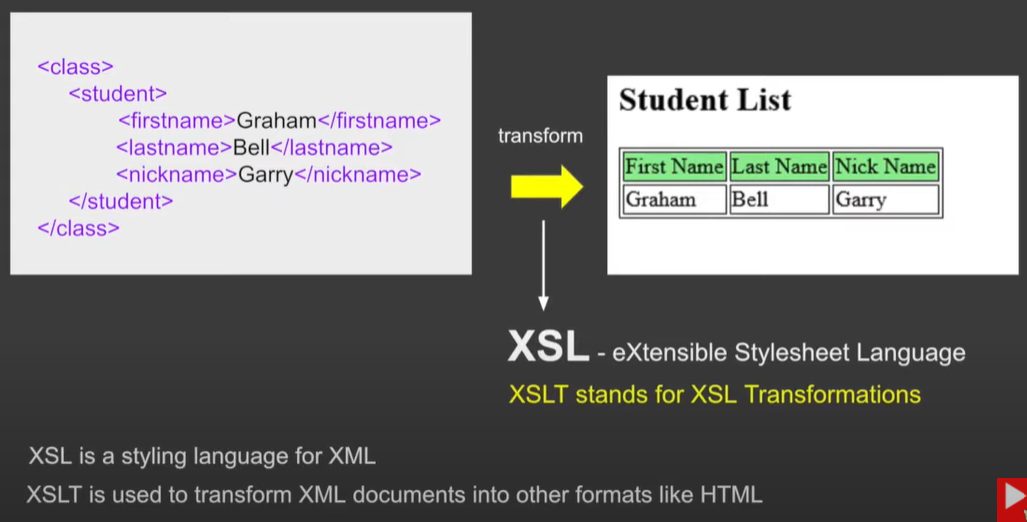
VÍ dụ với trang google



# 6, XSLT

Example -> want to table on a browser



****

**DEMO**

Code

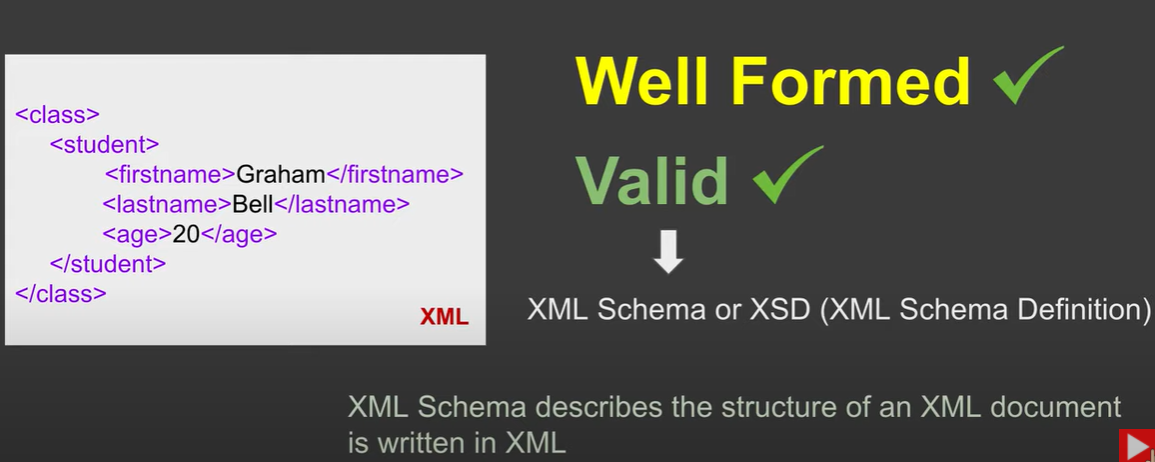
|  |
| --- |
| <class>      <student>          <firstname> Graham </firstname>          <lastname> Bell </lastname>          <nickname> Garry </nickname>      </student>  </class> |

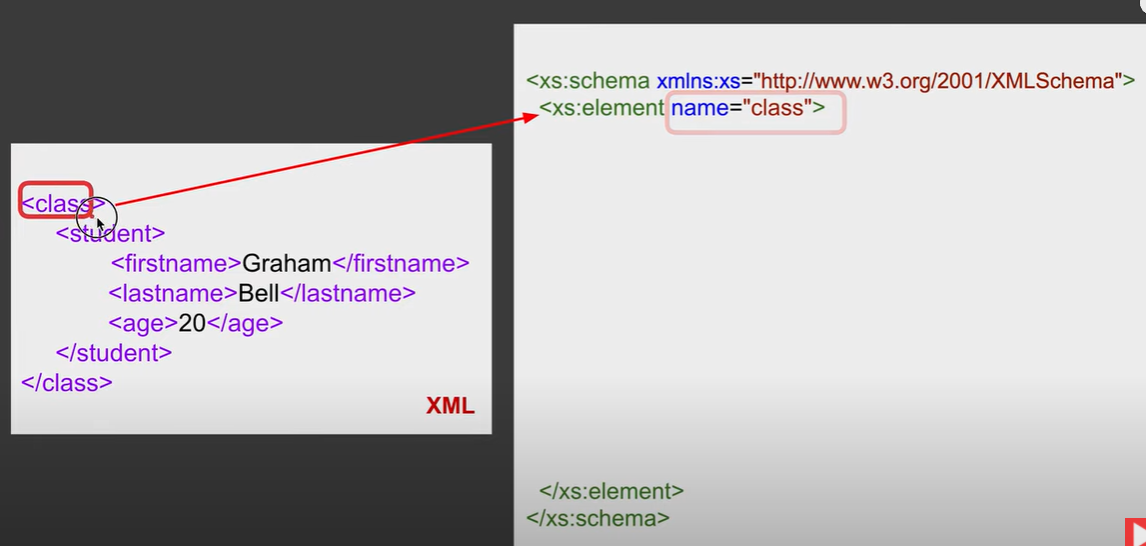
**Create the file XSLT -> draw the table on the brower HTML**

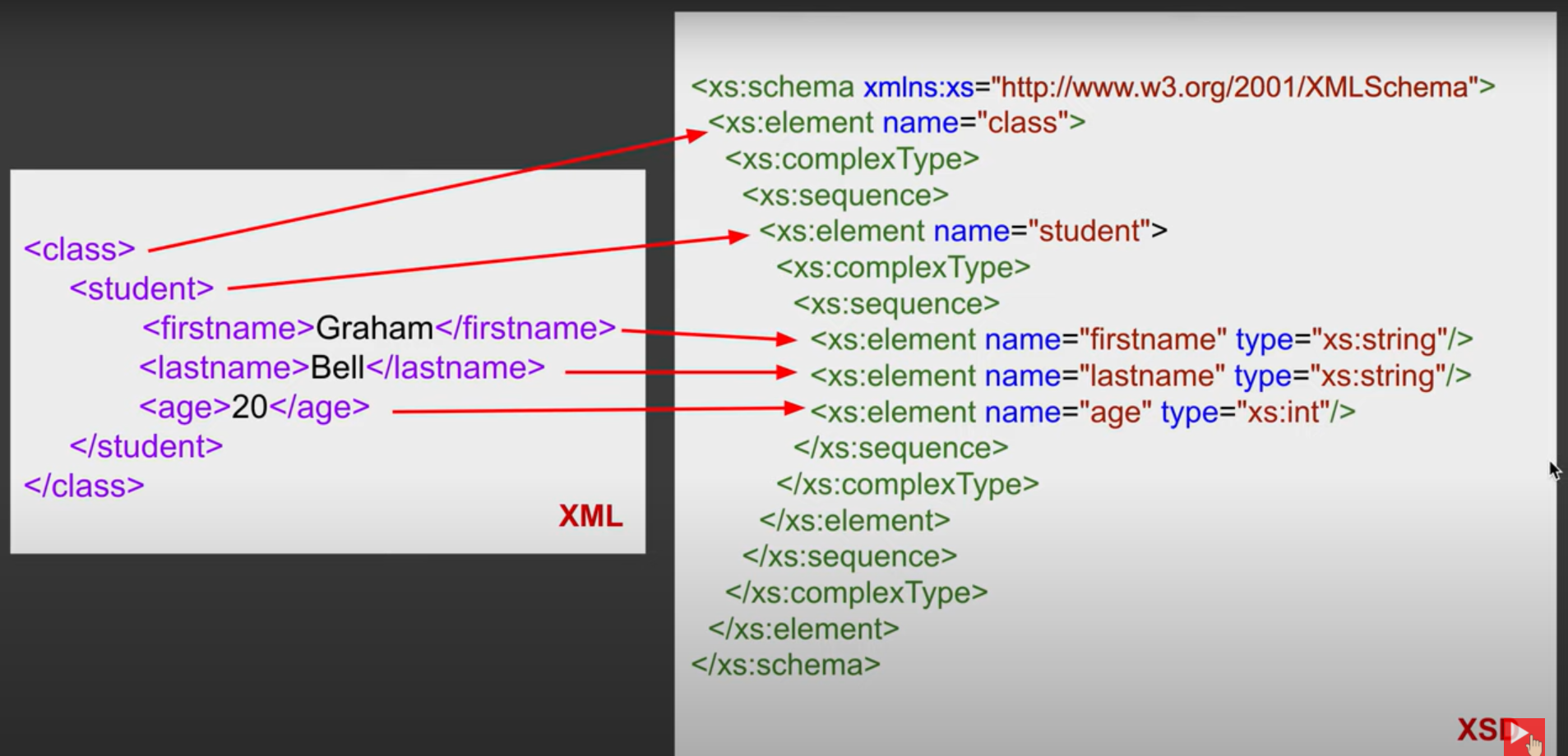
|  |
| --- |
| <xsl:stylesheet version = "1.0"  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">  <xsl:template match = "/class">      <html>          <body>              <h2> Student List </h2>              <table>                  <tr>                      <th>First Name</th>                      <th>Last Name </th>                      <th>Nick Name </th>                  </tr>              </table>          </body>      </html>  </xsl:template>  </xsl:stylesheet> |

|  |
| --- |
| <xsl:stylesheet version = "1.0"  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">  <xsl:template match = "/class">      <html>          <body>              <h2> Student List </h2>              <table border="1">                  <tr bgcolor ="lightgreen">                      <th>First Name</th>                      <th>Last Name </th>                      <th>Nick Name </th>                  </tr>                  <xsl:for-each select name="student">                      <tr>                          <td><xsl:value-of select = "firstname"/></td>                          <td><xsl:value-of select = "lastname"/></td>                          <td><xsl:value-of select = "nickname"/></td>                      </tr>                  </xsl:for-each>              </table>          </body>      </html>  </xsl:template>  </xsl:stylesheet> |

# 7, XML Schema (XSD)

****

****

****

**Code XSD**

|  |
| --- |
| <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">    <xs:element name="class">      <xs:complexType>        <xs:sequence>          <xs:element name="student">            <xs:complexType>              <xs:sequence>                <xs:element name="firstname" type="xs:string"/>                <xs:element name="lastname" type="xs:string"/>                <xs:element name="age" type="xs:int"/>              </xs:sequence>            </xs:complexType>          </xs:element>        </xs:sequence>      </xs:complexType>    </xs:element>  </xs:schema> |

|  |  |  |
| --- | --- | --- |
| <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> | **Xs:schema** | **Start tag** |
|  | xmlns:xs | **Access attribute** |
| <xs:element name="class"> |  | **Element and name of element** |
| <xs:complexType> | **Define that type of element** | **If have parent and child element** |
| <xs:sequence> | **Element occur in a specific sequence** |  |
| <xs:element name="firstname" type="xs:string"/> | name="firstname" | **Name of element** |
|  | type="xs:string" | **Type of element you allow in XML** |

**Nếu muốn có nhiều child element mà file XSD không bị lỗi thì them**

**<xs:element name=”student” maxOccurs=”unbounded” minOccurs=”0”>**

**DOM viewer online**

**XML paser**

**XPath tools online**

**Online XML Validator**