XSL

XML – teorie a praxe značkovacích jazyků (IZI238)

Jirka Kosek

Poslední modifikace: $Date: 2005/12/01 09:35:37 $

Copyright © 2001-2005 Jiří Kosek

Princip XSL

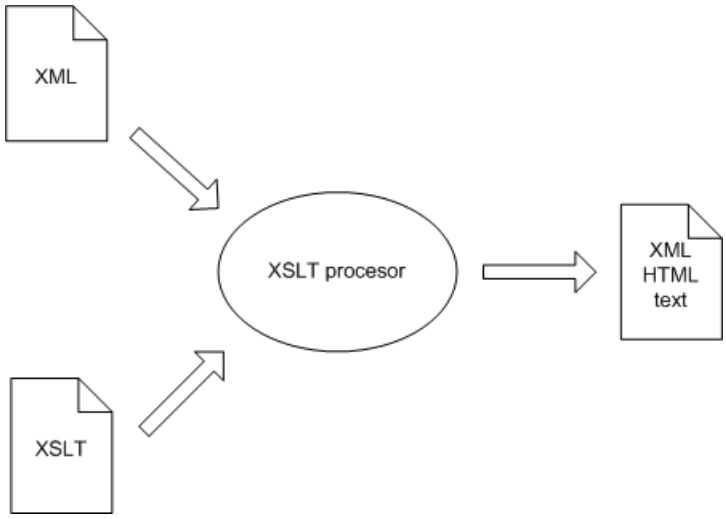
Copyright © 2001-2005 Jiří Kosek

XSL

- **XSL = eXtensible Stylesheet Language**
- **stylový jazyk speciálně vyvinutý pro XML**
- **obsahuje dvě samostatné části – XSLT a FO**
- **transformační jazyk XSLT**
  - **umožňuje popsat transformaci z XML do XML, HTML nebo čistého textu**
- **formátovací objekty (FO)**
  - **abstraktní popis vzhledu dokumentu využívající bohatý formátovací slovník**
  - **interpretované FO se zobrazí na obrazovce, převedou do PDF, PS apod.**

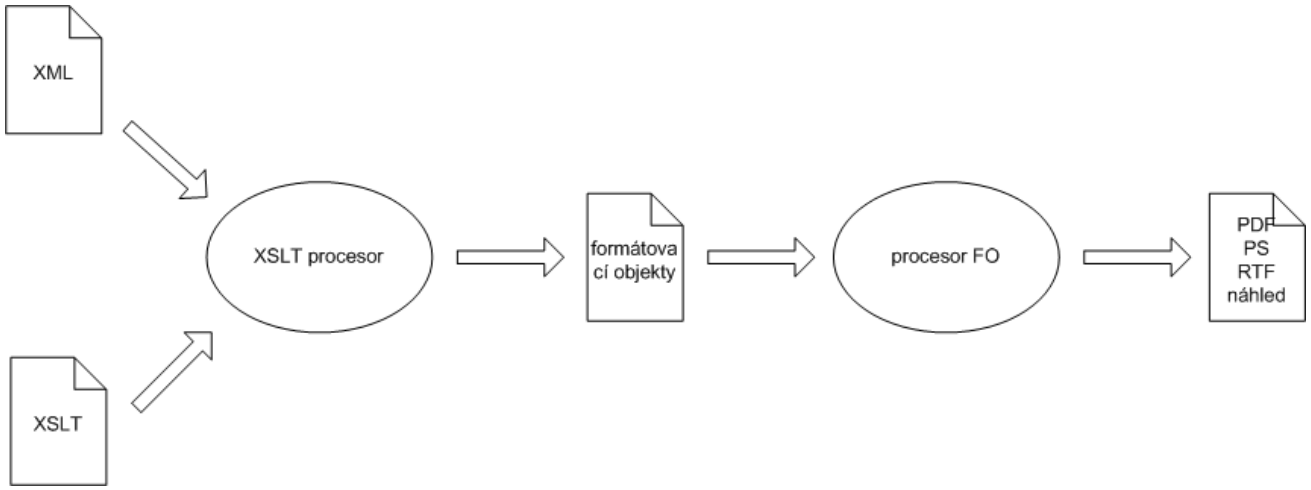
Copyright © 2001-2005 Jiří Kosek

Princip XSLT transformace



Copyright © 2001-2005 Jiří Kosek

Princip použití FO



Copyright © 2001-2005 Jiří Kosek

XSLT

- **standard W3C od roku 1999**
- **styl obsahuje šablony, které určují, jak se budou jednotlivé části dokumentu převádět**
- **části dokumentu jsou v šablonách vybírány pomocí jazyka XPath**
- **kromě výkonného mechanismu šablon lze používat podmínky, cykly, proměnné, funkce, třídění části XML dokumentu, …**
- **styl je sám o sobě XML dokumentem, který obsahuje dva druhy značek**
  - **instrukce pro XSLT procesor**
  - **značky výstupního formátu (HTML, FO, XML)**
  - **k odlišení se používají jmenné prostory**

Copyright © 2001-2005 Jiří Kosek

XPath

Copyright © 2001-2005 Jiří Kosek

XPath

- **jednoduchý dotazovací jazyk**
- **využívá se v XSLT, XPointeru, XML schématech a dalších jazycích, proto tvoří samostatný standard**
- **XPath se dotazuje nad stromovou reprezentací dokumentu**
  - **jednotlivé elementy a atributy tvoří uzly stromu**
- **XPath výraz nejčastěji vybírá ze stromu určitou množinu uzlů**

Copyright © 2001-2005 Jiří Kosek

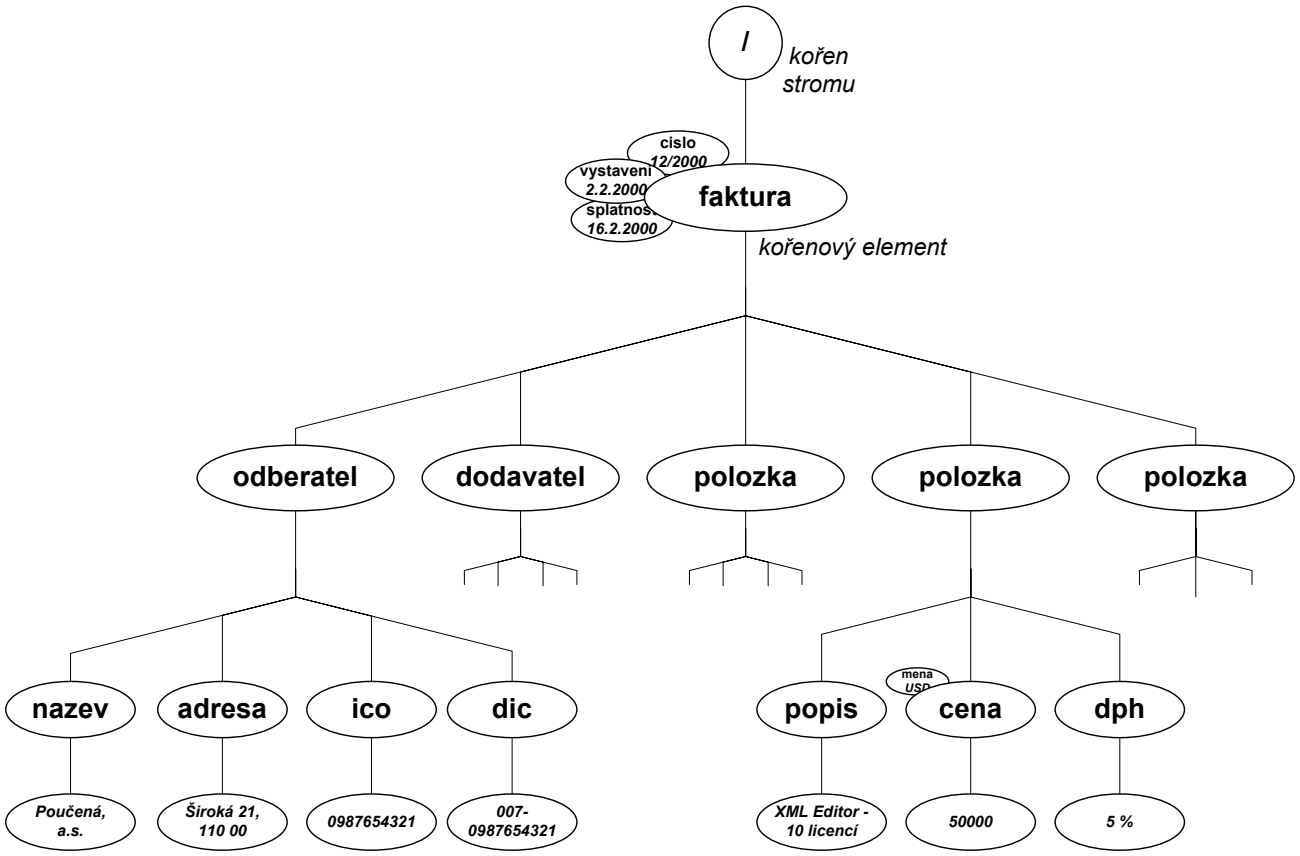
Stromová reprezentace dokumentu

Příklad 1. Ukázkový dokument

```
<faktura vystaveni="2.2.2000" splatnost="16.2.2000"
         cislo="12/2000">
  <odberatel>
    <nazev>Poučená, a.s.</nazev>
    <adresa>Široká 21, Praha 1, 110 00</adresa>
    <ico>0987654321</ico>
    <dic>007-0987654321</dic>
  </odberatel>
  <dodavatel>
    ...
  </dodavatel>
  <polozka>
    ...
  </polozka>
  <polozka>
    <popis>XML Editor - 10 licencí</popis>
    <cena mena="USD">5000</cena>
    <dph>5</dph>
  </polozka>
  <polozka>
    ...
  </polozka>
</faktura>
```


Copyright © 2001-2005 Jiří Kosek

Stromová reprezentace dokumentu (Pokračování)



Copyright © 2001-2005 Jiří Kosek

Výrazy

- vždy se vztahuje k nějakému aktuálnímu uzlu (obvykle kořenový uzel)

para	všechny elementy para, které jsou dětmi
*	všechny elementy, které jsou dětmi
text()	všechny textové uzly, které jsou dětmi
id("pqz")	uzel, který má id nastaveno na "pqz"
@name	atribut name aktuálního uzlu
@*	všechny atributy aktuálního uzlu
para[1]	první element para, který je dítětem akt. uzlu
para[last()]	poslední element para, který je dítětem akt. uz.
*/para	všechny elementy para, které jsou vnoučaty
kapitola//para	všechny elementy para, které jsou potomkem elementu kapitola
//para	všechny elementy para (které jsou potomky kořenového uzlu)
/dokument	element dokument, který je zároveň kořenovým elementem (je přímo pod kořenovým uzlem)
.	aktuální uzel
..	rodič aktuálního uzlu
.//para	všechny elementy para, které jsou potomky aktuálního uzlu
../@lang	atribut lang u rodiče aktuálního uzlu
//para[@type="warning"]	vybere všechny elementy para, které mají atribut type nastaven na "warning"
//para[@type="warning"][5]	vybere pátý element para, který má atribut type nastaven na "warning"

- predikáty v hranatých závorkách jsou vyhodnocovány zleva doprava

Copyright © 2001-2005 Jiří Kosek

Základy XSLT

Copyright © 2001-2005 Jiří Kosek

Nejdůležitější elementy

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

- **<xsl:template match="*XPath*">**
  …
  </xsl:template>

- **<xsl:apply-templates/>**

  hledá další šablony

- **<xsl:value-of select="…"/>**

  vybere pouze text

Copyright © 2001-2005 Jiří Kosek

Podmínky

- **<xsl:if test="*podmínka*">**
  ...
  </xsl:if>

- **náhrada if-then-else**

```
<xsl:choose>
  <xsl:when test="podmínka">
    …
  </xsl:when>
  <xsl:otherwise>
    …
  </xsl:otherwise>
</xsl:choose>
```

- **výběr z více variant**

```
<xsl:choose>
  <xsl:when test="podmínka">
    …
  </xsl:when>
  <xsl:when test="podmínka">
    …
  </xsl:when>
  <xsl:otherwise>
    …
  </xsl:otherwise>
</xsl:choose>
```


Copyright © 2001-2005 Jiří Kosek

Cykly, třídění, číslování

- **iterace přes množinu uzlů**

```
<xsl:for-each select="XPath výraz">
  …
</xsl:for-each>
```

- **setřídění uzlů před zpracováním**

```
<xsl:for-each select="XPath výraz">
  <xsl:sort select="výraz"/>
  …
</xsl:for-each>
```

- **číslování**

```
<xsl:number value="…" format="…"/>
```


Copyright © 2001-2005 Jiří Kosek

Implementace XSLT

- **přímá podpora v prohlížečích**
  - **IE6**
  - **IE5+ – po updatu novou verzí MSXML**
  - **Mozilla 0.9.8+**
- **samostatné XSLT procesory**
  - **Saxon[1] – velké množství funkcí, jeden z nejrychlejších procesorů v Javě**
  - **Xalan[2] – javová i C++ verze**
  - **XT[3] – jedna z prvních implementací; není zcela 100%**
  - **libxslt/xsltproc[4] – velmi rychlá implementace v C**
  - **MSXML – implementace od Microsoftu; velmi rychlá, COM rozhraní**
  - **několik dalších**

[1] http://saxon.sourceforge.net
[2] http://xml.apache.org
[3] http://www.jclark.com/xml/xt.html
[4] http://xmlsoft.org/XSLT/

Copyright © 2001-2005 Jiří Kosek

Architektura FO

Copyright © 2001-2005 Jiří Kosek

Úvod

- **XSL = formátovací objekty + XSLT**
- **formátovací objekty:**
  - **abstraktní popis vzhledu dokumentu (rozvržení stránek + objekty na stránce)**
  - **XML syntaxe**
- **FO objekty definují vzhled vysázeného dokumentu podobně jako HTML popisuje zobrazení stránky v prohlížeči**
- **FO obvykle nepíšeme přímo, ale pomocí XSLT stylu je automaticky vytvoříme z XML dokumentu**
- **pro jeden dokument můžeme mít několik stylů (HTML, FO, apod.)**

Copyright © 2001-2005 Jiří Kosek

Ukázkový dokument s FO

```
<?xml version="1.0" encoding="utf-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
   <fo:layout-master-set>
      <fo:simple-page-master margin-bottom="0.5cm"
                             margin-left="0.5cm"
                             margin-right="0.5cm"
                             margin-top="0.5cm"
                             page-width="9cm"
                             page-height="5cm"
                             master-name="my-master">
         <fo:region-body/>
      </fo:simple-page-master>
   </fo:layout-master-set>
   <fo:page-sequence master-reference="my-master">
      <fo:flow flow-name="xsl-region-body"
               font-family="Times New Roman"
               font-size="8pt">
         <fo:block>
            <fo:float float="right">
               <fo:block margin-left="6pt" margin-bottom="2pt">
                  <fo:external-graphic src="url(kosek.jpg)" ►
content-width="2cm"/>
               </fo:block>
            </fo:float>
            <fo:block font-family="Helvetica"
                      font-size="200%"
                      font-weight="bold">Jirka Kosek</fo:block>
            <fo:block> e-mail: jirka@kosek.cz</fo:block>
            <fo:block space-before="6pt" font-style="italic"
                      text-align="justify" language="cs"
                      hyphenate="true">Je to úplný <fo:inline ►
color="red" font-weight="bold">magor do
               XML</fo:inline>. Už mu z toho asi hráblo, pořád ►
brblá něco o XSLT a XML schématech.
               Ale jinak je <fo:inline color="red" ►
font-weight="bold">převážně neškodný</fo:inline>.
               Napíšeme ještě něco, aby textu bylo více. Ještě ►
více než více. Až ho bude úplně
              nejvíce. Více než nejvíce. Pořád málo. Tak ještě ►
přidáme. Ať má dost. A to by bylo,
               abysedlouhéslovonerozdělilozvláštěkdyž ►
hoprotáhnemeopravduhodně.</fo:block>
         </fo:block>
      </fo:flow>
```


Copyright © 2001-2005 Jiří Kosek

Ukázkový dokument s FO (Pokračování)

```
   </fo:page-sequence>
</fo:root>
```

Jirka Kosek

e-mail: jirka@kosek.cz

*Je to úplný **magor do XML**. Už mu z toho asi hráblo, pořád brblá něco o XSLT a XML schématech. Ale jinak je **převážně neškodný**. Napíšeme ještě něco, aby textu bylo více. Ještě více než více. Až ho bude úplně nejvíce. Více než nejvíce. Pořád málo. Tak ještě přidáme. Ať má dost. A to by bylo, abysedlouhéslovonerozdělilozvláštěkdyž hoprotáhne-meopravduhodně.*

- **spuštění převodu z FO do PDF**

```
xep -fo vizitka.fo
```


Copyright © 2001-2005 Jiří Kosek

Transformace do FO pomoc XSLT stylu

Příklad 2. Zdrojový XML dokument – `vizitka.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<osoba>
  <jmeno>Jirka Kosek</jmeno>
  <email>jirka@kosek.cz</email>
  <foto>kosek.jpg</foto>
  <poznamka>Je to úplný <dulezite>magor do
    XML</dulezite>. Už mu z toho asi hráblo, pořád brblá
    něco o XSLT a XML schématech.  Ale jinak je
    <dulezite>převážně neškodný</dulezite>.  Napíšeme ještě
    něco, aby textu bylo více. Ještě více než více.  Až ho
    bude úplně nejvíce. Více než nejvíce. Pořád málo.  Tak
    ještě přidáme. Ať má dost. A to by bylo,
    abysedlouhéslovonerozdělilozvláštěkdyž
    hoprotáhnemeopravduhodně.</poznamka>
</osoba>
```


Copyright © 2001-2005 Jiří Kosek

Transformace do FO pomoc XSLT stylu (Pokračování)

Příklad 3. XSLT styl pro generování FO – `vizitka.xsl`

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:fo="http://www.w3.org/1999/XSL/Format"
                version="1.0">

<xsl:template match="/">
  <fo:root>
    <fo:layout-master-set>
      <fo:simple-page-master margin-bottom="0.5cm"
                             margin-left="0.5cm"
                             margin-right="0.5cm"
                             margin-top="0.5cm"
                             page-width="9cm"
                             page-height="5cm"
                             master-name="my-master">
        <fo:region-body/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="my-master">
      <fo:flow flow-name="xsl-region-body"
               font-family="Times New Roman"
               font-size="8pt">
        <fo:block>
          <fo:float float="right">
            <fo:block margin-left="6pt" margin-bottom="2pt">
              <fo:external-graphic src="url({osoba/foto})"
                                   content-width="2cm"/>
            </fo:block>
          </fo:float>
          <xsl:apply-templates/>
        </fo:block>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>

<xsl:template match="jmeno">
  <fo:block font-family="Helvetica"
            font-size="200%"
            font-weight="bold">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```


Copyright © 2001-2005 Jiří Kosek

Transformace do FO pomoc XSLT stylu (Pokračování)

```
<xsl:template match="email">
  <fo:block>
    e-mail:
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="foto">
  <!-- Již jsme zpracovali, teď ignorujeme -->
</xsl:template>

<xsl:template match="poznamka">
  <fo:block space-before="6pt" font-style="italic"
            text-align="justify" language="cs"
            hyphenate="true">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="dulezite">
  <fo:inline color="red" font-weight="bold">
    <xsl:apply-templates/>
  </fo:inline>
</xsl:template>

</xsl:stylesheet>
```

- **spuštění převodu z XML do PDF přes FO**

```
saxon -o vizitka.fo vizitka.xml vizitka.xsl
xep -fo vizitka.fo
```

- **nebo v jednom kroku**

```
xep -xml vizitka.xml -xsl vizitka.xsl
```


Copyright © 2001-2005 Jiří Kosek

Nejdůležitější formátovací objekty

`block`	Objekt odpovídá blokovým elementům, které známe z kaskádových stylů. Typicky se používá se pro odstavce, nadpisy apod.
`external-graphic`	Objekt zastupuje obrázek, který je uložen mimo výsledný dokument formátovacích objektů. Obvykle je obrázek uložen v externím souboru (např. GIF, JPEG, PNG, EPS apod.).
`float`	Plovoucí objekt – umístí se na vhodné místo stránky. Obvykle se používá pro obrázky a tabulky případně pro sazbu poznámek vedle textu (marginálií).
`footnote`, `footnote-body`	Objekty se používají pro poznámky pod čarou.
`inline`	Formátovací objekt nezpůsobující vznik nového odstavce. Používá se například pro změny druhu písma uvnitř odstavce.
`leader`	Objekt se používá pro čáry nebo opakované znaky (nejčastěji tečky), které mají vyplnit daný prostor. Používá se například v obsahu pro oddělení názvu kapitoly od čísla strany.
`list-block`, `list-item`, `list-item-body`, `list-item-label`	Objekty se používají pro seznamy.
`basic-link`	Umožňuje do výsledného dokumentu zařadit odkazy.
`table`, `table-*`	Několik objektů, které umožňují vytváření tabulek.
`marker`, `retrieve-marker`	Objekty umožňují vytváření záhlaví a zápatí, které obsahují proměnlivé texty – např. názvy kapitol a podkapitol.
`page-number`, `pagenumber-citation`	Objekty umožňují generování čísla stránky a čísla stránky s určitým objektem.
`wrapper`	Objekt se používá v případech, kdy je potřeba pro několik objektů nastavit společné vlastnosti.

Copyright © 2001-2005 Jiří Kosek

Nejpoužívanější vlastnosti pro formátování textu

font-family	**Použitá rodina písma. Může být uvedeno více hodnot oddělených čárkou. V případě, že nějaké atypické písmenko není v prvním písmu, použije se další se seznamu. Pro toto šikovné chování je potřeba nastavit vlastnost font-selection-strategy na hodnotu `character-by-character`.**
font-size	**Velikost písma. Může být zadána relativně i absolutně. Například: `12pt`, `150%`, `small`, `smaller`.**
font-style	**Normální (`normal`) písmo nebo kurzíva (`italic`).**
font-weight	**„Tloušťka“ písma – `normal`/`bold`.**
color	**Barva textu.**
background-color	**Barva pozadí.**

Copyright © 2001-2005 Jiří Kosek

Dělení slov

language Kód jazyka, který se má použít například pro dělení slov. Pro češtinu je to `cs`.

hyphenate Mají se dělit slova – `true`/`false`. Většinou se používá ve spojení s text-align="`justify`".

- pro správnou činnost dělení slov, je potřeba mít nainstalované vzory dělení slov pro odpovídající jazyk
- vzory ke stažení:
  - XEP[5]
  - FOP[6]
  - XSL Formatter[7]

[5] http://www.kosek.cz/sw/xep/index.html
[6] http://www.kosek.cz/sw/fop/index.html
[7] http://www.kosek.cz/sw/axf/index.html

Copyright © 2001-2005 Jiří Kosek

Nejpoužívanější vlastnosti pro fo:block

text-align	**Zarovnání odstavce – `start`/`end`/`center`/`justify`.**
text-align-last	**Zarovnání poslední řádky odstavce**
text-indent	**Velikost odstavcové zarážky.**
space-before	**Mezera před odstavcem.**
space-after	**Mezera za odstavcem.**
keep-together	**Zakázání stránkového zlomu odstavce (resp. objektu, na kterém je použito). Zákaz zlomu se provede pomocí keep-together="`always`".**
keep-with-next	**Zakázání zlomu mezi odstavcem a následujícím blokem.**
break-before	**Zalomení stránky před odstavcem. break-before="`page`".**
break-after	**Zalomení stránky za odstavcem. break-after="`page`".**

Copyright © 2001-2005 Jiří Kosek

Seznamy

Příklad 4. Ukázka seznamu s odrážkami

```
<fo:list-block provisional-distance-between-starts="1em">
  <fo:list-item>
    <fo:list-item-label end-indent="label-end()">
      <fo:block>&#x2022;</fo:block>
    </fo:list-item-label>
    <fo:list-item-body start-indent="body-start()">
      <fo:block>První položka seznamu</fo:block>
    </fo:list-item-body>
  </fo:list-item>
  <fo:list-item>
    <fo:list-item-label end-indent="label-end()">
      <fo:block>&#x2022;</fo:block>
    </fo:list-item-label>
    <fo:list-item-body start-indent="body-start()">
      <fo:block>Druhá položka seznamu</fo:block>
    </fo:list-item-body>
  </fo:list-item>
</fo:list-block>
```

Nejpoužívanější vlastnosti

provisional-distance-between-starts	**Místo vyhrazené pro odrážku. Musí být nastaveno u `fo:list-block`.**
start-indent	**Odsazení těla seznamu. Zase by mělo být nastaveno všude. Tělo položky seznamu by mělo používat hodnotu `body-start()`.**
end-indent	**Odsazení návěstí zprava. Typicky se používá hodnota `label-end()`.**

Copyright © 2001-2005 Jiří Kosek

Obrázky

Příklad 5. Ukázka vloženého obrázku

```
<fo:external-graphic src="url(foto.jpg)"
                     content-width="5cm"
                     width="40%"
                     text-align="center"/>
```

Nejpoužívanější vlastnosti

src	**URL adresa obrázku. Musí být ve tvaru `url(`*`URL`*`).`**
width, height	**Velikost plochy vyhrazené pro obrázek.**
content-width, content-height	**Skutečná šířka a výška obrázku**
text-align	**Způsob umístění obrázku uvnitř vyhrazené plochy.**

Copyright © 2001-2005 Jiří Kosek

Vkládání SVG obrázků přímo do FO

Příklad 6. Vložení obrázku SVG přímo mezi formátovací objekty

```
<fo:instream-foreign-object content-width="16cm">
  <svg viewBox="0 0 400 400" xmlns="http://www.w3.org/2000/svg">
    <desc>This is a blue circle with a red outline</desc>
    <g>
      <circle style="fill: blue; stroke: red" cx="200" cy="200" ►
r="100"/>
      <text x="150" y="160" stroke="yellow" fill="white"
     font-weight="bold" font-size="20">Hello World</text>
    </g>
  </svg>
</fo:instream-foreign-object>
```


Copyright © 2001-2005 Jiří Kosek

Tabulky

Příklad 7. Ukázka jednoduché tabulky

```
<fo:table>
  <fo:table-body>
    <fo:table-row>
      <fo:table-cell>
        <fo:block>A</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block>B</fo:block>
      </fo:table-cell>
   </fo:table-row>
    <fo:table-row>
      <fo:table-cell>
        <fo:block>C</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block>D</fo:block>
      </fo:table-cell>
   </fo:table-row>
 </fo:table-body>
</fo:table>
```

Nejpoužívanější vlastnosti:

column-width	**Šířka sloupce.**
number-columns-spanned	**Počet sloučených buněk.**
number-rows-spanned	**Počet buněk sloučených vertikálně.**

Copyright © 2001-2005 Jiří Kosek

Poznámky pod čarou

Příklad 8. Ukázka poznámky pod čarou

```
<fo:footnote>
  <fo:inline font-size="70%"
             baseline-shift="super">1</fo:inline>
  <fo:footnote-body>
    <fo:block>
      <fo:inline font-size="70%"
                 baseline-shift="super">1</fo:inline>
      Text poznámky
    </fo:block>
  </fo:footnote-body>
</fo:footnote>
```


Copyright © 2001-2005 Jiří Kosek

Plovoucí objekty

Příklad 9. Ukázka plovoucího objektu

```
<fo:float float="end">
  <fo:block>
    ...plovoucí obsah...
  </fo:block>
</fo:float>
```

Nejpoužívanější vlastnosti:

float Druh plovoucího objektu (`end`, `start`, `before`).

Copyright © 2001-2005 Jiří Kosek

Výplně

Příklad 10. Ukázka výplně

```
<fo:block text-align-last="justify">
  Úvod
  <fo:leader leader-pattern="dots"/>
  1
</fo:block>
```

Nejpoužívanější vlastnosti:

leader-pattern	**Druh výplně (`dots`, `space`, `rule`, `use-content`).**
leader-length	**Délka výplně.**
rule-style	**Druh výplně, pokud se používá čára (`dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`).**
rule-thickness	**Síla čáry.**

Copyright © 2001-2005 Jiří Kosek

Generování obsahu I.

Příklad 11. Ukázkový XML dokument

```
<kniha>
  <kapitola>
    <název>....</název>
    <podkapitola>
      <název>...</název>
      ....
    </podkapitola>
    <podkapitola>
      <název>...</název>
      ....
    </podkapitola>
  </kapitola>
  <kapitola>
    <název>....</název>
    <podkapitola>
      <název>...</název>
      ....
    </podkapitola>
    <podkapitola>
      <název>...</název>
      ....
    </podkapitola>
  </kapitola>
</kniha>
```


Copyright © 2001-2005 Jiří Kosek

Generování obsahu II.

Příklad 12. Řešení pomocí `for:each`

```
<xsl:template match="/">
  ...
  <fo:block>Obsah</fo:block>
  <xsl:for-each select="kniha/kapitola">
    <fo:block text-align-last="justify">
      <fo:basic-link
          internal-destination="{generate-id(.)}">
        <xsl:value-of select="název"/>
      </fo:basic-link>
      <fo:leader leader-pattern="dots"/>
      <fo:page-number-citation ref-id="{generate-id(.)}"/>
    </fo:block>
    <xsl:for-each select="podkapitola">
      <fo:block start-indent="1em"
                text-align-last="justify">
 <fo:basic-link
            internal-destination="{generate-id(.)}">
   <xsl:value-of select="název"/>
 </fo:basic-link>
 <fo:leader leader-pattern="dots"/>
 <fo:page-number-citation ref-id="{generate-id(.)}"/>
      </fo:block>
    </xsl:for-each>
  </xsl:for-each>

  <xsl:apply-templates/>
  ...
</xsl:template>

<xsl:template match="kapitola">
  <fo:block ... id="{generate-id(.)}">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="podkapitola">
  <fo:block ... id="{generate-id(.)}">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```


Copyright © 2001-2005 Jiří Kosek

Generování obsahu III.

Příklad 13. Řešení pomocí režimů

```
<xsl:template match="/">
  ...
  <fo:block>Obsah</fo:block>
  <xsl:apply-templates mode="toc"/>

  <xsl:apply-templates/>
  ...
</xsl:template>

<xsl:template match="kapitola">
  <fo:block ... id="{generate-id(.)}">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="podkapitola">
  <fo:block ... id="{generate-id(.)}">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="kapitola" mode="toc">
  <fo:block text-align-last="justify">
    <fo:basic-link internal-destination="{generate-id(.)}">
      <xsl:value-of select="název"/>
    </fo:basic-link>
    <fo:leader leader-pattern="dots"/>
    <fo:page-number-citation ref-id="{generate-id(.)}"/>
  </fo:block>
  <xsl:apply-templates mode="toc"/>
</xsl:template>

<xsl:template match="podkapitola" mode="toc">
  <fo:block start-indent="1em" text-align-last="justify">
    <fo:basic-link internal-destination="{generate-id(.)}">
      <xsl:value-of select="název"/>
    </fo:basic-link>
    <fo:leader leader-pattern="dots"/>
    <fo:page-number-citation ref-id="{generate-id(.)}"/>
  </fo:block>
</xsl:template>

<xsl:template match="text()" mode="toc"/>
```


Copyright © 2001-2005 Jiří Kosek

Aktuální stav specifikace XSL a podpora v aplikacích

- **standard XSL je doporučením W3C od 15. října 2001**
- **open-source/free implementace:**
  - **FOP**[8]
  - **PassiveTeX**[9]
  - **UFO**[10]
  - **XFC**[11]
  - **jfor**[12]
- **komerční implementace:**
  - **XEP**[13]
  - **Epic**[14]
  - **XSL Formatter**[15]
- **žádná z implementací zatím nepokrývá 100% standard, ale komerční implementace jsou pro většinu aplikací dostačující**
- **licence pro použití plné verze XEPu pro studenty VŠE**[16]

[8] http://xml.apache.org/fop/
[9] http://users.ox.ac.uk/~rahtz/passivetex/
[10] http://www.unicorn-enterprises.com/products_ufo.html
[11] http://www.alphaworks.ibm.com/tech/xfc
[12] http://www.jfor.org/
[13] http://www.renderx.com/FO2PDF.html
[14] http://www.arbortext.com/
[15] http://www.antennahouse.com/xslformatter.html
[16] http://badame.vse.cz/izi238/software.html#xep

Copyright © 2001-2005 Jiří Kosek

Typografické znaky a XML

Copyright © 2001-2005 Jiří Kosek

Typografické znaky a XML

Problém: Jak v XML zapisovat a zpracovávat znaky jako `–' (pomlčka), `„' a `“' (české uvozovky) apod.

Řešení:

1. **Přímý zápis znaků do dokumentu – umí jen některé editory, je možné jen v některých kódováních (utf-8, windows-1250)**

```
<doc>
– „Jak se máš?“
</doc>
```

2. **Zápis pomocí číselné znakové entity – dost nepohodlné**

```
<doc>
&#x2013; &#x201e;Jak se máš?&#x201c;
</doc>
```

3. **Vytvoření interních textových entit pro často používané znaky – rozumný kompromis**

```
<!DOCTYPE doc [
<!ENTITY ndash "&#x2013;">
<!ENTITY lq    "&#x201e;">
<!ENTITY rq    "&#x201c;">
]>
<doc>
&ndash; &lq;Jak se máš?&rq;
</doc>
```

4. **Uvozovky se často řeší speciálním elementem, který se zpracovává až ve stylu**

```
<!DOCTYPE doc [
<!ENTITY ndash "&#x2013;">
]>
<doc>
&ndash; <q>Jak se máš?</q>
</doc>

<xsl:template match="q">
  &#x201e;<xsl:apply-templates/>&#x201c;
</xsl:template>
```

 Copyright © 2001-2005 Jiří Kosek

Typografické znaky a XML (Pokračování)

Pokud jsou uvozovky označeny jako element, může styl použít uvozovky podle aktuálního jazyka nebo automaticky měnit znak uvozovek ve vnořených citacích

 Copyright © 2001-2005 Jiří Kosek