1. Tìm hiểu XML
2. Giới thiệu về XML
3. Định nghĩa

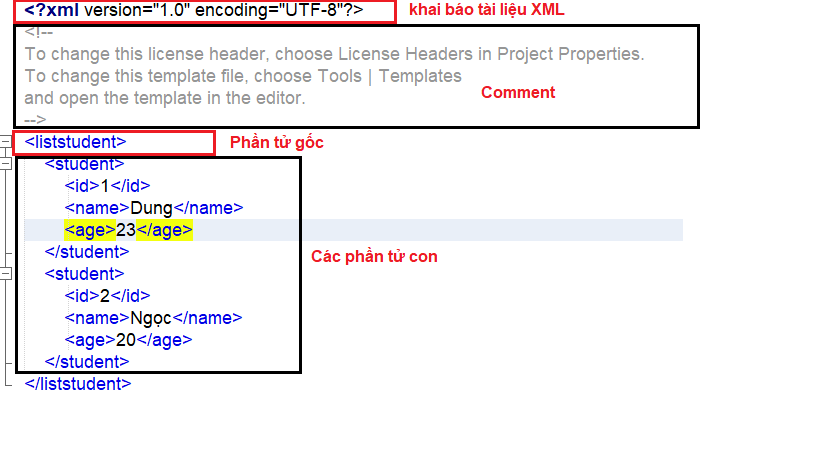
* Ngôn ngữ đánh dấu mở rộng
* XML không có thẻ riêng, người dùng có thể tạo bất kỳ thẻ nào theo ý muốn ( tuân theo quy tắc của XML)
* Các thẻ XML khá giống HTML: tag, data, attribute
* XML được xây dựng theo dạng cây, phải có tối thiểu 1 nút gốc
* XML được hỗ trợ trên các trình duyệt phổ biến hiện nay
* Phân biệt XML và HTML

| HTML | XML |
| --- | --- |
| * Đều là ngôn ngữ đánh dấu * Đều dựa trên chuẩn ngôn ngữ định dạng tổng quát SGML ( Standard Geeneralized Markup Language ) | |
| HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản | eXtensible Markup Language – ngôn ngữ đánh dấu mở rộng |
| Sử dụng tag định nghĩa sẵn, hiện có khoảng 400 tag | Tag người dùng tự định nghĩa ( theo quy định của w3c), không bị giới hạn số lượng tag sử dụng -> linh động |
| Các tag ( thẻ ) chỉ đưa ra các quy định và trình bày văn bản mà không mô tả được dữ liệu bên trong | tổ chức lưu trữ dữ liệu theo cấu trúc theo ngữ nghĩa dễ dàng giao tiếp hay trao đổi |
| Mục đích: hiển thị dữ liệu  <b>Dung</b> | Mục đích: mô tả dữ liệu  <name>Dung</name>  <content></content> |

1. XML dùng để làm gì?

* Truyền tải, trao đổi dữ liệu và liên lạc giữa các module trong cùng hệ thống hoặc giữa các ứng dụng khác nhau ( kết nối qua Internet ) không cùng nền tảng, ngôn ngữ phát triển.
* Lưu trữ thông tin nhỏ ( thông tin cấu hình, mô tả… ứng dụng, chương trình …)
* Tạo phần tóm tắt nội dung cho website (RSS)
* Tạo sơ đồ cho website (sitemap)
* Là cầu nối trao đổi dữ liệu giữa các ứng dụng web ( webservice Soap -> XML, REST -> Json )
* …

1. Quy tắc viết tài liệu XML



* Mỗi tài liệu XML phải có 1 phần tử gốc và nó là duy nhất
* Phải có thẻ đóng khi có thẻ mở
* Phải có thẻ đóng của phần tử con trước khi đóng phần tử cha
* Tên các phần tử của XML phân biệt hoa thường
* Giá trị thuộc tính đặt trong dấu nháy đôi
* Không sử dụng các giá trị đặc biệt & (&amp;) < (&lt;) > (&gt;) “ (&quot;) ‘(&apos;)
* Lưu trữ các ký hiệu đặc biệt trong XML: sử dụng CDATA

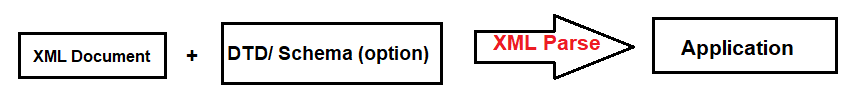
1. XML Parser

* Là một chương trình kiểm tra xem tập tin đã viết đúng cú pháp theo qui định hay chưa
* Được hỗ trợ cho người lập trình và ứng dụng để khai thác dữ liệu của XML
* Có 02 loại tất cả:

+ Hỗ trợ kiểm tra xem XML có đúng qui định định dạng của quá trình giao tiếp hay không (validation) thông qua các qui định thiết lập trong DTD hay Schema

+ Chỉ kiểm tra xem XML có viết đúng qui luật qui định hay không (kiểm tra well-formed, không kiểm tra validation)

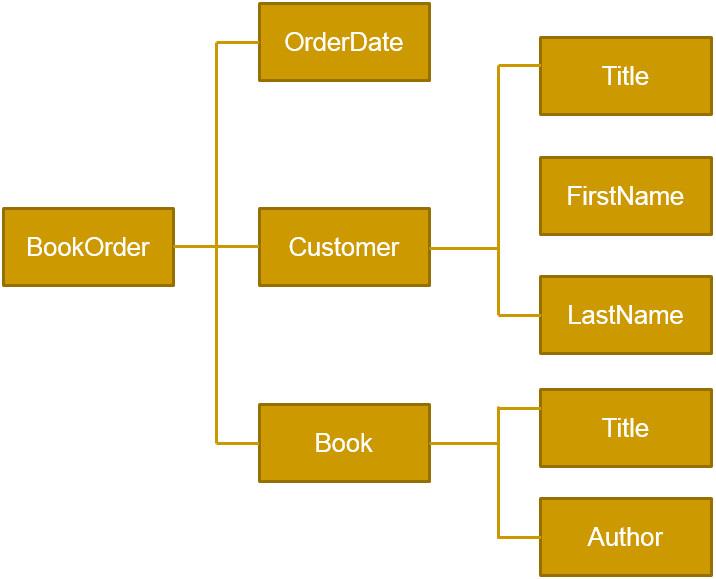
* Một tài liệu xml validation thì phải là tài liệu viết đúng cú pháp – well-formed
* Mặc định các bộ parser chỉ kiểm tra well-formed, do vậy, muốn kiểm tra validation thì bộ parser phải được thiết lập để thực hiện nội dung này



1. Cách tạo tài liệu XML hợp cấu trúc và cú pháp (XML well-formed)

* Chỉ kiểm tra xem XML có viết đúng qui luật qui định hay không
* Tài liệu XML chỉ có duy nhất một tag làm root duy nhất và sẽ chứa tất cả các node con bên trong. Nội dung này thể hiện khái niệm tương tự như mô hình cây, chỉ có một nốt root duy nhất và là cha duy nhất trong cây, sau đó là các node con bên dưới.
* Thẻ bắt đầu bằng < và kết thúc bằng >
* Bắt đầu phải bằng kí tự, hoặc dấu \_ hoặc dấu :
* Không được dùng khoảng trắng.
* Không được sử dụng từ xml làm thẻ
* Các thuộc tính của thẻ được đặt theo cú pháp: tên\_thuộc\_tính=‘giá trị’, có thể thay nháy đơn ' thành nháy kép ".

1. Bài tập: Xây dựng file Xml theo cấu trúc dữ liệu:



1. Namespace trong xml

* Vấn đề đặt ra:

Cấu trúc một tài liệu XML được xây dựng bởi người dùng, do đó họ có thể tự đặt tên thẻ XML ( element XML ) -> nếu trong một file XML, thẻ bị đặt trùng tên dẫn đến không phân biệt được thẻ nào dùng cho ứng dụng nào.

VD:

| <?xml version="1.0" encoding="UTF-8"?>  <ShopOrder>  <Order>  <Customer>  **<Title>Dung</Title>**  <Address>Thanh Hóa</Address>  </Customer>  <Product>  **<Title>Laptop HP</Title>**  <Qua>20</Qua>  <Price>15.000.000 vnđ</Price>  </Product>  </Order>  </ShopOrder> |
| --- |

* Giải pháp sử dụng **XML Namespace**
* Namespace trong xml là một phương pháp để phân biệt các phần tử trùng tên nhưng khác nhau về ý nghĩa hoặc số lượng hoặc tên phần tử con.
* Cú pháp tạo XML namespace: sử dụng tiền tố - prefix

Cú pháp khai báo: xmlns:prefix=”URI”

| <?xml version="1.0" encoding="UTF-8"?>  <ShopOrder **xmlns:cus="http://localhost/customer"**  **xmlns:product="http://localhost/product"**>  <Order>  <**cus:**Customer>  <**cus:**Title>Dung</cus:Title>  <**cus:**Address>Thanh Hóa</cus:Address>  </**cus:**Customer>  <**product:**Product>  <**product:**Title>Laptop HP</product:Title>  <**product:**Qua>20</product:Qua>  <**product:**Price>15.000.000 vnđ</product:Price>  </**product:**Product>  </Order>  </ShopOrder> |
| --- |

Cách sử dụng khai báo ở tag – thẻ: **prefix:**

**Các bước thực hiện:**

* Khai báo/ định nghĩa namespace

VD: **xmlns:cus="http://localhost/customer"**

**xmlns:product="http://localhost/product"**

* Khai báo namespace ở các tag – thẻ

VD: <**cus:**Customer>