1.XSL (eXtensible Stylesheet Language) —язык таблиц стилей для XML. XSL служит языком трансформирования документов XML и состоит из XML-словаря семантики форматирования. Имея класс произвольно структурированных XML-документов и файлов данных, дизайнеры используют таблицы стилей XSL для указания на то, как это структурированное содержимое должно быть пред-ставлено; как содержимое-источник должно быть стилизовано, расположено и разбито на странице в Web-браузере.

Расширение XSLT XSLT (eXtensible Stylesheet Language Transformations) — это декларативное описание преобразования (трансформации) любого XML-документа. Спецификация XSLT входит в состав XSL и яв-ляется рекомендацией W3C.

2.При помощи XSLT можно добавлять/удалять элементы и ат-рибуты в конечный файл. Также, можно реорганизовывать и сор-тировать элементы, выполнять тесты, определять, какие элементы скрыть или отобразить, и т. п.

3.Элемент верхнего уровня xsl:template определяет в преобразовании шаблонное правило, или просто шаблон.

этом правиле атрибут match говорит о том, что оно должно использоваться для обработки элементов bold — в данном случае они будут заменяться на элементы b.

5.<?xml-stylesheet type="text/xsl" href="catalog.xsl"?>

6.<xsl:stylesheet> определяет, что данный документ является таблицей стилей XSLT с атрибутами номера версии и простран-ства имен XSLT.

7.<xsl:apply-templates> применяет некий шаблон к текуще-му элементу или к дочернему узлу текущего элемента. Если в эле-мент <xsl:apply-templates> добавить атрибут select, то он будет относиться только к дочернему элементу, который соответствует значению этого атрибута и может использоваться для определения порядка, в котором будут обрабатываться дочерние узлы.

8.<xsl:sort> используется для сортировки выходных данных и располагается внутри элемента <xsl:for-each>

Атрибуты:

select обязательный атрибут, значением которого является выражение, называемое также ключевым выражением. Это выражение вычисляется для каждого узла обрабатываемого множества, преобразуется в строку и затем используется как значение ключа при сортировке. По умолчанию значением этого атрибута является ".", что означает, что в качестве значения ключа для каждого узла используется его строковое значение.orderнеобязательный атрибут, определяет порядок, в котором узлы должны сортироваться по своим ключам. Этот атрибут может принимать только два значения — "ascending", указывающее на восходящий порядок сортировки, и "descending", указывающее на нисходящий порядок. Значением по умолчанию является "ascending", то есть восходящий порядок.langнеобязательный атрибут, определяет язык ключей сортировки. Дело в том, что в разных языках символы алфавита могут иметь различный порядок, что, соответственно, должно учитываться при сортировке. Атрибут lang в XSLT может иметь те же самые значения, что и атрибут xml:lang (например: "en", "en-us", "ru" и т. д.). Если значение этого атрибута не определено, процессор может либо определять язык исходя из параметров системы, либо сортировать строки исходя из порядка кодов символов Unicode.

data-typeнеобязательный атрибут, определяет тип данных, который несут строковые значения ключей. Техническая рекомендация XSLT разрешает этому атрибуту иметь следующие значения:

"text" — ключи должны быть отсортированы в лексикографическом порядке исходя из языка, определенного атрибутом lang или параметрами системы. Это значение используется по умолчанию;

"number" — ключи должны сравниваться в численном виде. Если строковое значение ключа не является числом, оно будет преобразовано к не-числу (NaN), и, поскольку нечисловые значения неупорядочены, соответствующий узел может появиться в отсортированном множестве где угодно;

"имя" — в целях расширяемости XSLT также позволяет указывать в качестве типа данных произвольное имя. В этом случае реализация сортировки полностью зависит от процессора.

case-orderнеобязательный атрибут, указывает на порядок сортировки символов разных регистров. Значениями этого атрибута могут быть "upper-first", что означает, что заглавные символы должны идти первыми, или "lower-first", что означает, что первыми должны быть строчные символы. К примеру, строки "ночь", "Улица", "фонарь", "Аптека", "НОЧЬ", "Фонарь" при использовании case-order="upper-first" будут иметь порядок "Аптека", "НОЧЬ", "ночь", "Фонарь", "фонарь", "улица". При использовании case-order="lower-first" те же строки будут идти в порядке "Аптека", "ночь", "НОЧЬ", "фонарь", "Фонарь", "улица". Значение case-order по умолчанию зависит от процессора и языка сортировки. В большинстве случаев заглавные буквы идут первыми.

9.<xsl:choose> используется вместе с элементами <xsl:when> и <xsl:otherwise>, чтобы определить проверку на вы-полнение условия.

12.Состоит из трех частей: XSL Transformations (XSLT) — язык преобразования XML-документов. XSL Formatting Objects (XSL-FO) — язык разметки типографских макетов и иных предпечатных материалов. XPath — язык путей и выражений, используемый в том числе и в XSLT для доступа к отдельным частям XML-документа.

13. <?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/T  
ransform">  
<xsl:template match="/">  
<html>  
<head><title>My first template rule</title>  
</head>  
<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>

14.Процесс xslt-преобразования выглядит следующим образом: XSLT-процессор анализирует исходный XML-документ, последовательно перебирая каждый его элемент. При этом для каждого элемента процессор пытается найти подходящий шаблон (template) в XSL-файле путем сопоставлении исходного элемента и содержащегося в шаблоне образца (pattern). Если подходящий шаблон не найден, используется шаблон по умолчанию и процессор переходит к следующему элементу в исходном XML-документе. Если подходящий шаблон (template) найден процессор выполняет содержащиеся в шаблоне инструкции (Template Rules – правила шаблона), на основании чего строится конечное дерево. При этом текущим узлом для шаблона будет являться обрабатываемый узел XML-документа. В шаблоне могут содержаться правила (инструкции), выбирающие новый текущий узел.