

XSLT



XSL je kratica od EXTensible Stylesheet Language.

World Wide Web Consortium (W3C) započela je razvitak XSL jer postoji potreba za XML-utemeljenom jeziku stilskog uzorka.

XSL = XML Style Sheets

XML ne koristi unaprijed definirane tagove, pa njihovo značenje nije lako razumjeti.

Na primjer, <table> element može značiti tag od HTML tablice, komad namještaka ili nešto sasvim drugo, pa preglednik (browser) ne zna kako će to prikazati.

XSL **opisuje** kako će se XML document trebati prikazati i kakve elemente posjeduje!

XSL – je više od jezika stilskog uzorka

XSL sastoji se od tri dijela:

- XSLT – jezika za transformaciju XML dokumenata
- XPath - jezika za pozicioniranje (navigaciju) u XML dokumentima
- XSL-FO - jezika za formatiranje XML dokumenata

XSLT je jezik za transformaciju XML dokumenata u XHTML dokumente ili za druge XML dokumente.

XPath je jezik za navigaciju u XML dokumentu.

What is XSLT?

- XSLT znači XSL Transformations
- XSLT je najvažniji dio od XSL
- XSLT stabilno transformira XML dokument u drugi XML dokument (npr. XHTML)
- XPath za navigaciju u XML dokumentima
- XSLT je W3C preporuka

U transformacijskom procesu, XSLT koristi XPath da bi definirao dijelove izvornog dokumenta koji odgovaraju jednom ili više unaprijed definiranih uzoraka (predefined templates). Kad se odgovarajući uzorak podudara, XSLT će transformirati taj dio izvornog dokumenta u rezultirajući dokument.

XSLT je Web Standard

XSLT je nastao W3C preporukom od 16. studenoga 1999.

Svi današnji preglednici (browsers) podržavaju XML i XSLT.

Ispravna deklaracija stilskog uzorka

Korijenski (root) element koji deklarira dokument da bude XSL stilski uzorak je `<xsl:stylesheet>` ili `<xsl:transform>`. To su dva potpuna sinonima.

Ispravan način za deklaraciju po preporuci W3C je:

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

ili:

```
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

Da bi se pristupilo XSLT elementima, atributima i značajkama potrebno je deklarirati prostor XSLT imena na vrhu dokumenta.

Ako se koristi ovaj prostor imena, onda se mora također uključiti i atribut `version="1.0"`.

XML Dokument

Ako se želi **transformirati** sljedeći XML dokument ("cdcatalog.xml") u XHTML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
```

```
</cd>
.
.
.
</catalog>
```

Gledanje XML datoteke u Firefox i Internet Explorer pregledniku: Otvoriti XML datoteku (uobičajeno je klikomna link, poveznicu) - XML dokument će se prikazati u boji označenim korijenom i njegovim elementima, djecom (child elements). Znak plusa (+) ili minusa (-) s lijeve strane elementa može se klikom ekspanirati (razvući) ili zgsnuti u strukturu elementa. Gledanje izvornog koda, čistog teksta, bez + i - znakova, postiže se izborom "View Page Source" ili "View Source" iz preglednikove padajućih ponuda.

Stvaranje XSL stilskog uzorka

Neka XSL stilski uzorak ("cdcatalog.xsl") bude transformacijski predložak:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Title</th>
        <th align="left">Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
      </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Povezivanje XSL stilskog uzorka s XML dokumentom

Treba dodati XSL referencu unutar XML dokumenta ("cdcatalog.xml"):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>
<catalog>
```

```
<cd>
  <title>Empire Burlesque</title>
  <artist>Bob Dylan</artist>
  <country>USA</country>
  <company>Columbia</company>
  <price>10.90</price>
  <year>1985</year>
</cd>
.
.
.
</catalog>
```

Ako se gleda kroz preglednik koji podržava XSLT dogodit će se vrlo lijepa pretvorba XML-a XHTML.

XSL stilski uzorak sastoji se od jednog ili više skupova pravila koji se zovu predlošci (templates).

Svaki predložak sadrži pravila koja će se primijeniti na određenom čvoru u stablu dokumenta, koji odgovarate čvoru u uzorku.

<xsl:template> Element

<xsl:template> element se koristi u gradnji predložaka (templates).

Atribut podudaranja (**match** attribute) koristi se za pridruživanje predloška XML elementu. Atribut podudaranja se također može definirati za čitav XML dokument. Vrijednost match atributa je XPath izraz (npr. match="/" definira cijeli dokument).

Pojednostavljena verzija XSL datoteke prethodnog primjera:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
      </tr>
      <tr>
        <td>.</td>
        <td>.</td>
      </tr>
```

```
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Budući da je XSL stilski uzorak jedan XML dokument on uvijek počinje sa XML deklaracijom, npr.: **<?xml version="1.0" encoding="ISO-8859-1"?>**.

Sljedeći element, **<xsl:stylesheet>**, definira da je ovaj dokument XSLT stilski predložak document (along with the version number and XSLT namespace attributes).

Element **<xsl:template>** definira predložak. Podudaranje **match="/"** atribut pridružuje predložak korijenu XML izvornog dokumenta.

Sadržaj unutar **<xsl:template>** elementa definira neke HTML elemente koji se ispisuju na izlaz.

Posljednje dvije linije definiraju kraj predloška i konac stilskog uzorka.

Rezultat ove transformacije izgledat će ovako:

My CD Collection

Title	Artist
.	.

Rezultat ovog primjera je malo razočaravajući, budući da nema snimanja podataka iz XML dokumenta na izlaz.

<xsl:value-of> element

Element <xsl:value-of> služi za izlučivanje, vađenje, vrijednosti izabranog čvora u stalbu XML dokumenta.

Neka se doda **<xsl:value-of>** elementa za vađenje vrijednosti nekog XML element i načini transformacija na izlaz:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
```

```

</tr>
<tr>
  <td><xsl:value-of select="catalog/cd/title"/></td>
  <td><xsl:value-of select="catalog/cd/artist"/></td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Primjedba: vrijednost **select** atributa je XPath izraz. XPath izraz radi slično kao nagicaja u datotečnom sustavu (file system) gdje se sa znakom udesno kose crte (forward slash (/)) selektiraju podmape (subfolders, subdirectories).

Rezultat transformacije izgledat će ovog puta ovako:

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan

Rezultat je i opet malo razočaravajući, jer se u izlazni dokument presnimila samo jedan podatak ulaznog XML dokumenta.

<xsl:for-each> element

Element <xsl:for-each> dopušta izvođenje petlje u XSLT-u.

XSL element <xsl:for-each> može se koristiti za selektiranje i postavljanje svakog XML elementa određenog čvora:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="artist"/></td>
        </tr>

```

```

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

```

Rezultat transformacije izgledat će ovog puta ovako:

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary More
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many
For the good times	Kenny Rogers
Big Willie style	Will Smith
Tupelo Honey	Van Morrison
Soulsville	Jorn Hoel
The very best of	Cat Stevens
Stop	Sam Brown
Bridge of Spies	T` Pau
Private Dancer	Tina Turner
Midt om natten	Kim Larsen
Pavarotti Gala Concert	Luciano Pavarotti
The dock of the bay	Otis Redding
Picture book	Simply Red
Red	The Communards
Unchain my heart	Joe Cocker

Filtriranje izlaza

Također je moguće filtrirati izlaz iz XML datoteke dodavajući kriterij za odabiranje atributa u `<xsl:for-each>` elementu.

`<xsl:for-each select="catalog/cd[artist='Bob Dylan']">`

Dopušteni filterski operatori su:

- = (equal)
- != (not equal)
- < less than
- > greater than

Podršavanjem XSL stilskog uzorka:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
    <body>
      <h2>My CD Collection</h2>
      <table border="1">
        <tr bgcolor="#9acd32">
          <th>Title</th>
          <th>Artist</th>
        </tr>
        <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
          <tr>
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="artist"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

dobit će se ovakav rezultat transformacije:

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan

Element `<xsl:sort>` služi za slaganje, sortiranje izlaza.

Za slaganje izlaza treba jednostavno dodati `<xsl:sort>` element unutar `<xsl:for-each>` elementa u XSL datoteku:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <xsl:sort select="artist"/>
        <tr>
          <td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="artist"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Primjedba: `select` atribut pokazuje koji XML element se sortira.

Rezultat transformacije izgledat će ovako:

My CD Collection

Title	Artist
Romanza	Andrea Bocelli
One night only	Bee Gees
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
The very best of	Cat Stevens
Greatest Hits	Dolly Parton
Sylvias Mother	Dr.Hook
Eros	Eros Ramazzotti
Still got the blues	Gary Moore

Unchain my heart	Joe Cocker
Soulsville	Jorn Hoel
For the good times	Kenny Rogers
Midt om natten	Kim Larsen
Pavarotti Gala Concert	Luciano Pavarotti
1999 Grammy Nominees	Many
The dock of the bay	Otis Redding
When a man loves a woman	Percy Sledge
Maggie May	Rod Stewart
Stop	Sam Brown
Black angel	Savage Rose
Picture book	Simply Red
Bridge of Spies	T` Pau
Red	The Communards
Private Dancer	Tina Turner
Tupelo Honey	Van Morrison
Big Willie style	Will Smith

<xsl:if> element

Element <xsl:if> se koristi za stavljanje uvjetne provjere na sadržaj XML datoteke.

Sintaksa je ovakva:

```
<xsl:if test="expression">
  ...
  ...some output if the expression is true...
  ...
</xsl:if>
```

<xsl:if> element dodaje se unutar <xsl:for-each> elementa u XSL datoteci:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
```

```

<tr bgcolor="#9acd32">
  <th>Title</th>
  <th>Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
  <xsl:if test="price > 10">
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="artist"/></td>
    </tr>
  </xsl:if>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Primjedba: Vrijednost zahtijevanog **test** atributa sadrži izraz koji se izračunava.

Ovako napisani kod prosljedit će na izlaz samo elemente s naslovom i autorom od Cd-a kojima je cijena veća od 10.

Rezultat transformacije izgledat će ovako:

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Still got the blues	Gary Moore
One night only	Bee Gees
Romanza	Andrea Bocelli
Black Angel	Savage Rose
1999 Grammy Nominees	Many

<xsl:choose> element

Element <xsl:choose> koristi se u sprezi sa elementima <xsl:when> i <xsl:otherwise> da bi se izrazili višestruki test uvjeti.

Sintaksa je ovakva:

```

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

```

```
... some output ....  
</xsl:otherwise>  
</xsl:choose>
```

Umetanje višestrukog uvjetnog testa u XML datoteku izgleda ovako:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
<xsl:template match="/">  
  <html>  
    <body>  
      <h2>My CD Collection</h2>  
      <table border="1">  
        <tr bgcolor="#9acd32">  
          <th>Title</th>  
          <th>Artist</th>  
        </tr>  
        <xsl:for-each select="catalog/cd">  
          <tr>  
            <td><xsl:value-of select="title"/></td>  
            <xsl:choose>  
              <xsl:when test="price > 10">  
                <td bgcolor="#ff00ff">  
                  <xsl:value-of select="artist"/></td>  
              </xsl:when>  
              <xsl:otherwise>  
                <td><xsl:value-of select="artist"/></td>  
              </xsl:otherwise>  
            </xsl:choose>  
          </tr>  
        </xsl:for-each>  
      </table>  
    </body>  
  </html>  
</xsl:template>  
</xsl:stylesheet>
```

S ovakvim kodom naćinit će se transformacija koja će s ljubićastom pozadinom obojiti stupac "Artist" kada (WHEN) je cijena CD-a veća od 10.

Rezultat transformacije izgledat će ovako:

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore

Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many
For the good times	Kenny Rogers
Big Willie style	Will Smith
Tupelo Honey	Van Morrison
Soulsville	Jorn Hoel
The very best of	Cat Stevens
Stop	Sam Brown
Bridge of Spies	T` Pau
Private Dancer	Tina Turner
Midt om natten	Kim Larsen
Pavarotti Gala Concert	Luciano Pavarotti
The dock of the bay	Otis Redding
Picture book	Simply Red
Red	The Communards
Unchain my heart	Joe Cocker

Sljedeći primjer

Slijedi još jedan primjer koji sadrži dva <xsl:when> elementa:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <xsl:choose>
```

```

<xsl:when test="price > 10">
  <td bgcolor="#ff00ff">
    <xsl:value-of select="artist"/></td>
</xsl:when>
<xsl:when test="price > 9">
  <td bgcolor="#cccccc">
    <xsl:value-of select="artist"/></td>
</xsl:when>
<xsl:otherwise>
  <td><xsl:value-of select="artist"/></td>
</xsl:otherwise>
</xsl:choose>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Taj kod će načiniti isto kao prethodni sa stupcima u kojima je cijena CD-a veća od 10, a sa sivom podlogom označit će stupce kada je cijena CD-a veća od 9, a manja od 10.

Rezultat transformacije izgledat će ovako:

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many
For the good times	Kenny Rogers
Big Willie style	Will Smith
Tupelo Honey	Van Morrison
Soulsville	Jorn Hoel
The very best of	Cat Stevens

Stop	Sam Brown
Bridge of Spies	T` Pau
Private Dancer	Tina Turner
Midt om natten	Kim Larsen
Pavarotti Gala Concert	Luciano Pavarotti
The dock of the bay	Otis Redding
Picture book	Simply Red
Red	The Communards
Unchain my heart	Joe Cocker

<xsl:apply-templates> element

Element <xsl:apply-templates> primjenjuje predložak na trenutačni element ili na čvorove djece (child nodes) trenutačnog elementa.

Ako se doda atribut za selektiranje na <xsl:apply-templates> element on će obraditi samo djecu elementa koji sde podudara s vrijednosti atributa. Selekt atribut se koristi da se specificira redosljed po kojem se čvorovi s djecom elementa obrađuju.

Neka se promotri sljedeći XSL uzorak:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h2>My CD Collection</h2>
<xsl:apply-templates/>
</body>
</html>
</xsl:template>
<xsl:template match="cd">
<p>
<xsl:apply-templates select="title"/>
<xsl:apply-templates select="artist"/>
</p>
</xsl:template>
<xsl:template match="title">
Title: <span style="color:#ff0000">
<xsl:value-of select="."/></span>
<br />
</xsl:template>
<xsl:template match="artist">
Artist: <span style="color:#00ff00">
<xsl:value-of select="."/></span>
<br />
</xsl:template>
```

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

Rezultat transformacije izgledat će ovako:

My CD Collection

Title: **Empire Burlesque**
Artist: **Bob Dylan**

Title: **Hide your heart**
Artist: **Bonnie Tyler**

Title: **Greatest Hits**
Artist: **Dolly Parton**

Title: **Still got the blues**
Artist: **Gary Moore**

Title: **Eros**
Artist: **Eros Ramazzotti**

Title: **One night only**
Artist: **Bee Gees**

Title: **Sylvias Mother**
Artist: **Dr.Hook**

Title: **Maggie May**
Artist: **Rod Stewart**

Title: **Romanza**
Artist: **Andrea Bocelli**

Title: **When a man loves a woman**
Artist: **Percy Sledge**

Title: **Black angel**
Artist: **Savage Rose**

Title: **1999 Grammy Nominees**
Artist: **Many**

Title: **For the good times**
Artist: **Kenny Rogers**

Title: **Big Willie style**
Artist: **Will Smith**

Title: **Tupelo Honey**
Artist: **Van Morrison**

Title: **Soulsville**
Artist: **Jorn Hoel**

Title: **The very best of**
Artist: **Cat Stevens**

Title: **Stop**
Artist: **Sam Brown**

Title: **Bridge of Spies**
Artist: **T`Pau**

Title: **Private Dancer**
Artist: **Tina Turner**

Title: **Midt om natten**
Artist: **Kim Larsen**

Title: **Pavarotti Gala Concert**
Artist: **Luciano Pavarotti**

Title: **The dock of the bay**
Artist: **Otis Redding**

Title: **Picture book**
Artist: **Simply Red**

Title: **Red**
Artist: **The Communards**

Title: **Unchain my heart**
Artist: **Joe Cocker**

Ako preglednik to podržava, XSLT se može koristiti za transformaciju XML dokumenta u XHTML unutar preglednika.

XML i XSLT datoteka

Ako se pogleda XML dokument:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
```

```

    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
.
.
.
</catalog>

```

I pripadni XSL stilski uzorak:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Title</th>
        <th align="left">Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title" /></td>
          <td><xsl:value-of select="artist" /></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Primjećuje se da XML datoteka nema referencu na XSL datoteku.

Važno: To znači da se XML datoteka može transformirati koristeći različite XSL stilske uzorke.

Transformacija XML u XHTML na Serveru

Ovdje je ASP source kod potreban za transformaciju XML datoteke u XHTML na serveru:

```

<%
'Load XML
set xml = Server.CreateObject("Microsoft.XMLDOM")
xml.async = false

```

```
xml.load(Server.MapPath("cdcatalog.xml"))

'Load XSL
set xsl = Server.CreateObject("Microsoft.XMLDOM")
xsl.async = false
xsl.load(Server.MapPath("cdcatalog.xsl"))

'Transform file
Response.Write(xml.transformNode(xsl))
%>
```

XML Editor

Danas je XML važna tehnologija i razvitak projekata koji koriste na XML-utemeljenim tehnologijama poput:

- XML Schema za definiranje XML struktura i tipova podataka
- XSLT za transformaciju XML podataka
- SOAP za razmjenu XML podataka među različitim aplikacijama
- WSDL za opis web servisa
- RDF za opis web izvorišta (resources)
- XPath i XQuery za pristup XML podacima
- SMIL za definiciju grafike

Da bi se napisao XML dokument bez pogreški, potrebno je koristiti dobar XML editor!

Profesionalni XML editori omogućuju:

- Automatsko dodavanje završnih tag-ova na sve otvorene tag-ove
- Usmjeruje na pisanje valjanog XML-a
- Provjerava XML prema DTD
- Provjerava XML prema Schemi
- Bojom označuje XML sintaksu

XSL uključuje 3 jezika: XSLT, XPath i XSL-FO.

XPath

XPath se koristi za navigaciju kroz elemente i attribute u XML dokumentu.

XPath je glavni element u W3C-u za XSL standard. Razumijevanje Xpath-a je temelj za napredno korištenje XML-a.

Bez znanja Xpath-a, ne može se načiniti XSLT dokument.

XSL-FO

XSL-FO opisuje formatiranje XML podataka za izlaz na zaslon, papir ili neki drugi medij.

XSL-FO dokument su XML datoteke za informaciju o vanjskom izgledu i vanjskom sadržaju.

XSLT elementi

Elementi	Opis	IE	N
<u>apply-imports</u>	Applies a template rule from an imported style sheet	6.0	
<u>apply-templates</u>	Applies a template rule to the current element or to the current element's child nodes	5.0	6.0
<u>attribute</u>	Adds an attribute	5.0	6.0
<u>attribute-set</u>	Defines a named set of attributes	6.0	6.0
<u>call-template</u>	Calls a named template	6.0	6.0
<u>choose</u>	Used in conjunction with <when> and <otherwise> to express multiple conditional tests	5.0	6.0
<u>comment</u>	Creates a comment node in the result tree	5.0	6.0
<u>copy</u>	Creates a copy of the current node (without child nodes and attributes)	5.0	6.0
<u>copy-of</u>	Creates a copy of the current node (with child nodes and attributes)	6.0	6.0
<u>decimal-format</u>	Defines the characters and symbols to be used when converting numbers into strings, with the format-number() function	6.0	
<u>element</u>	Creates an element node in the output document	5.0	6.0
<u>fallback</u>	Specifies an alternate code to run if the processor does not support an XSLT element	6.0	
<u>for-each</u>	Loops through each node in a specified node set	5.0	6.0
<u>if</u>	Contains a template that will be applied only if a specified condition is true	5.0	6.0
<u>import</u>	Imports the contents of one style sheet into another. Note: An imported style sheet has lower precedence than the importing style sheet	6.0	6.0
<u>include</u>	Includes the contents of one style sheet into another. Note: An included style sheet has the same precedence as the including style sheet	6.0	6.0
<u>key</u>	Declares a named key that can be used in the style sheet with the key() function	6.0	6.0
<u>message</u>	Writes a message to the output (used to report errors)	6.0	6.0
<u>namespace-alias</u>	Replaces a namespace in the style sheet to a different namespace in the output	6.0	
<u>number</u>	Determines the integer position of the current node and formats a number	6.0	6.0
<u>otherwise</u>	Specifies a default action for the <choose> element	5.0	6.0
<u>output</u>	Defines the format of the output document	6.0	6.0
<u>param</u>	Declares a local or global parameter	6.0	6.0

<u>preserve-space</u>	Defines the elements for which white space should be preserved	6.0	6.0
<u>processing-instruction</u>	Writes a processing instruction to the output	5.0	6.0
<u>sort</u>	Sorts the output	6.0	6.0
<u>strip-space</u>	Defines the elements for which white space should be removed	6.0	6.0
<u>stylesheet</u>	Defines the root element of a style sheet	5.0	6.0
<u>template</u>	Rules to apply when a specified node is matched	5.0	6.0
<u>text</u>	Writes literal text to the output	5.0	6.0
<u>transform</u>	Defines the root element of a style sheet	6.0	6.0
<u>value-of</u>	Extracts the value of a selected node	5.0	6.0
<u>variable</u>	Declares a local or global variable	6.0	6.0
<u>when</u>	Specifies an action for the <choose> element	5.0	6.0
<u>with-param</u>	Defines the value of a parameter to be passed into a template	6.0	6.0

XQuery 1.0, XPath 2.0, i XSLT 2.0 dijele istu funkcijsku knjižnicu.

XSLT Functions

XSLT uključuje preko 100 ugrađenih (built-in) funkcija. Postoje funkcije za string vrijednosti, numeričke vrijednosti, usporedbu datuma i vremena, manipulaciju nizovima, Boolove vrijednosti i drugo.

Name	Description
<u>current()</u>	Returns the current node
<u>document()</u>	Used to access the nodes in an external XML document
<u>element-available()</u>	Tests whether the element specified is supported by the XSLT processor
<u>format-number()</u>	Converts a number into a string
<u>function-available()</u>	Tests whether the function specified is supported by the XSLT processor
<u>generate-id()</u>	Returns a string value that uniquely identifies a specified node
<u>key()</u>	Returns a node-set using the index specified by an <xsl:key> element
<u>system-property()</u>	Returns the value of the system properties
<u>unparsed-entity-uri()</u>	Returns the URI of an unparsed entity