

Прекрасно, что мы можем выполнить функцию, но дело кончилось константой, запрятанной в глубине таблицы стилей. Не лучше ли было бы задавать переменные в начале страницы? Конечно, да (см. листинг 34).

Листинг 34. Задание переменной

```
<xsl:stylesheet version="1.0"
```

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

```
xmlns="http://www.w3.org/TR/xhtml1/strict"
```

```
xmlns:scaler = "com.backstop.RecipeScaler"
```

```
extension-element-prefixes="scaler">
```

```
<xsl:variable name="numberOfServings" select="3" />
```

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

```
<html>
```

```
<head>
```

```
<title>Recipes</title>
```

```
</head>
```

```
<body>
```

```
<scaler:scaleMessage servings="$numberOfServings" />
```

```
<xsl:apply-templates select="/recipes/recipe"/>
```

```
</body>
```

```
</html>
```

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

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

```
...
```

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

```
<xsl:template match="ingredients/ingredient">
```

```
<xsl:value-of select="scale" data-bbox="288 658 485 678"/> Number Of Servings ,
```

```
./qty"/>
```

```
<xsl:text> </xsl:text><xsl:value-of select="./unit"/>
```

```
<xsl:text> </xsl:text><xsl:value-of select="./food"/><br
```

/>

</xsl:template>

...

XSLT позволяет создавать переменные; в ссылке используется знак доллара (\$), который можно увидеть в листинге. Если применить таблицу стилей, можно увидеть два эффекта (см. рисунок 12).

Рисунок 12. Использование переменной

```

Command Prompt
C:\> xslt -s "C:\Documents and Settings\Nick\Desktop\style.xsl" -o "C:\Documents and Settings\Nick\Desktop\output.xml" -i "C:\Documents and Settings\Nick\Desktop\input.xml"
Factor of 2 (number of servings) is 2.
<h2>Gosh!</h2>
<h3>Ingredients</h3>
<p>
  2 pound hamburger</p>
  3 pound elbow macaroni</p>
  6 cups brown sugar</p>
  3 bag chopped onion</p>
  3 teaspoon dried basil</p>
</p>
<h3>Directions</h3>
<ol>
  <li>Brown the hamburger.</li>
  <li>Add onion and cook until transparent.</li>
  <li>Add brown sugar and basil.</li>
  <li>Cook and drain pasta.</li>
  <li>Combine meat and pasta.</li>
</ol>
<h2>Balanced Breakfast</h2>
<h3>Ingredients</h3>
<p>
  2 cup cereal</p>
  1 glass orange juice</p>
  2 cup milk</p>
  6 slices toast</p>
</p>
<h3>Directions</h3>
<ol>
  <li>Combine cereal and milk in bowl.</li>
  <li>Add all ingredients to table.</li>
</ol>
</body>
  
```

Обратите внимание, что ингредиенты были умножены на количество порций, как и ожидалось. Однако если вы посмотрите внимательнее, то увидите, что элемент extension сработал неверно, приняв переменную за строку, вместо того чтобы использовать значение самой переменной. Это не сбой; спецификация не требует, чтобы процессор что-либо делал со значениями атрибутов перед обработкой элемента extension. Поэтому нам надо как-то избежать этой проблемы.

Условная обработка

Первое, что мы можем сделать - это использовать условную обработку, чтобы сообщение отображалось, только если оно нужно. Например, см. листинг 35.

Листинг 35. Использование элемента `if`

```
<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns="http://www.w3.org/TR/xhtml1/strict"

xmlns:scaler = "com.backstop.RecipeScaler"

extension-element-prefixes="scaler">

<xsl:variable name="numberOfServings" select="1" />
```

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

```
<html>
```

```
<head>
```

```
<title>Recipes</title>
```

```
</head>
```

```
<body>
```

```
<xsl:if test="$numberOfServings > 1">
```

```
<scaler:scaleMessage servings="3" />
```

```
</xsl:if>
```

```
<xsl:apply-templates select="/recipes/recipe"/>
```

```
</body>
```

```
</html>
```

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

```
...
```

Содержимое элемента if, заданное атрибутом test, должно быть равно true (истина). Если это не так, что и произошло в данном случае, вывод не появится вовсе (см. рисунок 13).

Рисунок 13. Результаты предложения if

