



Bài 5. Thực hành XSD và XSLT (Tiếp)

- Mục đích, yêu cầu: Cung cấp cho sinh viên cách thức thực hành tạo lập lược đồ cấu trúc file XML đó là XSD và lập trình chuyển đổi XML trên ngôn ngữ XSLT, thông qua đó sinh viên có thể lập trình được các bài toán.
- Hình thức tổ chức dạy học: Thực hành, tự học
- Thời gian: Thực hành (trên lớp: 3; online: 0) Tự học, tự nghiên cứu: 2
- Nội dung chính:

Thực hành XSD và XSLT

1. Tổng quan về XSLT

a. XSL là viết tắt của eXtensible Stylesheet Language. Được sử dụng để chuyển đổi từ tài liệu xml sang một dạng tài liệu khác, ví dụ như html.

b. Sự cần thiết của xsl

Trong tài liệu html, các thẻ đã được định nghĩa trước như table, div, span...; và trình duyệt biết cách áp dụng các style cho chúng và hiển thị theo cách css đã viết. Nhưng với tài liệu xml, các thẻ không được định nghĩa trước nên để hiểu và định dạng tài liệu xml, tổ chức W3C đã phát triển ra ngôn ngữ XSL, là ngôn ngữ dựa trên Stylesheet Language. Một tài liệu XSL sẽ chỉ ra cách trình duyệt tạo và hiển thị một tài liệu xml như thế nào.

XSL có 3 phần chính:

- + XSLT – được dùng để chuyển tài liệu XML sang một dạng tài liệu khác
- + XPath – được dùng để định hướng tới các phần tử trong tài liệu XML
- + XSL-FO – được dùng để định dạng tài liệu XML.

c. XSLT là gì

XSLT – là viết tắt của Extensible Stylesheet Language Transformations, cung cấp khả năng tự động chuyển đổi tài liệu XML sang một dạng tài liệu khác.

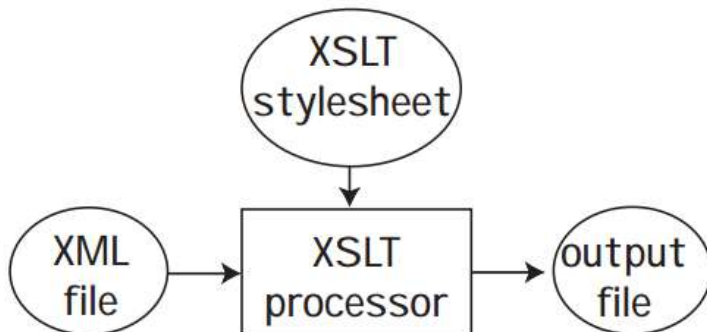
d. XSLT làm việc như thế nào

Một bảng kiểu XSLT được sử dụng để định nghĩa các nguyên tắc biến đổi được áp

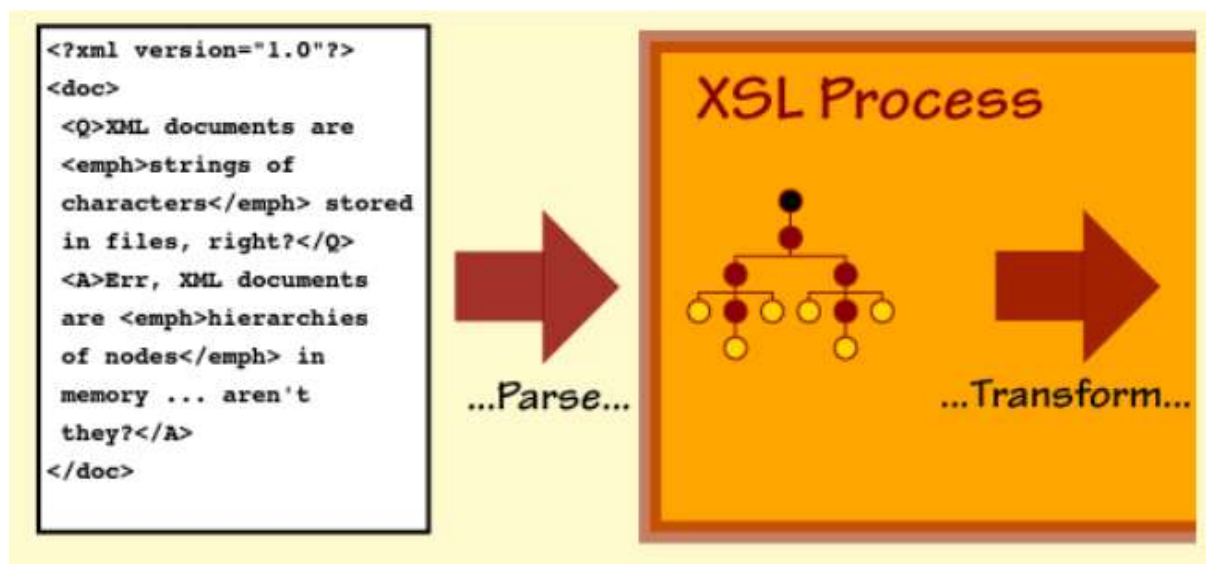


dụng cho tài liệu XML. Các lệnh của XSLT được viết theo nguyên tắc XML.

Structure of an XSLT System

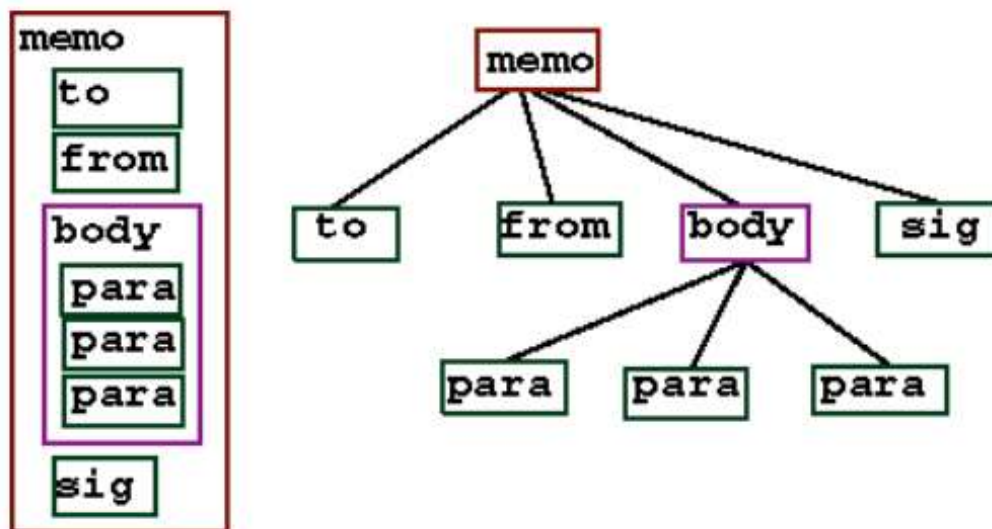


XML hoạt động theo cấu trúc hình cây





* Tài liệu XML là 1 cây:



* Ưu điểm của XSLT:

- Độc lập với lập trình. Sự biến đổi được viết riêng ra file .xsl
- Muốn thay đổi kết quả thì chỉ cần thay đổi trong file .xsl

2. Cú pháp XSLT

Ví dụ, với file **students.xml** như sau:



```
<?xml version="1.0"?>
<class>
  <student rollno="393">
    <firstname>Dinkar</firstname>
    <lastname>Kad</lastname>
    <nickname>Dinkar</nickname>
    <marks>85</marks>
  </student>
  <student rollno="493">
    <firstname>Vaneet</firstname>
    <lastname>Gupta</lastname>
    <nickname>Vinni</nickname>
    <marks>95</marks>
  </student>
  <student rollno="593">
    <firstname>Jasvir</firstname>
    <lastname>Singh</lastname>
    <nickname>Jazz</nickname>
    <marks>90</marks>
  </student>
</class>
```

Chúng ta cần định nghĩa tài liệu XSLT cho tài liệu XML trên theo các yêu cầu sau:

- + Trang có tiêu đề là Students
- + Trang có chứa bảng hiển thị chi tiết Student
- + Các cột có tiêu đề lần lượt là: Roll No, First Name, Last Name, Nick Name, Marks.
- + Bảng phải chứa chi tiết thông tin của Student.

Bước 1. Tạo file XSLT, đặt tên là **student.xsl** và ghi cùng vị trí với file student.xml.

File students.xsl sẽ có nội dung sau:



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0"
3      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4
5      <xsl:template match="/">
6          <!-- HTML tags. Used for formatting purpose.
7          Processor will skip them and browser will simply render them.
8          -->
9          <html>
10             <body>
11                 <h2>Students</h2>
12                 <table border="1">
13                     <tr bgcolor="#9acd32">
14                         <th>Roll No</th>
15                         <th>First Name</th>
16                         <th>Last Name</th>
17                         <th>Nick Name</th>
18                         <th>Marks</th>
19                     </tr>
20                     <!-- for-each processing instruction
21                     Looks for each element matching the XPath expression
22                     -->
23                     <xsl:for-each select="class/student">
24                         <tr>
25                             <td><xsl:value-of select="@rollno"/></td>
26                             <td><xsl:value-of select="firstname"/></td>
27                             <td><xsl:value-of select="lastname"/></td>
28                             <td><xsl:value-of select="nickname"/></td>
29                             <td><xsl:value-of select="marks"/></td>
30                         </tr>
31                     </xsl:for-each>
32                 </table>
33             </body>
34         </html>
35     </xsl:template>
36 </xsl:stylesheet>
    
```

Bước 2. Tạo liên kết từ xml tới file xslt: thêm dòng `<?xml-stylesheet ... ?>` vào vị trí dòng 2 như hình dưới.

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="students.xsl"?>
<class>
    
```

Kết quả hiển thị như sau : (chú ý chọn trình duyệt là FireFox hoặc Internet Explorer)



Students

Roll No	First Name	Last Name	Nick Name	Marks
393	Dinkar	Kad	Dinkar	85
493	Vaneet	Gupta	Vinni	95
593	Jasvir	Singh	Jazz	90



3. XSLT - <xsl: template ...>

<xsl:template> : tìm và so sánh các phần tử xem có khớp không.

Trong ví dụ trên, ở dòng số 5 : `<xsl:template match="/">` : tìm và so sánh với phần tử gốc.

4. XSLT - <xsl: value-of ...>

Được dùng để hiển thị giá trị tại node đang được chọn bởi biểu thức XPath.

```
<td><xsl:value-of select="@rollno"/></td>
<td><xsl:value-of select="firstname"/></td>
<td><xsl:value-of select="lastname"/></td>
<td><xsl:value-of select="nickname"/></td>
<td><xsl:value-of select="marks"/></td>
```

Lưu ý: Muốn lấy ra giá trị của thuộc tính, thêm @ trước tên của nó.

5. XSLT – <xsl: for – each ...>

Thẻ <xsl:for-each> sử dụng vòng lặp để áp dụng template cho mỗi node.

```
<xsl:for-each select="class/student">
```

6. XSLT – <xsl: sort ...>

Thẻ <xsl:sort> được dùng để sắp xếp các node theo điều kiện nào đó.

```
<xsl:for-each select="class/student">
  <xsl:sort select="firstname"/>
  <tr>
```

Bổ sung <xsl:sort> vào file students.xml như hình trên, kết quả hiển thị là các student sẽ được sắp xếp theo chiều tăng dần của firstname.

Chưa sắp xếp:



Students

Roll No	First Name	Last Name	Nick Name	Marks
393	Dinkar	Kad	Dinkar	85
493	Vaneet	Gupta	Vinni	95
593	Jasvir	Singh	Jazz	90

Đã sắp xếp:

Students

Roll No	First Name	Last Name	Nick Name	Marks
393	Dinkar	Kad	Dinkar	85
593	Jasvir	Singh	Jazz	90
493	Vaneet	Gupta	Vinni	95

7. XSLT – <xsl:if ...>

Thẻ <xsl:if> Kiểm tra dữ liệu có thỏa mãn điều kiện nào đó.

Tiếp ví dụ trên, chỉ những student có mark > 90 mới được hiển thị:

```
<xsl:for-each select="class/student">
  <xsl:if test="marks > 90">
    <tr>
      <td><xsl:value-of select="@rollno"/></td>
      <td><xsl:value-of select="firstname"/></td>
      <td><xsl:value-of select="lastname"/></td>
      <td><xsl:value-of select="nickname"/></td>
      <td><xsl:value-of select="marks"/></td>
    </tr>
  </xsl:if>
```

Kết quả:



Students

Roll No	First Name	Last Name	Nick Name	Marks
493	Vaneet	Gupta	Vinni	95

Các toán tử so sánh là

= : bằng

!= : không bằng

< : nhỏ hơn (<)

> : lớn hơn (>)

Ví dụ `<xsl:if test="price < 10">` // \rightarrow `<xsl:if test="price<10">`

8. XSLT – `<xsl:choose>`

Thẻ `<xsl:choose>` cho phép đưa vào nhiều điều kiện để thay đổi hiển thị hay định dạng phần tử xml.

Thêm dòng 15 để bổ sung cột Grade và thêm dòng từ 24-36 như hình dưới:



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0"
3      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4      <xsl:template match="/">
5          <html>
6              <body>
7                  <h2>Students</h2>
8                  <table border="1">
9                      <tr bgcolor="#9acd32">
10                         <th>Roll No</th>
11                         <th>First Name</th>
12                         <th>Last Name</th>
13                         <th>Nick Name</th>
14                         <th>Marks</th>
15                         <th>Grade</th>
16                      </tr>
17                      <xsl:for-each select="class/student">
18                          <tr>
19                              <td><xsl:value-of select="@rollno"/></td>
20                              <td><xsl:value-of select="firstname"/></td>
21                              <td><xsl:value-of select="lastname"/></td>
22                              <td><xsl:value-of select="nickname"/></td>
23                              <td><xsl:value-of select="marks"/></td>
24                              <td>
25                                  <xsl:choose>
26                                      <xsl:when test="marks > 90">
27                                          High
28                                      </xsl:when>
29                                      <xsl:when test="marks > 85">
30                                          Medium
31                                      </xsl:when>
32                                      <xsl:otherwise>
33                                          Low
34                                      </xsl:otherwise>
35                                  </xsl:choose>
36                              </td>
37                          </tr>
38                      </xsl:for-each>
39                  </table>
40              </body>
41          </html>
42      </xsl:template>
43  </xsl:stylesheet>

```

Kết quả là:

Tạo thêm cột **Grade**, với kết quả là Low, High hay Medium tùy theo giá trị của mark.



Students

Roll No	First Name	Last Name	Nick Name	Marks	Grade
393	Dinkar	Kad	Dinkar	85	Low
493	Vaneet	Gupta	Vinni	95	High
593	Jasvir	Singh	Jazz	90	Medium



10. XSLT – <xsl:apply - template>

Thẻ <xsl:apply-template> dùng để tìm template phù hợp và áp dụng cho mỗi node.

File students.xsl được viết lại như sau:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0"
3      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4      <xsl:template match="/">
5          <html>
6              <body>
7                  <h2>Students</h2>
8                  <xsl:apply-templates select="class/student" />
9              </body>
10             </html>
11         </xsl:template>
12
13         <xsl:template match="class/student">
14             <xsl:apply-templates select="@rollno" />
15             <xsl:apply-templates select="firstname" />
16             <xsl:apply-templates select="lastname" />
17             <xsl:apply-templates select="nickname" />
18             <xsl:apply-templates select="marks" />
19             <br />
20         </xsl:template>
```



```

22 <xsl:template match="@rollno">
23     <span style="font-size:22px;">
24         <xsl:value-of select="." />
25     </span><br />
26 </xsl:template>
27
28 <xsl:template match="firstname">
29     First Name:<span style="color:blue;">
30         <xsl:value-of select="." />
31     </span><br />
32 </xsl:template>
33
34 <xsl:template match="lastname">
35     Last Name:<span style="color:green;">
36         <xsl:value-of select="." />
37     </span><br />
38 </xsl:template>
39
40 <xsl:template match="nickname">
41     Nick Name:<span style="color:red;">
42         <xsl:value-of select="." />
43     </span><br />
44 </xsl:template>
45
46 <xsl:template match="marks">
47     Marks:<span style="color:gray;">
48         <xsl:value-of select="." />
49     </span><br />
50 </xsl:template>
51
52 </xsl:stylesheet>

```



Kết quả thực hiện:

Students

393

First Name:Dinkar

Last Name:Kad

Nick Name:Dinkar

Marks:85

493

First Name:Vaneet

Last Name:Gupta

Nick Name:Vinni

Marks:95

593

First Name:Jasvir

Last Name:Singh

Nick Name:Jazz

Marks:90

11. Một số ví dụ

Ví dụ 1: Giả sử có file lophoc.xml có cấu trúc sau:



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <?xml-stylesheet type="text/xsl" href="lophoc.xsl"?>
3  <lophoc>
4    <sinhvien>
5      <hoten>Khanh</hoten>
6      <diem>7</diem>
7      <diachi>Ha Noi</diachi>
8    </sinhvien>
9    <sinhvien>
10     <hoten>Tuan</hoten>
11     <diem>6</diem>
12     <diachi>Ha Noi</diachi>
13   </sinhvien>
14   <sinhvien>
15     <hoten>Phuong</hoten>
16     <diem>3</diem>
17     <diachi>Ha Noi</diachi>
18   </sinhvien>
19   <sinhvien>
20     <hoten>Van Anh</hoten>
21     <diem>7</diem>
22     <diachi>Ha Noi</diachi>
23   </sinhvien>
24 </lophoc>

```

Yêu cầu: Tạo file lophoc.xsl để hiển thị lớp học:

- ✓ Lớp học nằm trong 1 bảng, mỗi sinh viên là 1 hàng.
- ✓ Mỗi hàng có thông tin chi tiết của 1 sinh viên
- ✓ Thêm cột Xếp loại và dữ liệu là Giỏi, Khá, Trung bình, Kém dựa trên điểm.
- ✓ Cuối bảng có thống kê số lượng sinh viên và tổng điểm của toàn bộ sinh viên.



File **lophoc.xsl** được viết như sau:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
4      <xsl:output method="html"/>
5      <xsl:template match="lophoc">
6          <html>
7              <head>
8                  <title>Vi du lop hoc</title>
9              </head>
10             <body>
11                 <h2>Danh sach sinh vien</h2>
12                 <table border="1">
13                     <tr>
14                         <th>Thu tu</th>
15                         <th>Ho ten</th>
16                         <th>Diem</th>
17                         <th>Dia chi</th>
18                         <th>Xep loai</th>
19                     </tr>
20                     <xsl:apply-templates select="sinhvien"/>
21                 </table>
22                 <br/>
23                 So luong sinh vien: <xsl:value-of select="count(sinhvien/diem)"/>
24                 <br/>
25                 Tong diem cua sinh vien: <xsl:value-of select="sum(sinhvien/diem)"/>
26                 <br/>
27             </body>
28         </html>
29     </xsl:template>
30
31     <xsl:template match="sinhvien">
32         <tr>
33             <td><xsl:number value="position()" format="1."/></td>
34             <td><xsl:value-of select="hoten"/> </td>
35             <td><xsl:value-of select="diem"/></td>
36             <td><xsl:value-of select="diachi"/></td>
37         <td>
38             <xsl:choose>
39                 <xsl:when test="diem >= 8">Gioi</xsl:when>
40                 <xsl:when test="diem >= 7">Kha</xsl:when>
41                 <xsl:when test="diem >= 5">Trung binh</xsl:when>
42                 <xsl:otherwise>Kem</xsl:otherwise>
43             </xsl:choose>
44         </td>
45     </tr>
46 </xsl:template>
47 </xsl:stylesheet>

```




Lưu ý: File trên sử dụng apply-templates select="sinhvien" nên không cần sử dụng vòng lặp để lấy ra các phần tử là sinhvien.

Kết quả là:

Danh sách sinh viên

Thu tu	Ho ten	Diem	Địa chỉ	Xếp loại
1.	Khanh	7	Ha Noi	Kha
2.	Tuan	6	Ha Noi	Trung bình
3.	Phuong	3	Ha Noi	Kem
4.	Van Anh	7	Ha Noi	Kha

Số lượng sinh viên: 4

Tổng điểm của sinh viên: 23

Ví dụ 2: Trường hợp file lophoc2.xml có nhiều cấp:



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <lophoc>
3    <sinhvien>
4      <masv>201812365</masv>
5      <hoten>Van Tuan</hoten>
6      <diachi>Hanoi</diachi>
7      <diemcacmon>
8        <cnxml>9</cnxml>
9        <java>7</java>
10       <csapr>8</csapr>
11     </diemcacmon>
12   </sinhvien>
13   <sinhvien>
14     <masv>201812366</masv>
15     <hoten>Lan Huong</hoten>
16     <diachi>Hanoi</diachi>
17     <diemcacmon>
18       <cnxml>7</cnxml>
19       <java>9</java>
20       <csapr>7</csapr>
21     </diemcacmon>
22   </sinhvien>
23   <sinhvien>
24     <masv>201812367</masv>
25     <hoten>Mai Anh</hoten>
26     <diachi>Hanoi</diachi>
27     <diemcacmon>
28       <cnxml>5</cnxml>
29       <java>8</java>
30       <csapr>8</csapr>
31     </diemcacmon>
32   </sinhvien>
33 </lophoc>

```

File lophoc2.xsl có nội dung sau:



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
3      <xsl:output method="html"/>
4      <xsl:template match="lophoc">
5          <html>
6              <head>
7                  <title>Vi du lop hoc 2</title>
8              </head>
9              <body>
10                 <h2>Danh sach sinh vien</h2>
11                 <table border="1">
12                     <tr>
13                         <th>Thu tu</th>
14                         <th>Ma sinh vien</th>
15                         <th>Ho ten</th>
16                         <th>Dia chi</th>
17                         <th>CN XML</th>
18                         <th>Java</th>
19                         <th>C shapr</th>
20                     </tr>
21                     <xsl:apply-templates select="sinhvien"/>
22                 </table>
23             </body>
24         </html>
25     </xsl:template>
26
27     <xsl:template match="sinhvien">
28         <tr>
29             <td><xsl:number value="position()" format="1."/></td>
30             <td><xsl:value-of select="masv"/></td>
31             <td><xsl:value-of select="hoten"/></td>
32             <td><xsl:value-of select="diachi"/></td>
33             <td align="center"><xsl:value-of select="diemcacmon/cnxml"/></td>
34             <td align="center"><xsl:value-of select="diemcacmon/java"/></td>
35             <td align="center"><xsl:value-of select="diemcacmon/cshapr"/></td>
36         </tr>
37     </xsl:template>
38 </xsl:stylesheet>
39

```

Kết quả là:

Danh sach sinh vien

Thu tu	Ma sinh vien	Ho ten	Dia chi	CN XML	Java	C shapr
1.	201812365	Van Tuan	Hanoi	9	7	8
2.	201812366	Lan Huong	Hanoi	7	9	7
3.	201812367	Mai Anh	Hanoi	5	8	8



Tài liệu tham khảo:

- Sách, giáo trình chính:

- [1]. Giáo trình dịch vụ web và ứng dụng/ Đại học KHTN – Đại học Quốc gia TP HCM, 2016.
- [2]. Anura Guruge. Web service: Theory and Practice. Elsevier press 20017.
- [3]. Scott Klein. Professional API- RESTfull Programming. Wiley Publishing, 2016.

- Sách, tài liệu tham khảo:

- [1]. Alex Ferra, Mathew MacDonald. Programming .NET web services. O'Reilly Media, 2012.
- [2]. Mark D. Hansen. SOA Using .NET web services. Prentice Hall, 2011.