

Semestrální práce "Katalog hotel#"

Dokumentace

Aleš Liberský

Semestrální práce "Katalog hotel#": Dokumentace

Aleš Liberský

Copyright © 2020 Aleš Liberský

Obsah

Úvod	v
1. Ukázkový dokument	1
2. Schema	2
3. Schematron	3
4. Transformace HTML	4
5. FO transformace	7
6. Schematron	8
Rejstřík	9

Seznam příkladů

2.1. Ošetření telefonního čísla	2
2.2. Ošetření počtu volných pokojů	2
3.1. Kontrola počtu dostupných pokojů	3
4.1. Generování jednotlivých souborů	4
4.2. Generování obsahu	5
4.3. Podmínka pro výpis	5
4.4. Podmínka pro výpis	5
5.1. Výpis obrázku pomocí FO transformace	7
6.1. Ukázka CSS stylů	8

Úvod

Semestrální práce "Katalog hotelů" je nástroj vhodný pro cestovní kanceláře, nebo majitele hotelů. Umožňuje uchovávat strukturované informace o jednotlivých hotelech. Lze jí aplikovat na neomezené množství hotelů.

Šablona zároveň kontroluje vstupní data, zda uživatel zadává opravdu informace, které jsou požadovány.

Kapitola 1. Ukázkový dokument

Jednoduchý dokument ve formátu xml. Obsahuje náhodná data pro demonstraci funkcí nástroje.

Obsahuje elementy: hotely, budova, nazev, hvezdy, kontakt, telefon, email, adresa, ulice, cp, mesto, psc, predstavenstvo, osoba, jmeno, pocetVolnychPokoju, pokoje, pokoj, pocetLuzek, typ, socialniZarizeni, vybaveniPokoje, cenaZaNoc, vybaveniHotelu.

Kapitola 2. Schema

Většina hodnot formátu string je ošetřena regulárními výrazy.

Příklad 2.1. Ošetření telefonního čísla

```
<xs:simpleType name="telefonType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\+\d{3}\s?\d{3}\s?\d{3}\s?\d{3}" />
  </xs:restriction>
</xs:simpleType>
```

Naopak číselné hodnoty, jsou ve většině případů ošetřeny intervalem použitelných hodnot.

Příklad 2.2. Ošetření počtu volných pokojů

```
<xs:simpleType name="pocetVolnychPokojuType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0" />
  </xs:restriction>
</xs:simpleType>
```

Kapitola 3. Schematron

Schematron ov#uje, že po#et dostupných pokoj# odpovídá skute#nému stavu uvedenému u jednotlivých pokoj#.

P#íklad 3.1. Kontrola po#tu dostupných pokoj#

```
<sch:pattern>
  <sch:title>Kontrola po#tu dostupných pokoj#</sch:title>
  <sch:rule context="h:hotely/h:budova">
    <sch:report test="count(h:pokoje/h:pokoj[contains(@prov
  </sch:rule>
</sch:pattern>
```

Kapitola 4. Transformace HTML

Zajišťuje transformaci do několika HTML souborů. Soubory jsou navzájem provázány odkazy.

Následující úsek kódu zajišťuje generování jednotlivých HTML souborů, obsahující grafický výpis jednotlivých hotelů. Unikátnost jednotlivých názvů je zajištěná funkcí generate-id(.

Příklad 4.1. Generování jednotlivých souborů

```
<xsl:result-document href="{generate-id(.)}.html" format="html">
  <xsl:text disable-output-escaping="yes">&lt;!DOCTYPE html&g
  <html>
    <head>
      <title>
        <xsl:value-of select="h:nazev"/>
      </title>
      <style>
        ...
      </style>
    </head>
    <body>
      <header>
        <div class="header">
          <a href="hotelyHTML.html">Zpět na seznam ho
          <h1><xsl:value-of select="h:nazev"/></h1>
          <h2><xsl:value-of select="h:hvezdy"/><xsl:t
        </div>
      </header>
      <main>
        <div class="main">
          <div class="obrazek">
            <xsl:text disable-output-escaping="yes">
            <xsl:value-of select="h:nazev"/>
            <xsl:text disable-output-escaping="yes">
            <xsl:value-of select="h:nazev"/>
            <xsl:text disable-output-escaping="yes">
            že nebyl nahrán, nebo nebyl uložen ve f
            <xsl:value-of select="h:nazev"/>
            <xsl:text disable-output-escaping="yes">
          </div>
          <div class="pokoje">
            <h2><xsl:text>Pokoje:</xsl:text></h2>
            <xsl:apply-templates select="h:pokoje"/>
          </div>
        </div>
      </main>
      <footer>
        <div class="footer">
          <div class="kontakt">
            <h2><xsl:text>Kontakt:</xsl:text></h2>
            <xsl:apply-templates select="h:kontakt"/>
          </div>
          <div class="predstavenstvo">
            <h2><xsl:text>Predstavenstvo:</xsl:text>
            <xsl:apply-templates select="h:predstav
```

```
        </div>
    </div>
</footer>
</body>
</html>
</xsl:result-document>
```

Následující úsek kódu zajišťuje generování obsahu, kliknutím na jednotlivé položky se uživatel přesune na stránku, která obsahuje výpisy jednotlivých hotelů. Stejně jako v předchozím případě, je i zde využita funkce `generate-id(.)` pro opětovné vygenerování názvu již uloženého souboru.

Příklad 4.2. Generování obsahu

```
<li>
  <a href="{generate-id(.)}.html">
    <xsl:value-of select="h:nazev"/>
    <xsl:text disable-output-escaping="yes"> - </xsl:text>
    <xsl:value-of select="generate-id(.)"/>
  </a>
</li>
```

Následující úsek kódu zajišťuje textový výpis, který uživatele informuje o přítomnosti sociálního zařízení v pokoji. Funkce je realizovaná pomocí funkcionality `xsl:choose`. V tagu `xsl:when` je uvedena podmínka, následuje výpis, který se zobrazí v případě splnění podmínky. Naopak tag `xsl:otherwise` označuje výpis, který se zobrazí při nesplnění podmínky.

Příklad 4.3. Podmínka pro výpis

```
<xsl:choose>
  <xsl:when test="h:socialniZarizeni = 'Ano'">
    <xsl:text>Pokoj má vlastní sociální zařízení.</xsl:text>
    <br/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>Pokoj nemá vlastní sociální zařízení.</xsl:text>
    <br/>
  </xsl:otherwise>
</xsl:choose>
```

Následující ukázka zajišťuje vygenerování HTML kódu pro obrázek hotelu, do kódu bylo nutné vložit název konkrétního hotelu, aby se zobrazil odpovídající obrázek.

Příklad 4.4. Podmínka pro výpis

```
<xsl:text disable-output-escaping="yes">&lt;img src="
<xsl:value-of select="h:nazev"/>
<xsl:text disable-output-escaping="yes">.jpg" alt="Obrázek hote
```

```
<xsl:value-of select="h:nazev"/>
<xsl:text disable-output-escaping="yes">. (Pokud se obrázek nez
znamená to, že nebyl nahrán, nebo nebyl uložen ve formátu .jpg
<xsl:value-of select="h:nazev"/>
<xsl:text disable-output-escaping="yes">.jpg)" width="500"&gt;<
```

Kapitola 5. FO transformace

Kód uvedený v následujícím výpisu, zprostředkovává výpis obrázku konkrétního hotelu. Většina kódu je vypsána ručně pomocí funkcionality `xlt:text`, konkrétní název a díky `name` i správný obrázek je do kódu vložen pomocí funkce `xsl:value-of`.

Příklad 5.1. Výpis obrázku pomocí FO transformace

```
<xsl:text disable-output-escaping="yes">&lt;img src="</xsl:text>
<xsl:value-of select="h:nazev"/>
<xsl:text disable-output-escaping="yes">.jpg" alt="Obrázek hotelu
<xsl:value-of select="h:nazev"/>
<xsl:text disable-output-escaping="yes">. (Pokud se obrázek nez
znamená to, že nebyl nahrán, nebo nebyl uložen ve formátu .jpg
<xsl:value-of select="h:nazev"/>
<xsl:text disable-output-escaping="yes">.jpg)" width="500"&gt;</pre>
```

Kapitola 6. Schematron

CSS styly upravují vzhled ukázkového XML dokumentu.

Příklad 6.1. Ukázka CSS sytůl#

```
osoba email:before{
    display: inline;
    font-weight:bold;
    content: "Email: ";
    margin-left: 20px;
}

osoba email{
    display: block;
    margin-left: 20px;
}
```

Rejstřík

C

CSS, 8

F

FO transformace, 7

G

Generování obsahu, 5

H

HTML transformace, 4

O

Ošetření, 2

P

Podmínka, 5

S

Schematron, 3

Semestrální práce, v