

— 表の表示 —

Sample22

複数の表のデータをソートする

Sample22 - Microsoft Internet Explorer

ファイル(F) アドレス(A) リンク

価格の高い順にソートされています。

番号	品名	価格
0002	ノートパソコン	300000
0001	デスクトップパソコン	250000
0004	スキャナ	50000
0003	プリンタ	20000

価格の低い順にソートされています。

番号	品名	価格
0003	プリンタ	20000
0004	スキャナ	50000
0001	デスクトップパソコン	250000
0002	ノートパソコン	300000

XML

XSL



CD-ROM収録先:

sample22/sample22.xml

sample22/sample22.xsl

XML sample22.xml

XSLTを使うと、XML文書中の同じ内容を複数回、同一画面上に表示できます。

このXSLTのメリットを活かして、同じ内容を表示することができます。

このサンプルは、複数行の表のデータを、任意のキーを指定して、降順、昇順にソートし、その両方の結果を同一画面上に表示するものです。

<全体>には、4個の<商品>のデータがあります。<商品>が4個しかないことに注目して、ソート結果を見てください。

XSL sample22.xsl

XML文書中の同じ内容を複数回、同一画面上に表示させる場合、<div>タグを使ってブロック分けしておくといでしょう。

価格をキーに、降順でソートした結果を表示させています。

次に、昇順でソートした結果を表示させています。

XML文書中には、「4個の商品しかありませんが、画面上には、降順でソートした4個の商品と、昇順でソートした4個の商品の両方が、表示されています。このように、XSLTコードの書きかた一つで、いろいろな表示が可能になります。

XML sample22.xml

XML文書のデータは、先のsample21(p.76参照)と同じです。

XSL sample22.xsl

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html" version="4.01" encoding="Shift_JIS" indent="yes" />
<xsl:template match="/">
<html lang="ja">
<head>
<title><xsl:value-of select="全体/題名" /></title>
<style type="text/css">~CSS略~</style>
</head>
<body><xsl:apply-templates select="全体" /></body>
</html>
</xsl:template>
<xsl:template match="全体">
<div>
<p>価格の高い順にソートされています。</p>
<table>
<tr><th>番号</th><th>品名</th><th>価格</th></tr>
<xsl:for-each select="商品">
<xsl:sort select="価格" lang="ja" data-type="number" order="descending" />
<tr>
<td><xsl:value-of select="番号" /></td>
<td><xsl:value-of select="品名" /></td>
<td style="text-align:right"><xsl:value-of select="価格" /></td>
</tr>
</xsl:for-each>
</table>
</div>
<div>
<p>価格の低い順にソートされています。</p>
<table>
<tr><th>番号</th><th>品名</th><th>価格</th></tr>
<xsl:for-each select="商品">
<xsl:sort select="価格" lang="ja" data-type="number" order="ascending" />
<tr>
<td><xsl:value-of select="番号" /></td>
<td><xsl:value-of select="品名" /></td>
<td style="text-align:right"><xsl:value-of select="価格" /></td>
</tr>
</xsl:for-each>
</table>
</div>
</xsl:template>
</xsl:stylesheet>
```

— 表の表示 —

Sample23

二者択一の条件分岐

Sample23 - Microsoft Internet Explorer

ファイル(F) アドレス(O) リンク

価格が10万未満については反転表示

番号	品名	価格
0001	デスクトップパソコン	250000
0002	ノートパソコン	300000
0003	プリンタ	20000
0004	スキャナ	50000



CD-ROM収録先：
sample23/sample23.xml
sample23/sample23.xsl

XML sample23.xml

条件分岐によって、書体を変更するのは、カタログページで、特価品を示す場合にも有効な方法です。文字の色を変えてもいいですね。

このサンプルは、XML文書の要素の内容によって条件を指定し、条件に応じて異なる表示結果を得るものです。ここでは、商品価格が100000円未満の価格を反転表示します。

1つ目と2つ目の<商品>の<価格>は、100000円より高価です。
3つ目と4つ目の<商品>の<価格>は、100000円より低価格です。

XSL sample23.xsl

XSLファイル中で、条件分岐の条件を指定する場合は、>は、>と書きます。また、<は、<と書きます。<や>をそのまま使うと、タグの記号として解釈され、正しく入れ子になったXML文書ではなくなります。

<価格>の内容が100000円より大きければ、という条件を指定するには、<xsl:if test="価格[.>100000]">のように書きます。

<価格>の内容が100000円未満であれば、という条件を指定するには、<xsl:if test="価格[.<100000]">のように書きます。価格が100000円未満の場合、反転表示させています。

XML sample23.xml

```
<?xml version="1.0" encoding="Shift_JIS"?>
<?xml-stylesheet type="text/xsl" href="sample23.xsl"?>
<全体>
  <題名>Sample23</題名>
  <商品>
    <番号>0001</番号>
    <品名>デスクトップパソコン</品名>
    <価格>250000</価格>
  </商品>
  <商品>
    <番号>0002</番号>
    <品名>ノートパソコン</品名>
    <価格>300000</価格>
  </商品>
  <商品>
    <番号>0003</番号>
    <品名>プリンタ</品名>
    <価格>20000</価格>
  </商品>
  <商品>
    <番号>0004</番号>
    <品名>スキャナ</品名>
    <価格>50000</価格>
  </商品>
</全体>
```

XSL sample23.xsl

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html" version="4.01" encoding="Shift_JIS" indent="yes" />
<xsl:template match="/">
<html lang="ja">
<head>
<title><xsl:value-of select="全体/題名" /></title>
<style type="text/css"><xsl:comment>
table{border:solid 1pt black;}
th{background-color:silver;padding:5px;}
td{border:solid 1pt black;padding:10px;}
</xsl:comment></style>
</head>
<body>
<div>価格が10万未満については反転表示</div>
<div><xsl:apply-templates select="全体" /></div>
</body>
</html>
</xsl:template>
<xsl:template match="全体">
<table>
<tr><th>番号</th><th>品名</th><th>価格</th></tr>
<xsl:for-each select="商品">
<xsl:if test="価格[.> 100000]">
<tr>
<td><xsl:value-of select="番号" /></td>
<td><xsl:value-of select="品名" /></td>
<td style="text-align:right"><xsl:value-of select="価格" /></td>
</tr>
</xsl:if>
<xsl:if test="価格[.< 100000]">
<tr>
<td><xsl:value-of select="番号" /></td>
<td><xsl:value-of select="品名" /></td>
<td style="text-align:right;background:black;color:white;">
<xsl:value-of select="価格" /></td>
</tr>
</xsl:if>
</xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>
```

XPathのstarts-with関数を使ってみよう

XPathのstarts-with関数は、最初の引数に要素ノードや属性ノードを指定し、2番目の引数に検索キーとなる文字列を指定して、検索キーで始まるデータを抽出するなどの場合に利用できます。この関数は、最初の引数の文字列1が、2番目の引数の文字列2で始まる場合にtrueを返し、そうでない場合はfalseを返します。書式は次の通りです。

戻り値(trueまたはfalse)=starts-with(文字列1, 文字列2)

住所を記述したXMLファイル(starts-with.xml)から、「京都市」の文字列が含まれるデータ(下線部)を抽出してみましょう(CD-ROM column/starts-with.xml , starts-with.xsl)。表示結果は図のようになります。

XSLファイル(starts-with.xsl)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/
1999/XSL/Transform">
<xsl:output method="html" version="4.01" indent="yes" />
<xsl:template match="/">
<html lang="ja"><head><title>start-with関数</title></head>
<body><div><xsl:apply-templates select="全体" /></div></body>
</html>
</xsl:template>
<xsl:template match="全体">
<xsl:for-each select="住所">
    <xsl:variable name="address" select="." />
    <xsl:if test="starts-with($address,'京都市')">
        <div><xsl:value-of select="." /></div>
    </xsl:if>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

「京都市」の含まれる住所が表示されている

