



## Bài 3. XSLT

- Mục đích, yêu cầu: Cung cấp cho sinh viên kiến thức về ngôn ngữ XSLT trong XML dùng để chuyển đổi file XML sang một định dạng khác. Thông qua đó sinh viên có thể chuyển đổi định dạng các file XML sang HTML và các dạng khác.
- Hình thức tổ chức dạy học: Lý thuyết, trực tuyến + tự học
- Thời gian: Lý thuyết (trên lớp: 0; online: 4) Tự học, tự nghiên cứu: 8
- Nội dung chính:

### XSLT

#### 1. - Giới thiệu

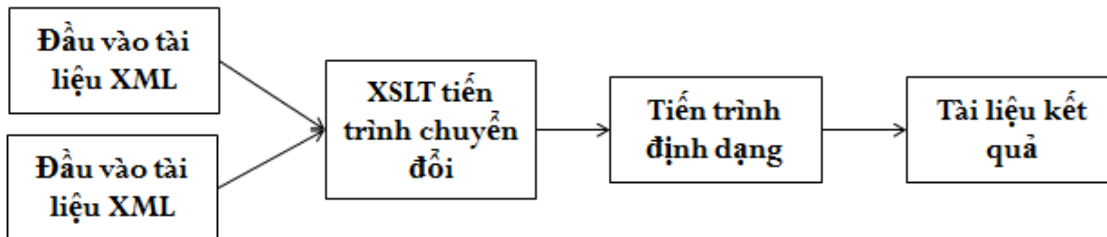
- CSS – trình bày 1 tài liệu HTML
- CSS giúp
  - Tính ẩn/hiện của 1 phần tử
  - Vị trí và kích thước của 1 phần tử
  - Màu và nền
  - Font và văn bản
  - Khoảng cách
- XSL trình bày 1 tài liệu XML
- XSL được tạo nên bởi 2 phần
  - Một bộ từ vựng XML chỉ rõ các ngữ nghĩa định dạng
  - Một ngôn ngữ chuyển đổi các tài liệu XML đó là XSLT để chuyển đổi từ XML sang định dạng khác như HTML, Text

#### 2. - XSL – Extension Style Languages

- XSL là 1 ứng dụng bảng định kiểu được tạo ra để đi kèm với XML
- XSL hỗ trợ các tính năng sau
  - Cung cấp 1 ngôn ngữ chuyển đổi XSLT
  - XSLT được sử dụng để chuyển đổi các tài liệu được viết trong 1 XML DTD sang 1 DTD khác
  - XSL được sử dụng như 1 ngôn ngữ định dạng
  - XSL được sử dụng để trình bày 1 tài liệu XML
  - XSL có thể được sử dụng để sắp xếp và sàng lọc
  - XSL có thể được sử dụng để khớp mẫu (pattern matching) cho phép tìm kiếm bản ghi



- Tính năng Xpath được thêm vào XSLT, nó là 1 ngôn ngữ biểu diễn đơn giản
- Hai nhánh của XSL là dịch và định dạng được thực hiện theo thứ tự đầu tiên là việc dịch sau đó là định dạng
- Các file đầu vào và đầu ra có thể ở dạng bất kỳ chứ không nhất thiết là 1 file XML hợp khuôn dạng
- Các file đầu ra có thể được kiểm soát qua phần tử `xsl:text` để có được đầu ra tùy ý
- Một XSL giống với 1 tài liệu XML gồm các phần tử, các node, các thuộc tính,... Phần tử `xsl:stylesheet` là cây gốc của cây XML cho XSL stylesheet. Có 1 số thành phần con của `xsl:stylesheet` như:
  - `xsl:import`, `xsl:include`, `xsl:key`, `xsl:local`,...
- Một bộ xử lý XSL nhận 1 tài liệu XML và phân tích nó theo bảng định kiểu của XSL



### 3 -Chuyển đổi – Transformation

- Xét ví dụ sau: Thông tin một số ngôi sao bóng đá

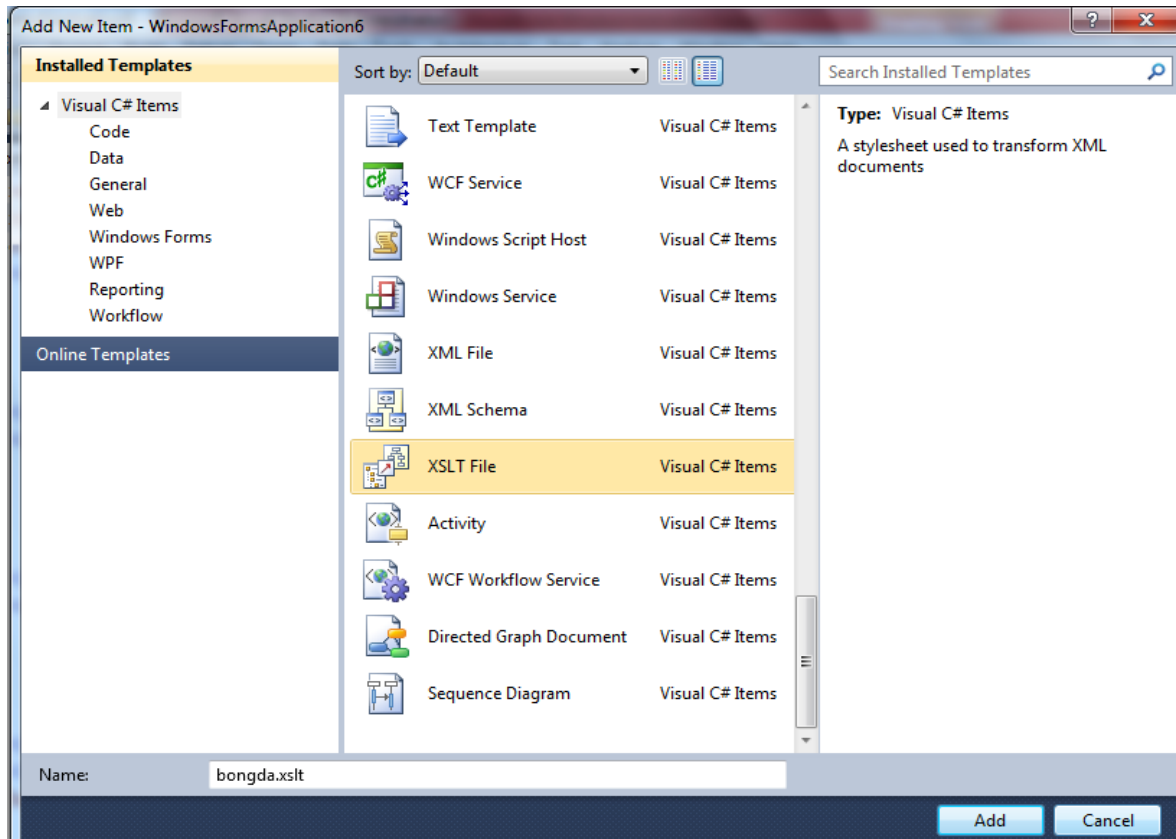


```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type="text/xsl" href="bongda.xslt"?>
<bongda>
  <cauthu>
    <hoten>David Beckham</hoten>
    <ghiban>250</ghiban>
    <sosut>300</sosut>
    <cohoi>70</cohoi>
  </cauthu>
  <cauthu>
    <hoten>Jose Mourinho</hoten>
    <ghiban>0</ghiban>
    <sosut>250</sosut>
    <cohoi>490</cohoi>
  </cauthu>
  <cauthu>
    <hoten>Donal Trump</hoten>
    <ghiban>200</ghiban>
    <sosut>250</sosut>
    <cohoi>49</cohoi>
  </cauthu>
  <!--vân vân-->
</bongda>
```

- Thông tin về các cầu thủ không theo 1 thứ tự nào cả
- Dòng bôi đen liên kết tới file bongda.xslt giống như HTML tới CSS
- Tạo file bongda.xslt để hiển thị file XML ra HTML theo khuôn dạng

### Trên Visual Studio (C#)

- Click chuột phải lên Solution Explorer –Add – New Item



- Xuất hiện mẫu xslt

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl"
>
  <xsl:output method="xml" indent="yes"/><!--Method - Kết quả chuyển đổi-->

  <xsl:template match="@* | node()">
    <!--Nội dung chuyển đổi-->
    <xsl:copy>
      <xsl:apply-templates select="@* | node()"/>
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

- Sửa file xslt trình bày file XML vừa tạo ra HTML

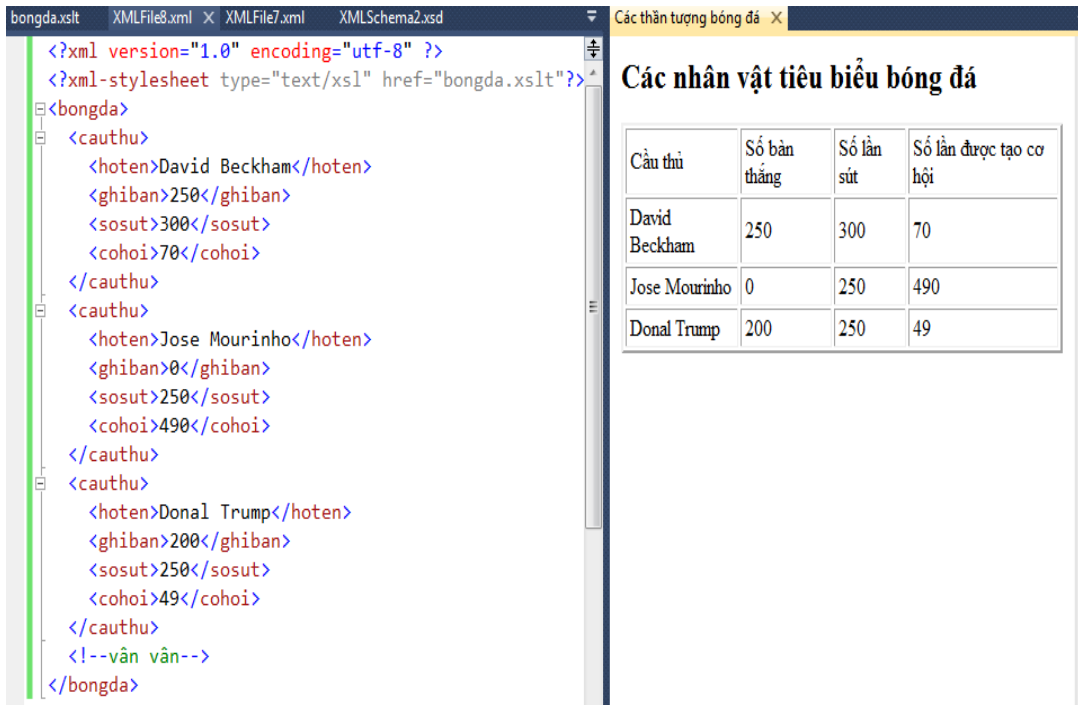


```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl"
>
  <xsl:output method="html" indent="yes"/><!--Method - Kết quả chuyển đổi-->

  <xsl:template match="/">
    <!--Nội dung chuyển đổi-->
    <html>
      <head>
        <title>Các thần tượng bóng đá</title>
      </head>
      <body>
        <h2>Các nhân vật tiêu biểu bóng đá</h2>
        <table border="2" cellpadding="3">
          <tr>
            <td>Cầu thủ</td>
            <td>Số bàn thắng</td>
            <td>Số lần sút</td>
            <td>Số lần được tạo cơ hội</td>
          </tr>
          <xsl:for-each select="bongda/cauthu">
            <tr>
              <td>
                <xsl:value-of select="hoten"/>
              </td>
              <td>
                <xsl:value-of select="ghiban"/>
              </td>
              <td>
                <xsl:value-of select="sosut"/>
              </td>
              <td>
                <xsl:value-of select="cohoi"/>
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

- **Kết quả debug**

- Chọn file XML cần chuyển đổi – XML start xslt Debugging (Alt+F5)



The screenshot shows an IDE with two panes. The left pane displays XML code for a football player database. The right pane shows the XSL transformation result, which is a table titled "Các nhân vật tiêu biểu bóng đá" (Famous Football Figures).

**XML Code:**

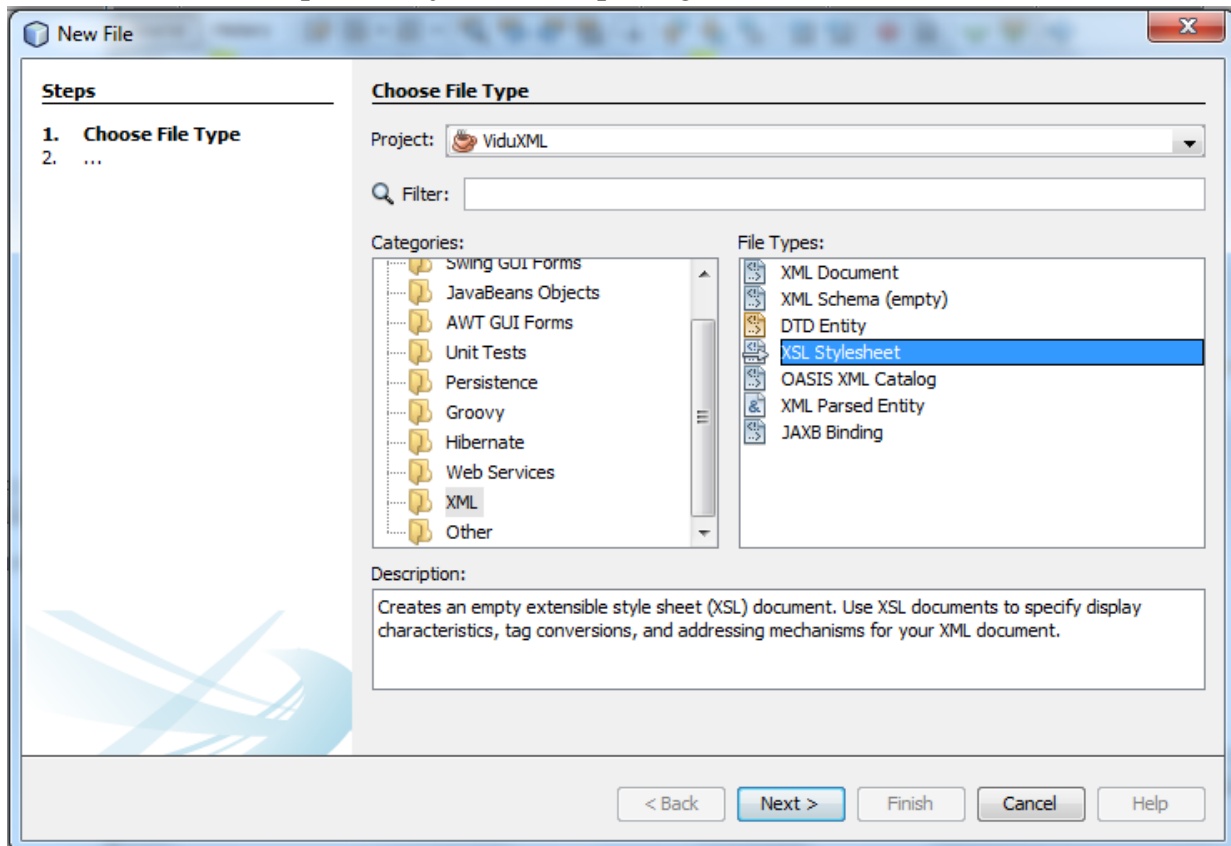
```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type="text/xsl" href="bongda.xslt"?>
<bongda>
  <cauthu>
    <hoten>David Beckham</hoten>
    <ghiban>250</ghiban>
    <sosut>300</sosut>
    <cohoi>70</cohoi>
  </cauthu>
  <cauthu>
    <hoten>Jose Mourinho</hoten>
    <ghiban>0</ghiban>
    <sosut>250</sosut>
    <cohoi>490</cohoi>
  </cauthu>
  <cauthu>
    <hoten>Donal Trump</hoten>
    <ghiban>200</ghiban>
    <sosut>250</sosut>
    <cohoi>49</cohoi>
  </cauthu>
  <!--vân vân-->
</bongda>
```

**XSL Transformation Result Table:**

| Cầu thủ       | Số bàn thắng | Số lần sút | Số lần được tạo cơ hội |
|---------------|--------------|------------|------------------------|
| David Beckham | 250          | 300        | 70                     |
| Jose Mourinho | 0            | 250        | 490                    |
| Donal Trump   | 200          | 250        | 49                     |

## Trên Java

- click chuột phải trên java source package (viduXML) – New – others





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

```
<!--
  Document    : trangmoi.xml
  Created on  : March 12, 2017, 10:41 AM
  Author      : Asus_Pro
  Description:
      Purpose of transformation follows.
-->

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html"/>

  <!-- TODO customize transformation rules
       syntax recommendation http://www.w3.org/TR/xslt
  -->
  <xsl:template match="/">
    <html>
      <head>
        <title>trangmoi.xml</title>
      </head>
      <body>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

#### 4 - Các Template

- Các Template là những lệnh trong 1 bảng định kiểu XSL kiểm soát việc 1 phần tử và nội dung của nó được chuyển đổi như thế nào
- Cú pháp

```
<xsl:template match="myElement">
```

```
<!--Mô tả style ở đây-->
```

```
</xsl:template>
```

- Mỗi phần tử con của phần tử “myElement” sẽ áp dụng kiểu được mô tả trong phần tử đó
- Một template được áp dụng bởi phần tử <xsl:apply-templates/> nó là đệ quy của <xml:template>
- Toán tử “/” tham chiếu tới node gốc của tài liệu XML

#### 5 - Cách trình bày các biểu thức

- Các hàm boolean  
=, < (hay &lt;=), <= (hay &lt;=), >, >=
- Các hàm số học



+, -, \*, div (chia), mod (phần dư), ceiling() – Lấy giá trị trần, floor() – lấy giá trị sàn, round() – hàm làm tròn

- Hàm làm việc với chuỗi:
  - Concat(s1,s2) – nối chuỗi
  - Contains(xâu nguồn,xâu so sánh) – kiểm tra chuỗi so sánh có trong chuỗi nguồn không
  - Substring(s,x,n) – lấy chuỗi con từ chuỗi s, bắt đầu từ vị trí x, độ dài cần lấy n

```
<xsl:template match="//cauthu">
```

Tên cầu thủ

```
<xsl:value-of select="substring(cauthu,0,10)">
```

```
</xsl:template>
```

VD -

```
<td>
  <xsl:value-of select="substring(hoten,0,5)"/>
</td>
```

VD – Thêm dòng thông kê vào ví dụ ở trên

```
<h2>Tổng số cú sút</h2>
<xsl:value-of select="sum(bongda/cauthu/sosut)"/>
<h2>Số lượng cầu thủ</h2>
<xsl:value-of select="count(bongda/cauthu)"/>
</body>
</html>
```

- Ta được kết quả





```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type="text/xsl" href="bongda.xslt"?>
<bongda>
  <cauthu>
    <hoten>David Beckham</hoten>
    <ghiban>250</ghiban>
    <sosut>300</sosut>
    <cohoi>70</cohoi>
  </cauthu>
  <cauthu>
    <hoten>Jose Mourinho</hoten>
    <ghiban>0</ghiban>
    <sosut>250</sosut>
    <cohoi>490</cohoi>
  </cauthu>
  <cauthu>
    <hoten>Donal Trump</hoten>
    <ghiban>200</ghiban>
    <sosut>250</sosut>
    <cohoi>49</cohoi>
  </cauthu>
  <!--vân vân-->
</bongda>
```

## Các nhân vật tiêu biểu bóng đá

| Cầu thủ | Số bàn thắng | Số lần sút | Số lần được tạo cơ hội |
|---------|--------------|------------|------------------------|
| Davi    | 250          | 300        | 70                     |
| Jose    | 0            | 250        | 490                    |
| Dona    | 200          | 250        | 49                     |

## Tổng số cú sút

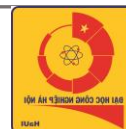
800

## Số lượng cầu thủ

3

## 6 - Một số phần tử thường dùng của XSL

- Phần tử value-of: lựa chọn 1 phần tử nào đó để trích xuất  
VD <xsl:value-of select="DS/HoaDon/MaHD"/>
- Phần tử attribute: Lấy giá trị thuộc tính trong thẻ XML  
VD <xsl:value-of select="DS/HoaDon/@tenthuoctinhcuathe"/>
- VD Tổng hợp



Cho file **HoaDonHang.xml** lưu trữ thông tin về các mặt hàng của từng hóa đơn.

<DS>

<HoaDon>

<MaHD> HD1 </MaHD>

<Ngày Bán> 2015-11-09 </Ngày Bán>

<LoạiHàng MaLoai="A" TenLoai="Điện tử">

<Hàng MaHàng="H1">

<TenHàng> Ti Vi </TenHàng>

<SoLuong> 60 </SoLuong>

<ĐơnViTính> Chiếc </ĐơnViTính>

<ĐơnGiá> 6000000 </ĐơnGiá>

</Hàng>

<Hàng MaHàng="H2">

<TenHàng> Tủ Lạnh </TenHàng>

<SoLuong> 80 </SoLuong>

<ĐơnViTính> Chiếc </ĐơnViTính>

<ĐơnGiá> 4200000 </ĐơnGiá>

</Hàng>

...

</LoạiHàng>

...

</HoaDon>

...

</DS>

Sử dụng XSLT để hiển thị thông tin về các hóa đơn theo mẫu sau:

\*Trong đó: Thành tiền = Số lượng \* đơn giá

### PHIẾU MUA HÀNG



Hóa đơn: ? (hiển thị mã hóa đơn) Ngày bán: ? (hiển thị ngày bán)

Loại hàng: ? (hiển thị tên loại hàng)

| Mã hàng | Tên hàng | Số lượng | Đơn giá | Thành tiền |
|---------|----------|----------|---------|------------|
|         |          |          |         |            |
|         |          |          |         |            |

Người giao



```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl"
>
  <xsl:output method="html" indent="yes"/>

  <xsl:template match="/">
    <html>
      <head>
        <title>Danh mục hóa đơn</title>
      </head>
      <body>
        <h2>PHIẾU MUA HÀNG </h2>
        <table border="0" >
          <tr>
            <td>
              Hóa đơn:<xsl:value-of select="DS/HoaDon/MaHD"/>
            </td>
            <td>
              Ngày bán:<xsl:value-of select="DS/HoaDon/NgayBan"/>
            </td>
          </tr>
          <tr>
            <td>
              Loại Hàng:<xsl:value-of select="DS/HoaDon/LoaiHang/@TenLoai"/>
            </td>
            <td>
            </td>
          </tr>
        </table>
```



```
<table border="2" cellpadding="3">
  <tr>
    <td>Mã Hàng</td><td>Tên hàng</td><td>Số lượng</td>
    <td>Đơn giá</td><td>Thành Tiền</td>
  </tr>
  <xsl:for-each select="DS/HoaDon/LoaiHang/Hang">
    <tr>
      <td>
        <xsl:value-of select="@MaLoai"/>
      </td>
      <td>
        <xsl:value-of select="TenHang"/>
      </td>
      <td>
        <xsl:value-of select="SoLuong"/>
      </td>
      <td>
        <xsl:value-of select="DonGia"/>
      </td>
      <td>
        <xsl:value-of select="DonGia*SoLuong"/>
      </td>
    </tr>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
/xsl:stylesheet>
```

- Kết quả file

| Danh mục hóa đơn <span>×</span>      |          |          |         |            |
|--------------------------------------|----------|----------|---------|------------|
| DANH MỤC HÓA ĐƠN BÁN HÀNG            |          |          |         |            |
| Hóa đơn: HD1    Ngày bán: 2015-11-09 |          |          |         |            |
| Loại Hàng: Điện tử                   |          |          |         |            |
| Mã Hàng                              | Tên hàng | Số lượng | Đơn giá | Thành Tiền |
| H1                                   | Ti Vi    | 60       | 6000000 | 360000000  |
| H2                                   | Tủ lạnh  | 80       | 4200000 | 336000000  |



- Nếu đưa ra kết quả như hình dưới đây thực hiện:

## DANH MỤC HÓA ĐƠN BÁN HÀNG

Hóa đơn: HD1 Ngày bán: 2015-11-09

Loại Hàng: Điện tử

| Mã Hàng | Tên hàng | Số lượng | Đơn giá | Thành Tiền |
|---------|----------|----------|---------|------------|
| H1      | Ti Vi    | 60       | 6000000 | 360000000  |
| H2      | Tủ lạnh  | 80       | 4200000 | 336000000  |

## DANH MỤC HÓA ĐƠN BÁN HÀNG

Hóa đơn: HD2 Ngày bán: 2015-1-03

Loại Hàng: Điện tử

| Mã Hàng | Tên hàng  | Số lượng | Đơn giá | Thành Tiền |
|---------|-----------|----------|---------|------------|
| H1      | Máy Giặt  | 60       | 6000000 | 360000000  |
| H2      | Đài Radio | 80       | 4200000 | 336000000  |

## DANH MỤC HÓA ĐƠN BÁN HÀNG

Hóa đơn: HD3 Ngày bán: 2017-06-09

Loại Hàng: Điện tử

| Mã Hàng | Tên hàng   | Số lượng | Đơn giá | Thành Tiền |
|---------|------------|----------|---------|------------|
| H1      | Lò vi sóng | 60       | 6000000 | 360000000  |
| H2      | Quạt cây   | 80       | 4200000 | 336000000  |

- Tạo vòng lặp ngoài

```
<head>
  <title>Danh mục hóa đơn</title>
</head>
<xsl:for-each select="DS/HoaDon">
  <body>

    <h2>DANH MỤC HÓA ĐƠN BÁN HÀNG</h2>
```

- Các câu lệnh con bỏ đi  
“DS/HoaDon/”
- Kết thúc lặp



```

        </body>
    </xsl:for-each>
</html>
</xsl:template>
</xsl:stylesheet>

```

## 7 - Phần tử variable – khai báo biến

- Dùng để khai báo 1 biến lưu trữ
- Cú pháp:

```
<xsl:variable name="tên biến" select="gán giá trị cho biến">
```

```
</xsl:variable>
```

- Hoặc

```
<xsl:variable name="tên biến" > Gán giá trị cho biến
```

```
</xsl:variable>
```

- Sử dụng biến: \$tên biến
- VD

```
<xsl:variable name="i" select="5">
```

```
</xsl:variable>
```

## 8 - Phần tử if

- Phần tử if là phần tử dùng để kiểm tra điều kiện của 1 biểu thức logic
- Cú pháp:

```
<xsl:if test="Biểu thức">...</xsl:if>
```

- VD: Đưa ra danh sách các mặt hàng có số lượng  $\geq 50$



```
<xsl:for-each select="LoaiHang/Hang">
  <xsl:if test="SoLuong>50">
    <tr>
      <td>
        <xsl:value-of select="@MaLoai"/>
      </td>
      <td>
        <xsl:value-of select="TenHang"/>
      </td>
      <td>
        <xsl:value-of select="SoLuong"/>
      </td>
      <td>
        <xsl:value-of select="DonGia"/>
      </td>
      <td>
        <xsl:value-of select="DonGia*SoLuong"/>
      </td>
    </tr>
  </xsl:if>
</xsl:for-each>
```

- **Kết quả**

Danh mục hóa đơn ✕

## DANH MỤC HÓA ĐƠN BÁN HÀNG

Hóa đơn: HD1 Ngày bán: 2015-11-09

Loại Hàng: Điện tử

| Mã Hàng | Tên hàng | Số lượng | Đơn giá | Thành Tiền |
|---------|----------|----------|---------|------------|
| H1      | Ti Vi    | 60       | 6000000 | 360000000  |
| H2      | Tủ lạnh  | 80       | 4200000 | 336000000  |

## DANH MỤC HÓA ĐƠN BÁN HÀNG

Hóa đơn: HD2 Ngày bán: 2015-1-03

Loại Hàng: Điện tử

| Mã Hàng | Tên hàng  | Số lượng | Đơn giá | Thành Tiền |
|---------|-----------|----------|---------|------------|
| H2      | Đài Radio | 70       | 4200000 | 294000000  |

## DANH MỤC HÓA ĐƠN BÁN HÀNG

Hóa đơn: HD3 Ngày bán: 2017-06-09

Loại Hàng: Điện tử

| Mã Hàng | Tên hàng   | Số lượng | Đơn giá | Thành Tiền |
|---------|------------|----------|---------|------------|
| H1      | Lò vi sóng | 90       | 6000000 | 540000000  |



## 9 - Phần tử điều khiển choose

- Tương tự switch trong C
- Cú pháp

<xsl:choose>

<xsl:when test="Biểu thức 1">...</xsl:when>

<xsl:when test="Biểu thức 2">...</xsl:when>

...

<xsl:when test="Biểu thức n">...</xsl:when>

<xsl:otherwise>...</xsl:when>

</xsl:choose>

- VD: Tạo file XML

```
<?xml version="1.0" encoding="utf-8"?>
<!--Danh sách sinh viên và điểm-->
<DSSV>
  <sv masv="1">
    <hoten>Nguyễn Văn A</hoten>
    <diem>7</diem>
  </sv>
  <sv masv="2">
    <hoten>Lê Thị B</hoten>
    <diem>4</diem>
  </sv>
  <sv masv="3">
    <hoten>Đào Văn C</hoten>
    <diem>9</diem>
  </sv>
</DSSV>
```

- Tạo file XSLT





```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude
  >

  <xsl:output method="html" indent="yes"/>

  <xsl:template match="/">
    <html>
      <head>
        <title>ví dụ về choose</title>
      </head>
      <body>
        <h2>
          <b>DANH SÁCH SINH VIÊN</b>
        </h2>
        <xsl:for-each select="DSSV/sv">
          <!--Gán biến điểm để sử dụng cho choose-->
          <xsl:variable name="biendiem" select="diem">
          </xsl:variable>
          <B>Mã SV:</B>
          <xsl:value-of select="@masv"/>

          <br></br>
          <B>Họ tên:</B>
          <xsl:value-of select="hoten"/>
          <br></br>
          <B>Điểm:</B>
          <xsl:value-of select="diem"/>
          <br></br>
          <B>Điểm bằng chữ:</B>
          <xsl:choose>
            <xsl:when test="$biendiem<5">Kém</xsl:when>
            <xsl:when test="$biendiem>8">Giỏi</xsl:when>
            <xsl:otherwise>Trung bình</xsl:otherwise>
          </xsl:choose>
          <br></br>
          <hr></hr>
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

## 10 - Thành phần <xsl:sort> Sắp xếp theo phần tử được chỉ định

### Cú pháp:

```
<xsl:sort select="Tên phần tử" order="ascending | descending"/>
```

### Ví dụ:

```
<xsl:for-each select="catalog/cd">
```




```
<xsl:sort select="title" order="descending"/>
<xsl:value-of select="title"/>
</xsl:for-each>
```

## 11- In vị trí xuất hiện của một phần tử trong tài liệu.

Ví dụ:

```
<xsl:value-of select="position()"/>
```



| STT | ID | Title            | Artist       | Country | Price | Year |
|-----|----|------------------|--------------|---------|-------|------|
| 1   | 01 | Empire Burlesque | Bob Dylan    | USA     | 10.90 | 1986 |
| 2   | 02 | Hide your heart  | Bonnie Tyler | UK      | 9.90  | 1985 |

## 12 – Sử dụng cả xml-schema và xslt trong tạo ràng buộc và hiển thị file xml

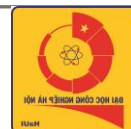
- Trước tiên tạo file xml-schema tạo các ràng buộc trên file XML
- Tạo file XML sử dụng ràng buộc xsd vừa được tạo (trở tới file xsd)
- Tạo file xslt với 1 namespace trở tới file xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl"
  xmlns:a="http://tempuri.org/ktra.xsd"
>
```

- Tất cả các element trong file xslt đều được viết dưới tham chiếu của namespace này
- Các attribute thì giữ nguyên không cần tham chiếu namespace

```
<xsl:for-each select="a:ThuVien/a:LoaiSach">
  <table cellpadding="20">
    <tr>
      <td align="left">
        Ma Loai: <xsl:value-of select="@MaLoai"/>
      </td>
      <td align="right">
        Loai Sach: <xsl:value-of select="a:TenLoai"/>
      </td>
    </tr>
  </table>
```

- File XML trở tham chiếu tới file xslt và chạy file xml



**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.