

Склонение числительных

Вариант №1

Для склонения числительных обычно применяется подшаблон следующего вида, для использования в него необходимо вставить варианты требуемого для склонения слова:

```
<xsl:template name="declension1">
  <xsl:param name="number" />

  <xsl:variable name="nominative">
    <xsl:text>слово</xsl:text>
  </xsl:variable>

  <xsl:variable name="genitive_singular">
    <xsl:text>слова</xsl:text>
  </xsl:variable>

  <xsl:variable name="genitive_plural">
    <xsl:text>слов</xsl:text>
  </xsl:variable>

  <xsl:variable name="last_digit">
    <xsl:value-of select="$number mod 10"/>
  </xsl:variable>

  <xsl:variable name="last_two_digits">
    <xsl:value-of select="$number mod 100"/>
  </xsl:variable>

  <xsl:choose>
    <xsl:when test="$last_digit = 1 and $last_two_digits != 11">
      <xsl:value-of select="$nominative"/>
    </xsl:when>
    <xsl:when test="$last_digit = 2 and $last_two_digits != 12 or
$last_digit = 3 and $last_two_digits != 13 or $last_digit = 4 and
$last_two_digits != 14">
      <xsl:value-of select="$genitive_singular"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$genitive_plural"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

Как использовать:

```
<xsl:variable name="count" select="2" />
```

```

<!-- Выводим -->
<xsl:value-of select="$count" />
<xsl:text> </xsl:text>
<xsl:call-template name="declension1">
  <xsl:with-param name="number" select="$count"/>
</xsl:call-template>

```

Результат:

2 слова

Вариант №2

Следующий вариант подходит в случае, если склонение слова требуется использовать лишь однажды, он более нагляден и универсален, его можно использовать с любым словом:

```

<xsl:template name="declension2">
  <xsl:param name="number" />
  <xsl:param name="word" />

  <xsl:variable name="nominative">
    <xsl:value-of select="substring-before($word, '|')" />
  </xsl:variable>

  <xsl:variable name="genitive_singular">
    <xsl:value-of select="substring-before(substring-after($word, '|'),
'|')" />
  </xsl:variable>

  <xsl:variable name="genitive_plural">
    <xsl:value-of select="substring-after(substring-after($word, '|'),
'|')" />
  </xsl:variable>

  <xsl:variable name="last_digit">
    <xsl:value-of select="$number mod 10"/>
  </xsl:variable>

  <xsl:variable name="last_two_digits">
    <xsl:value-of select="$number mod 100"/>
  </xsl:variable>

  <xsl:choose>
    <xsl:when test="$last_digit = 1 and $last_two_digits != 11">
      <xsl:value-of select="$nominative"/>
    </xsl:when>
    <xsl:when test="$last_digit = 2 and $last_two_digits != 12 or
$last_digit = 3 and $last_two_digits != 13 or $last_digit = 4 and

```

```

$last_two_digits != 14">
    <xsl:value-of select="$genitive_singular"/>
</xsl:when>
<xsl:otherwise>
    <xsl:value-of select="$genitive_plural"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

```

Как использовать:

```

<xsl:variable name="count" select="2" />

<xsl:value-of select="$count" />
<xsl:text> </xsl:text>
<xsl:call-template name="declension2">
    <xsl:with-param name="number" select="$count"/>
    <xsl:with-param name="word" select="'слово|слова|слов'"/>
</xsl:call-template>

```

Результат:

2 слова

PS

Редко требуется использовать само слово в нужной форме без самого числа, задающего эту форму, то есть обычно выводится что-то вроде «2 яблока», «5 яблок», а сами слова «яблока» и «яблок» никакого смысла не несут и по отдельности от числа не используются, поэтому в шаблоны можно добавить код, который это число будет выводить, например:

```

<!-- ... -->
<xsl:value-of select="$number" />
<xsl:text> </xsl:text>

<xsl:choose>
    <xsl:when test="$last_digit = 1 and $last_two_digits != 11">

        <!-- ... -->

```

Страницу создал Максим Засорин 27.09.16 в 15:23