1. **Xslt trong XML**
2. Định nghĩa

XSL(eXtensible Stylesheet Language) là một dạng ngôn ngữ dựa trên nền tảng XML để tạo ra stylesheet nhằm mục đích định dạng nội dung XML cho nhiều mục đích hiển thị khác nhau, đồng thời XSL còn có khả năng chuyển đổi XML thành dạng định dạng khác. XSL bao gồm 3 phần:

* XSL Transformation (XSLT): là một ngôn ngữ dạng XML bao gồm các thẻ xử lý cho phép chuyển đổi XML thành định dạng khác như HTML, XML khác
* XML Path Language (X-Path): là một ngôn ngữ dùng để truy xuất các thành phần trong **XML**
* XSL Formatting Object (XSL-FO): là ngôn ngữ dùng để định dạng XML
* XML Path Language (XPath) là một ngôn ngữ thiết kế ra với mục đích giúp cho ứng dụng có thể di chuyển bên trong XML document và truy xuất các giá trị cũng như thuộc tính của các elements.

1. Chuyển đổi XML bằng XSLT

Tài liệu XML 🡪 Chương trình XSLT 🡪 Tài liệu văn bản

Như:

* XML 🡪 HTML

VD:

|  |  |  |
| --- | --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <tong>  <songuyen1>10</songuyen1>  <songuyen2>20</songuyen2>  </tong> | 🡪 | Phép cộng 2 số: 10 + 20 |

* XML 🡪 XML

|  |  |  |
| --- | --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <tong>  <songuyen1>10</songuyen1>  <songuyen2>20</songuyen2>  </tong> | 🡪 | <?xml version="1.0" encoding="UTF-8"?>  <tong songuyen1="10" songuyen2="20" /> |

1. Cú pháp cơ bản của XSLT

* Thành phần <xsl:template>
* Dùng để so khớp với các phần tử XML
* Tìm và so khớp với phần tử gốc (root)

**<xsl:template** match=**"**/ **">**

* Tìm và so khớp với các phần tử con

**<xsl:template match="Tên phần tử">**

* Thành phần <xsl:value-of>
* Dùng để trích xuất thông tin của một phần tử XML

|  |
| --- |
| **<xsl:value-of select="Tên phần tử"/>** |

VD:

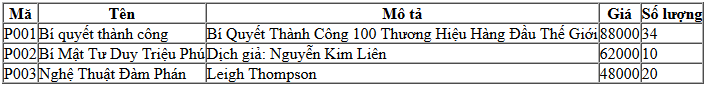
|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="newss.xsl"?>  <tong>  **<songuyen1>10</songuyen1>**  <songuyen2>20</songuyen2>  </tong> | <?xml version="1.0" encoding="UTF-8"?>  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">  <xsl:output method="html"/>  <xsl:template match="/">  <html>  <head>  <title>newss.xsl</title>  </head>  <body>  **<xsl:value-of select="/tong/songuyen1"/>**  <xsl:text>+</xsl:text>  <xsl:value-of select="/tong/songuyen2"/>  </body>  </html>  </xsl:template>  </xsl:stylesheet> |

* 10+20
* Thành phần <xsl:for-each>
* Vòng lặp

|  |
| --- |
| **<xsl:for-each select="Tên phần tử">** |

VD:

|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="new.xsl"?>  <PRODUCTDATA>  <Product ProdID="P001">  <ProductName>Bí quyết thành công</ProductName>  <ProductDesc>Bí Quyết Thành Công 100 Thương Hiệu Hàng Đầu Thế Giới</ProductDesc>  <Price>88000</Price>  <Quantity>34</Quantity>  </Product>  <Product ProdID="P002">  <ProductName>Bí Mật Tư Duy Triệu Phú </ProductName>  <ProductDesc>Dịch giả: Nguyễn Kim Liên</ProductDesc>  <Price>62000</Price>  <Quantity>10</Quantity>  </Product>  <Product ProdID="P003">  <ProductName>Nghệ Thuật Đàm Phán</ProductName>  <ProductDesc>Leigh Thompson</ProductDesc>  <Price>48000</Price>  <Quantity>20</Quantity>  </Product>  </PRODUCTDATA> | <xsl:for-each select="PRODUCTDATA/Product">  <tr>  <td>  <xsl:value-of select="@ProdID"/>  </td>  <td>  <xsl:value-of select="ProductName"/>  </td>  <td>  <xsl:value-of select="ProductDesc"/>  </td>  <td>  <xsl:value-of select="Price"/>  </td>  <td>  <xsl:value-of select="Quantity"/>  </td>  </tr>  </xsl:for-each> |



* Thành phần <xsl:sort>
* Sắp xếp theo phần tử được chỉ định

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

* Thành phần <xsl:if>
* Kiểm tra dữ liệu thỏa mãn điều kiện nào đó.

|  |
| --- |
| <xsl:if test=**"**expression**"**>…</xsl:if> |

Các toán tử so sánh:

|  |
| --- |
| = (equal)  != (not equal)  &lt; less than (>)  &gt; greater than (<) |

* Thành phần <xsl:choose>
* Lọc dữ liệu với nhiều trường hợp

|  |
| --- |
| **<xsl:choose>**  **<xsl:when test="expression">**  **... some output ...**  **</xsl:when>**  **<xsl:otherwise>**  **... some output ....**  **</xsl:otherwise>**  **</xsl:choose>** |

* Thuộc tính Select
* In vị trí xuất hiện của một phần tử trong tài liệu.

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

1. Khai báo trong file XML

Khai báo thêm trong file XML:

|  |
| --- |
| <?xml-stylesheet type="text/xsl" href="file\_xsl"?> |

VD: <?xml-stylesheet type="text/xsl" href="sinhvien.xsl"?>

1. Bài tập mẫu
2. Đề bài:

Tạo tập tin SinhVien.xml có định nghĩa kiểu dữ liệu sử dụng XML Schema.

* Giới tính chỉ nhập Nam hoặc Nữ
* Mã lớp nhập theo qui ước: 4 ký tự đầu là chữ cái, 2 ký tự kế tiếp là số

Truy vấn và hiển thị thông tin sinh viên tăng dần theo tên

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **MaSv** | **TenSv** | **GioiTinh** | **NgaySinh** | **MaLop** |
| 1 | sv01 | Phan Tuấn Cường | Nam | 03/03/1993 | CNTT14 |
| 2 | sv02 | Huỳnh Chí Tâm | Nam | 08/03/1993 | CNTT14 |
| 3 | sv03 | Vương Chí Dũng | Nam | 14/04/1991 | CNTT14 |

1. Làm bài:

* B1: Tạo file schema sinhvienchema.xsd

|  |
| --- |
| <?xml version="1.0"?>  <!--  To change this license header, choose License Headers in Project Properties.  To change this template file, choose Tools | Templates  and open the template in the editor.  -->  <xs:schema version="1.0"  xmlns:xs="http://www.w3.org/2001/XMLSchema"  elementFormDefault="qualified">  <xs:element name="listSV">  <xs:complexType>  <xs:sequence>  <xs:element name="SinhVien" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="MaSv" type="xs:string">  </xs:element>  <xs:element name="TenSv" type="xs:string">  </xs:element>  <xs:element name="GioiTinh">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="Nam|Nữ"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="NgaySinh" type="xs:string">  </xs:element>  <xs:element name="MaLop">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="[a-zA-Z]{4}[0-9]{2}"></xs:pattern>  </xs:restriction>  </xs:simpleType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

* B2: Tạo file XML sinhvien.xml:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="sinhvien.xsl"?>  <listSV  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'  xsi:noNamespaceSchemaLocation='sinhvienchema.xsd'>  <SinhVien>  <MaSv>sv01</MaSv>  <TenSv>Phan Tuấn Cường</TenSv>  <GioiTinh>Nam</GioiTinh>  <NgaySinh>03/03/1993</NgaySinh>  <MaLop>CNTT14</MaLop>  </SinhVien>  <SinhVien>  <MaSv>sv02</MaSv>  <TenSv>Huỳnh Chí Tâm</TenSv>  <GioiTinh>Nam</GioiTinh>  <NgaySinh>08/03/1993</NgaySinh>  <MaLop>CNTT14</MaLop>  </SinhVien>  <SinhVien>  <MaSv>sv03</MaSv>  <TenSv>Vương Chí Dũng</TenSv>  <GioiTinh>Nam</GioiTinh>  <NgaySinh>14/04/1991</NgaySinh>  <MaLop>CNTT14</MaLop>  </SinhVien>  </listSV> |

* B3: Tạo file sinhvien.xsl

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">  <xsl:output method="html"/>  <xsl:template match="/">  <html>  <head>  <title>sinhvien.xsl</title>  </head>  <body>  <h1>Danh sách sinh viên</h1>  <table border="1" cellpadding="0" cellspacing="0">  <tr>  <th>STT</th>  <th>MaSv</th>  <th>TenSv</th>  <th>GioiTinh</th>  <th>NgaySinh</th>  <th>MaLop</th>  </tr>  <xsl:for-each select="listSV/SinhVien">  <xsl:sort select="MaSv" order="descending"/>  <tr>  <td>  <xsl:value-of select="position()"/>  </td>  <td>  <xsl:value-of select="MaSv"/>  </td>  <td>  <xsl:value-of select="TenSv"/>  </td>  <td>  <xsl:value-of select="GioiTinh"/>  </td>  <td>  <xsl:value-of select="NgaySinh"/>  </td>  <td>  <xsl:value-of select="MaLop"/>  </td>  </tr>  </xsl:for-each>  </table>  </body>  </html>  </xsl:template>  </xsl:stylesheet> |