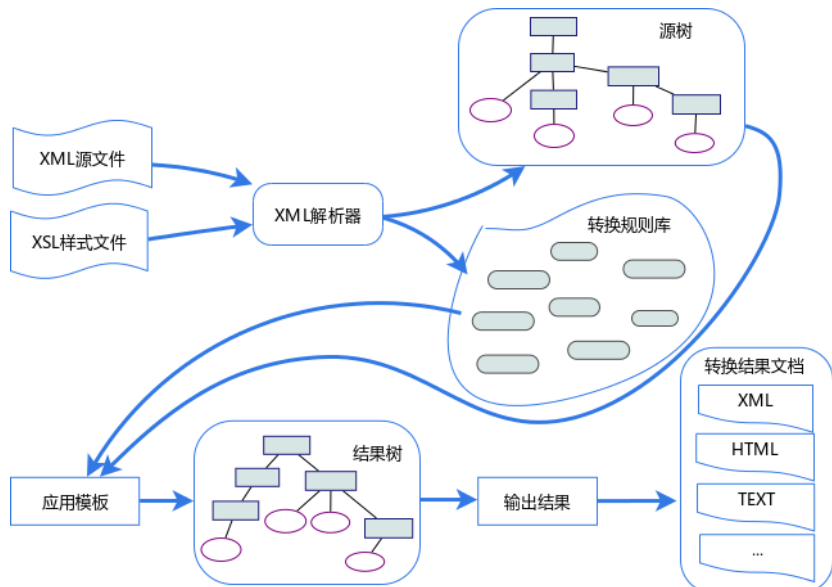


# XML 原理与应用

夏天

中国人民大学



# 本章学习目标

- 了解 XSLT 的发展历史、XSLT 与 XSL 的关系
- 掌握 XSLT 的转换处理过程
- 掌握 XSLT 的基本语法
- 能够使用 SAXON 进行 XSLT 测试

# 目录

- XSLT 概述
- 如何测试 XSLT
- XSLT 快速入门
- XSLT 的输出格式控制
- XSLT 的逻辑处理元素
- XSLT 的模式
- XSLT 的命名模板
- XSLT 的函数
- XSLT 2.0

## 7.1 概述

- XSLT 与 XSL
- XSLT 的作用
- XSLT 的工作流程
- XSLT 的应用模式
- XSLT 与 CSS 的区别

## 7.1.1 XSLT 与 XSL I

- XSLT

- 可扩展样式语言转换 ( Extensible Stylesheet Language Transformations )
- 可扩展样式语言 XSL ( Extensible Stylesheet Language ) 的一个组成部分

- XSL

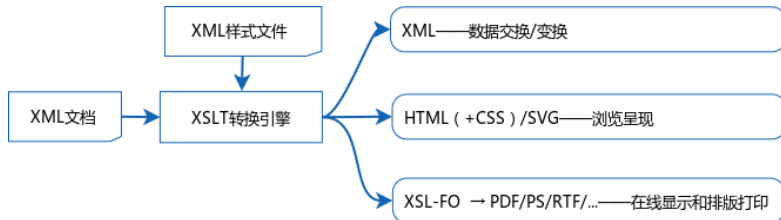
- 定义 XML 文档的格式化和呈现方式，以方便在屏幕、纸张等媒介上显示或通过语音输出
- 包含两个相对独立的阶段：
  - 结构转换阶段：实现 XML 文档元素的选择、分组、重新排列
  - 格式化处理阶段，实现具体的呈现功能
- XSL 又被拆分为两部分：

## 7.1.1 XSLT 与 XSL II

- XSLT 用于定义转换
  - 剩余的部分——仍称为 XSL，也有人称之为 XSL-FO ( XSL Formatting )
- XSL 目前包含三个组成部分
- XSL 转换 XSLT：描述如何把一个 XML 文档转换为另外一个文本格式的文档，如 XML 文档或 HTML 文档
  - XML 路径语言 XPath：在 XSLT 之中实现存取或引用 XML 文档特定部分的表达式语言
  - XSL 格式化对象 ( XSL Formatting Objects, XSL-FO )：定义 XML 文档的格式化显示方式

## 7.1.2 XSLT 的作用

- 把一个 XML 文档转换为另一个类型的文档，可以在输出文档里增删元素和属性，可以对源文档的元素进行重新排列，并有选择的组织显示
- 常见的三种转换





# 把一个版本的 XML 转换为另一个版本

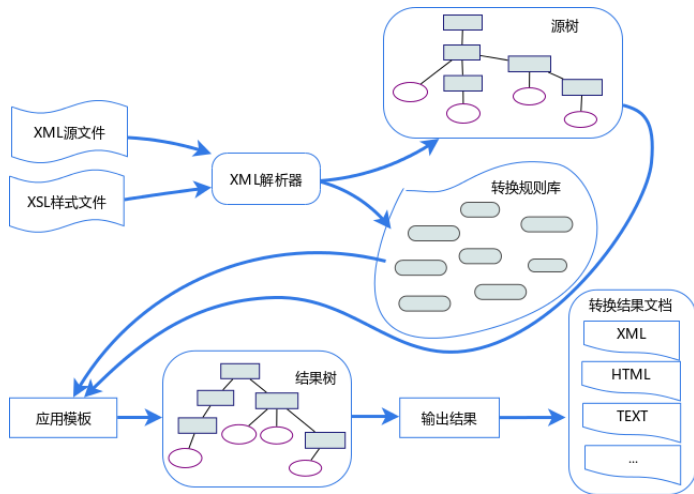
```
1 <v1:order xmlns:v1="http://www.test.com/v1">  
2   <product name="打印纸" price="30" quantity="5"/>  
3 </v1:order>
```

```
1 <v2:order xmlns:v2="http://www.test.com/v2">  
2   <product quantity="5">  
3     <name>打印纸</name>  
4     <price>30</price>  
5   </product>  
6 </v2:order>
```

# 把 XML 转换为 HTML

```
1 <html>
2   <head><title>Order Details</title></head>
3   <body>
4     <p>名称：打印纸</p>
5     <p>单价：30元；数量：5 </p>
6   </body>
7 </html>
```

## 7.1.3 XSLT 的工作流程



## 7.1.4 XSLT 的应用模式

- 服务器端应用模式
  - 转换是在服务器端完成
- 客户端应用模式
  - 转换由客户端的浏览器完成
- 独立模式
  - 转换由独立程序完成，如 Saxon

## 7.1.5 XSLT 与 CSS 的区别

- 转换动作不同
  - XSLT 采用转换方式，把 XML 文档由一种格式转换为另一种格式；CSS 无转换动作，只是针对 XML 元素设定显示样式。
- 互动性不同
  - CSS 可以实现闪烁等动态效果，XSLT 转换本身则没有这方面的功能。
- 语法不同
  - XSL 样式文件为标准的 XML 文件，而 CSS 的语法自成一格

## 7.2 如何测试 XSLT

- 浏览器
- XML 工具
- 独立的 XSLT 处理器

## 7.2.1 通过浏览器测试 XSLT

- 在 XML 文档中通过 `<?xml-stylesheet?>` 处理指令进行样式的关联
- 通过浏览器打开 XML 文档查看转换效果
- 看不到转换结果的源代码

## 7.2.2 通过 XML 专业工具测试





## 7.2.2 通过 XSLT 处理器测试

- Saxon

- 安装 Java
- 下载并配置 Saxon
- 指定参数实现样式转换
  - 语法：
    - `java -jar saxon9he.jar -s:source -xsl:stylesheet -o:output`
  - 例子：
    - `java -jar saxon9he.jar -s:7-1.1.xml -xsl:7-1.2.xsl -o:7-1.out.xml`

## 7.3 XSLT 快速入门 I

Code 1: "7-2.xml"

```
1 <?xml version="1.0"?>
2 <books>
3   <provider>Summer</provider>
4   <book isbn="978-7-115-21703-5">
5     <title>Python高级编程</title>
6     <price>45.00</price>
7     <authors>
8       <author>Tarek Ziade</author>
9       <author type="translator">姚军</author>
10      <author type="translator">夏海伦</author>
11      <author type="translator">王秀丽</author>
12    </authors>
13    <press>人民邮电出版社</press>
14    <pages>306</pages>
```

## 7.3 XSLT 快速入门 II

```
15      <description>介绍了Python语言的最佳实践和敏捷开发方法...</description>
16      <cover>book-python.jpg</cover>
17  </book>
18  <book isbn="978-7-115-28282-8">
19      <title>数学之美</title>
20      <price>45.00</price>
21      <authors>
22          <author>吴军</author>
23      </authors>
24      <press>人民邮电出版社</press>
25      <pages>304</pages>
26
27      <description>读了“数学之美”，才发现大学时学的数学知识...</description>
28      <cover>book-math.jpg</cover>
29  </book>
```

## 7.3 XSLT 快速入门 III

```
29 <book isbn="978-7-5641-1139-7">
30   <title>集体智慧编程(影印版)</title>
31   <price>58.00</price>
32   <authors>
33     <author>Toby Segaran</author>
34   </authors>
35   <press>东南大学出版社</press>
36   <pages>334</pages>
37
38   <description>Toby的书非常成功地将复杂的机器学习算法问题...</description>
39   <cover>book-collective.jpg</cover>
40 </book>
</books>
```

## 7.3 XSLT 快速入门 IV

Code 2: "7-2.xsl"

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   version="2.0">
3   <xsl:template match="/">
4     <html>
5       <head><title>图书列表</title></head>
6       <body>
7         <xsl:apply-templates/>
8       </body>
9     </html>
10  </xsl:template>
11  <xsl:template match="books">
12    <table>
13      <xsl:apply-templates select="book"/>
```

## 7.3 XSLT 快速入门 V

```
14     </table>
15 </xsl:template>
16 <xsl:template match="book">
17     <tr>
18         <td><xsl:value-of select="title"/></td>
19     </tr>
20 </xsl:template>
21 </xsl:stylesheet>
```

```
java -jar saxon9he.jar -s:7-2.xml -xsl:7-2.xsl -o:7-2.out.html
```

## 7.3 XSLT 快速入门 VI

Code 3: "7-2.out.html"

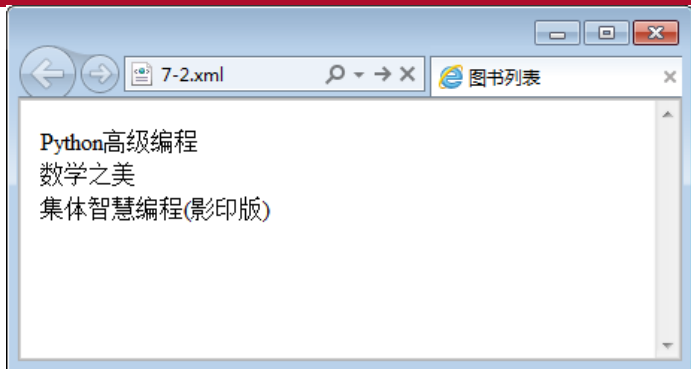
```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html;
      charset=UTF-8">
4     <title>图书列表</title>
5   </head>
6   <body>
7     <table>
8       <tr><td>Python高级编程</td></tr>
9       <tr><td>数学之美</td></tr>
10      <tr><td>集体智慧编程(影印版)</td></tr>
11    </table>
12  </body>
13 </html>
```

## 7.3 XSLT 快速入门 VII

利用 `<?xml-stylesheet?>` 处理指令在“7-2.xml”中建立与“7-2.xsl”的关联，通过浏览器查看结果



## 7.3 XSLT 快速入门 VIII



## 7.3.1 stylesheet 元素

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="2.0">
```

## 7.3.2 template 元素

```
<xsl:template match="要匹配的模式串"  
    name="名称"  
    priority="代表优先级的数字">  
    <!--模板的具体内容 -->  
</xsl:template>
```

## 7.3.3 apply-templates 元素

```
<xsl:template match="/books">  
  <xsl:apply-templates select="book"/>  
</xsl:template>
```

```
<xsl:template match="/books">  
  <xsl:apply-templates/>  
</xsl:template>
```

## 7.3.4 value-of 元素

- 根据 select 属性值从源树中读取信息
- xsl:value-of 在 XSLT 的 1.0 和 2.0 版本中转换行为不一致

```
<xsl:value-of select="author"/>
```

- 在 1.0 中，仅选取第一个结点的值，结果为：Tarek Ziade
- 在 2.0 中，顺序输出所有的结点，默认通过空格分隔：Tarek Ziade 姚军夏海伦王秀丽
- 可更换分隔符，如下：

```
<xsl:value-of select="author" separator=","/>
```

- 此时结果为：Tarek Ziade, 姚军, 夏海伦, 王秀丽

## 7.3.5 attribute 元素

- 生成属性信息，例如：

```
<img>  
  <xsl:attribute name="src"><xsl:value-of  
    select="cover"/></xsl:attribute>  
  <xsl:attribute name="width">80</xsl:attribute>  
  <xsl:attribute name="height">100</xsl:attribute>  
</img>
```

- 结果：

```

```

## 7.4 XSLT 的输出格式控制

- 用于指定转换生成的文档格式：
  - XML、HTML、XHTML 或纯文本格式
- 例如：声明转换结果的文本为纯文本格式

```
<xsl:output method="text"/>
```

- method 的取值：
  - text、xml、xhtml 或 html
- XSLT 1.0 不支持该指令

## 7.5 XSLT 的逻辑处理元素

- 条件处理元素
  - if
  - choose
- 循环：for-each
- 排序：sort



Code 4: 样式文档

```
1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
   version="2.0">  
2   <xsl:template match="/">  
3     <html><body>  
4       <xsl:apply-templates select="books/book"/>  
5     </body></html>  
6   </xsl:template>  
7   <xsl:template match="book">  
8     <xsl:if test="price < 50">  
9       <p><xsl:value-of select="title"/></p>  
10    </xsl:if>  
11  </xsl:template>  
12 </xsl:stylesheet>
```

## Code 5: 转换结果

```
1 <html>
2   <body>
3     <p>Python高级编程</p>
4     <p>数学之美</p>
5   </body>
6 </html>
```

## Code 6: 样式文档

```
1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="2.0">  
2   <xsl:template match="/">  
3     <html><body>  
4       <xsl:apply-templates select="books/book"/>  
5     </body></html>  
6   </xsl:template>  
7   <xsl:template match="book">  
8     <xsl:choose>  
9       <xsl:when test="price < 50"><p><xsl:value-of  
    select="title"/></p></xsl:when>  
10      <xsl:when test="price < 100"><p>* <xsl:value-of  
    select="title"/></p></xsl:when>
```

# choose 元素 II

```
11      <xsl:otherwise><p># <xsl:value-of  
      select="title"/></p></xsl:otherwise>  
12    </xsl:choose>  
13  </xsl:template>  
14 </xsl:stylesheet>
```

## Code 7: 转换结果

```
1 <html>  
2   <body>  
3     <p>Python高级编程</p>  
4     <p>数学之美</p>  
5     <p>*集体智慧编程(影印版)</p>  
6   </body>  
7 </html>
```

## Code 8: 样式文档

```
1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="2.0">  
2   <xsl:template match="/">  
3     <html><body>  
4       <xsl:apply-templates select="books"/>  
5     </body></html>  
6   </xsl:template>  
7   <xsl:template match="books">  
8     <xsl:for-each select="book">  
9       <p><xsl:value-of select="title"/></p>  
10    </xsl:for-each>  
11  </xsl:template>  
12 </xsl:stylesheet>
```

## Code 9: 转换结果

```
1 <html>
2   <body>
3     <p>Python高级编程</p>
4     <p>数学之美</p>
5     <p>集体智慧编程(影印版)</p>
6   </body>
7 </html>
```

Code 10: 样式文档

```
1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
   version="2.0">  
2   <xsl:template match="/">  
3     <html><body>  
4       <xsl:apply-templates select="books"/>  
5     </body></html>  
6   </xsl:template>  
7   <xsl:template match="books">  
8     <xsl:for-each select="book">  
9       <xsl:sort select="price" order="descending"/>  
10      <p><xsl:value-of select="title"/></p>  
11    </xsl:for-each>  
12  </xsl:template>  
13 </xsl:stylesheet>
```

## Code 11: 转换结果

```
1 <html>
2   <body>
3     <p>集体智慧编程(影印版)</p>
4     <p>Python高级编程</p>
5     <p>数学之美</p>
6   </body>
7 </html>
```



## 7.6 XSLT 的模式 -mode I

Code 12: 样式文档

```
1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="2.0">  
2   <xsl:template match="/">  
3     <html><body>  
4       <h2>按照价格降序排列的图书名称</h2>  
5       <xsl:apply-templates select="books" mode="simple"/>  
6       <h2>图书详细列表</h2>  
7       <xsl:apply-templates select="books" mode="detail"/>  
8     </body></html>  
9   </xsl:template>  
10  <xsl:template match="books" mode="simple">  
11    <ul>  
12      <xsl:for-each select="book">  
13        <xsl:sort select="price" order="descending"/>
```

## 7.6 XSLT 的模式 -mode II

```
14      <li><xsl:value-of select="title"/></li>
15    </xsl:for-each>
16  </ul>
17 </xsl:template>
18
19 <xsl:template match="books" mode="detail">
20   <table>
21     <tr>
22
23       <td>ISBN</td><td>标题</td><td>作者</td><td>出版社</td><td>价格
24     </tr>
25     <xsl:for-each select="book">
26       <xsl:sort select="pages" order="ascending"/>
27       <tr>
28         <td><xsl:value-of select="@isbn"/></td>
29         <td><xsl:value-of select="title"/></td>
```

## 7.6 XSLT 的模式 -mode III

```
29      <td><xsl:value-of select="authors"/></td>
30      <td><xsl:value-of select="press"/></td>
31      <td><xsl:value-of select="price"/></td>
32      <td><img><xsl:attribute name="src"
select="cover"/></img></td>
33    </tr>
34  </xsl:for-each>
35 </table>
36 </xsl:template>
37 </xsl:stylesheet>
```

## 7.6 XSLT 的模式 -mode IV

Code 13: 转换结果

```
1 <html>
2   <body>
3     <h2>按照价格降序排列的图书名称</h2>
4     <ul>
5       <li>集体智慧编程(影印版)</li>
6       <li>Python高级编程</li>
7       <li>数学之美</li>
8     </ul>
9     <h2>图书详细列表</h2>
10    <table>
11      <tr>
12
13        <td>ISBN</td><td>标题</td><td>作者</td><td>出版社</td><td>价格
```

## 7.6 XSLT 的模式 -mode V

```
14    <tr>
15        <td>978-7-115-28282-8</td>
16        <td>数学之美</td>
17        <td>吴军</td>
18        <td>人民邮电出版社</td>
19        <td>45.00</td>
20        <td></td>
21    </tr>
22    <tr>
23        <td>978-7-115-21703-5</td>
24        <td>Python高级编程</td>
25        <td>Tarek Ziade 姚军 夏海伦 王秀丽</td>
26        <td>人民邮电出版社</td>
27        <td>45.00</td>
28        <td></td>
29    </tr>
```

## 7.6 XSLT 的模式 -mode VI

```
30     <tr>
31         <td>978-7-5641-1139-7</td>
32         <td>集体智慧编程(影印版)</td>
33         <td>Toby Segaran</td>
34         <td>东南大学出版社</td>
35         <td>58.00</td>
36         <td></td>
37     </tr>
38 </table>
39 </body>
40 </html>
```

## 7.7 XSLT 的命名模版 I

- 模版定义语法：

```
<xsl:template name="模板名称">  
  <!--模板的具体内容 -->  
</xsl:template>
```

- 定义好的模板通过 xsl:call-template 指令进行调用
- 示例：

## 7.7 XSLT 的命名模版 II

Code 14: 样式文档

```
1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="2.0">  
2   <xsl:template match="/">  
3     <html><body>  
4       <h2>图书提供者信息</h2>  
5       <xsl:call-template name="providerTemplate"/>  
6       <h2>按照价格降序排列的图书名称</h2>  
7       <xsl:apply-templates select="books" mode="simple"/>  
8       <h2>图书详细列表</h2>  
9       <xsl:apply-templates select="books" mode="detail"/>  
10    </body></html>  
11  </xsl:template>  
12  
13  <xsl:template name="providerTemplate">
```



## 7.7 XSLT 的命名模版 III

```
14    <p>本图书列表由 :  
15    <strong><xsl:value-of select="/books/provider"/></strong>提供.  
16    </p>  
17  </xsl:template>  
18  
19  <xsl:template match="books" mode="simple">  
20    <!-- 省略, 同7-7.xsl对应内容... -->  
21  </xsl:template>  
22  
23  <xsl:template match="books" mode="detail">  
24    <!-- 省略, 同7-7.xsl对应内容... -->  
25  </xsl:template>  
26 </xsl:stylesheet>
```

## 7.7 XSLT 的命名模版 IV

Code 15: 转换结果

```
1 <html>
2   <body>
3     <h2>图书提供者信息</h2>
4     <p>本图书列表由：<strong>Summer</strong>提供. </p>
5     <h2>按照价格降序排列的图书名称</h2>
6     <!-- 以下内容与7-7.out.html相同，此处省略... -->
7   </body>
8 </html>
```

## 7.8 XSLT 的函数 I

- 可以直接使用 XPath 的函数
- XSLT 本身的函数
  - document()
  - key()
  - format-number()
  - generate-id()
- 示例：

## 7.8 XSLT 的函数 II

Code 16: 样式文档

```
1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   version="2.0">
2   <xsl:template match="/">
3     <html><body>
4       <h2>图书平均价格信息</h2>
5       <xsl:call-template name="priceTemplate"/>
6     </body></html>
7   </xsl:template>
8   <xsl:template name="priceTemplate">
9     平均价格 :
10    <xsl:value-of select="format-number(sum(/books/book/price) div
       count(/books/book/price), '0')"/>
11  </xsl:template>
12 </xsl:stylesheet>
```

## 7.8 XSLT 的函数 III

Code 17: 转换结果

```
1 <html>
2   <body>
3     <h2>图书平均价格信息</h2>
4     平均价格：49
5   </body>
6 </html>
```

## 7.9 XSLT 2.0 新特性

- 新增数据模型
- 支持 Schema 数据类型
- 新增元素和函数
- 支持非 XML 输入源
- 改进了字符处理
- 支持多文件输出

# END

