

EXSLT

extensions to XSLT

Mike Raschendörfer

Erweiterungen

- Extension Elements
- Extension Functions

Elemente

- Elemente aus dem xslt Namensraum
- [Extension Elements](#)
- Literal Result Elements

Exstension Elements

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:saxon="http://icl.com/saxon"
  extension-element-prefixes="saxon">

  <xsl:variable name="i" select="0" saxon:assignable="yes"/>
  <xsl:template match="/">
    <saxon:while test="$i &lt; 25">
      Der Wert von i: <xsl:value-of select="$i"/>
      <saxon:assign name="i" select="$i+1"/>
    </saxon:while>
  </xsl:template>
</xsl:stylesheet>
```

element-available

```
<xsl:choose>  
  <xsl:when test="element-available('saxon:while')">  
    Alles OK  
  </xsl:when>  
  <xsl:otherwise>  
    ein Fehler ist aufgetreten  
  </xsl:otherwise>  
</xsl:choose>
```

fallback

```
<saxon:while test="$i &lt; 10">  
  Der Wert von i: <xsl:value-of select="$i"/>  
  <saxon:assign name="i" select="$i+1"/>  
  
  <xsl:fallback>  
    Erweiterungs-Element nicht verfügbar  
  </xsl:fallback>  
</saxon:while>
```

Extension Function

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:math1="java.lang.Math">

  <xsl:template match="/">
    <xsl:value-of select="math1:sin(3.14)"/>
  </xsl:template>

</xsl:stylesheet>
```

function-available

```
<xsl:choose>  
  <xsl:when test="function-available('math:sin')">  
    <xsl:value-of select="math:sin(3.14)" />  
  </xsl:when>  
  <xsl:otherwise>  
    Funktion nicht verfügbar  
  </xsl:otherwise>  
</xsl:choose>
```

Probleme:

Portabilität

Implementation von Erweiterung nicht in
XSLT definierbar

Lösung:

EXSLT ?

XSLT-2.0 ?

EXSLT-Module

- Dates and Times
- Dynamic
- Common
- Functions
- Math
- Regular Expressions
- Sets
- Strings

Implementierung

- XSLT Templates
- EXSLT Funktion
- mit Programmiersprache
- nativ im XSLT-Prozessor

XSLT-Templates

```
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:math="http://exslt.org/math"  
  extension-element-prefixes="math">
```

```
<xsl:import href="math.min.template.xsl"/>
```

```
...
```

```
<xsl:call-template name="math:max">
```

```
...
```

```
</xsl:call-template>
```

```
...
```

```
</xsl:stylesheet>
```

EXSLT-Templates

```
<values>
```

```
  <value>7</value>
```

```
  <value>11</value>
```

```
  <value>8</value>
```

```
</values>
```

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

```
  Maximum:
```

```
  <xsl:call-template name="math:max">
```

```
    <xsl:with-param name="nodes" select="value" />
```

```
  </xsl:call-template>
```

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

EXSLT-Funktionen

```
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:math="http://exslt.org/math"  
  extension-element-prefixes="math">
```

```
<xsl:import href="math.max.function.xsl" />
```

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

Maximum:

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

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

```
</xsl:stylesheet>
```

Programmiersprachen

```
<xsl:stylesheet version="1.1"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:math="http://exslt.org/math"  
  xmlns:func="http://exslt.org/functions"  
  extension-element-prefixes="math func">
```

```
<func:script implements-prefix="math"  
  language="exslt:javascript"  
  src="math.min.js"/>
```

...

```
</xsl:stylesheet>
```

Packages

Function-Level Packages

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:math="http://exslt.org/math" extension-element-prefixes="math">
<xsl:import href="math.min.xsl" />
...
</xsl:stylesheet>
```

Module-Level Packages

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:math="http://exslt.org/math" extension-element-prefixes="math">
<xsl:import href="math.xsl" />
...
</xsl:stylesheet>
```

dates and times

- date:add()
- date:add-duration()
- date:date()
- date:date-time()
- date:day-abbreviation()
- date:day-in-month()
- date:day-in-week()
- date:day-in-year()
- date:day-name()
- date:day-of-week-in-month()
- date:difference()
- date:duration()
- date:format-date()
- date:hour-in-day()
- date:leap-year()
- date:minute-in-hour()
- date:month-abbreviation()
- date:month-in-year()
- date:month-name()
- date:parse-date()
- date:second-in-minute()
- date:seconds()
- date:sum()
- date:time()
- date:week-in-month()
- date:week-in-year()
- date:year()
- date:date-format

math

- `math:abs()`
- `math:acos()`
- `math:asin()`
- `math:atan()`
- `math:atan2()`
- `math:constant()`
- `math:cos()`
- `math:exp()`
- `math:highest()`
- `math:log()`
- `math:lowest()`
- `math:max()`
- `math:min()`
- `math:power()`
- `math:random()`
- `math:sin()`
- `math:sqrt()`
- `math:tan()`

sets

- `set:difference()`
- `set:distinct()`
- `set:has-same-node()`
- `set:intersection()`
- `set:leading()`
- `set:trailing()`

strings

- `str:align()`
- `str:concat()`
- `str:decode-uri()`
- `str:encode-uri()`
- `str:padding()`
- `str:replace()`
- `str:split()`
- `str:tokenize()`

regular expression

- `regexp:match()`
- `regexp:replace()`
- `regexp:test()`

dynamic

dynamische Auswertung von Zeichenfolgen,
die XPath Ausdrücke enthalten

- `dyn:closure()`
- `dyn:evaluate()`
- `dyn:map()`
- `dyn:max()`
- `dyn:min()`
- `dyn:sum()`

dynamic

Bsp.:

```
<dummy attr="/dummy"/>
```

```
<xsl:variable name="foo" select="@attr"/>
```

```
<xsl:variable name="bar" select="dyn:evaluate($foo)"/>
```

common

- `exsl:node-set()`
- `exsl:object-type()`
- `exsl:document`

```
<xsl:template match="/">
  <html>
    <head><title>Frame example</title></head>
    <frameset cols="20%, 80%">
      <frame src="toc.html"/>
      <exsl:document href="toc.html">
        <html>
          <head><title>Table of Contents</title></head>
          <body>
            <xsl:apply-templates mode="toc" select="*" />
          </body>
        </html>
      </exsl:document>
      <frame src="body.html"/>
      <exsl:document href="body.html">
        <html>
          <head><title>Body</title></head>
          <body>
            <xsl:apply-templates select="*" />
          </body>
        </html>
      </exsl:document>
    </frameset>
  </html>
</xsl:template>
```

function

- func:function
- func:result
- func:script

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:math="http://exslt.org/math"
    xmlns:exsl="http://exslt.org/functions"
    extension-element-prefixes="exsl"
    exclude-result-prefixes="math">
<exsl:function name="math:min">
  <xsl:param name="nodes" select="/.." />
  <xsl:choose>
    <xsl:when test="not($nodes)">
      <exsl:return select="number('NaN')" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:for-each select="$nodes">
        <xsl:sort data-type="number" />
        <xsl:if test="position() = 1">
          <exsl:return select="number(.)" />
        </xsl:if>
      </xsl:for-each>
    </xsl:otherwise>
  </xsl:choose>
</exsl:function>

</xsl:stylesheet>
```

XSLT-Parser

- nxsIt <http://www.tkachenko.com/dotnet/nxsIt.html>
- xsltproc <http://xmlsoft.org/XSLT/xsltproc2.html>
- exMsxsl <https://sourceforge.net/projects/fxsl>
- Xalan-c <http://xml.apache.org/xalan-c/>
- Xalan-j <http://xml.apache.org/xalan-j/>
- jd.xslt <http://www.aztecrider.com/xslt/index.html>
- XT <http://www.blz.com/xt/index.html>
- Unicorn http://www.unicorn-enterprises.com/products_uxt.html
- XML::XSL <http://xmlxslt.sourceforge.net/>
- iXSLT <http://www.infoteria.com/index.jsp>
- Saxon <http://saxon.sourceforge.net/>
- Sablotron http://www.gingerall.com/charlie/ga/xml/p_sab.xml
- Transformiix <http://www.mozilla.org/projects/xslt/>
- FastXML <http://www.geocities.com/fastxml/>
- 4Suite <http://4suite.org/index.xhtml>

Links

- [http://www.exslt.org/
community](http://www.exslt.org/community)
- <http://www-106.ibm.com/developerworks/library/x-exslt.html>
- <http://www-106.ibm.com/developerworks/library/x-xdexslt.html>
kleine Einführung in exslt
- <http://www.xmlpitstop.com/Default.asp?DataType=SSCEX>
EXSLT-Beispiele