ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Н.А. Соловьева |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8  **Формирование web-страницы на основе XML документа** |
| по дисциплине: Web-технологии |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Шумков Д.В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2024

**Цель работы:** изучение языка разметки XML и выполнение XSLT преобразования

**Задание:**

Подготовить файл xml, содержащий данные для таблицы. Наполнение таблицы определяется вариантом, выбранным в лабораторной работе № 1. Таблица должна содержать не менее 4 столбцов и 10 строк, один из столбцов должен отображать графические файлы. Данные таблицы не должны дублировать информацию, ранее использованную на разрабатываемом сайте.

Реализовать отображение на веб-сайте содержимого файла XML. Отобразить данные из файла XML двумя способами: в таблице (пример 1) и построчно (пример 2).

Использовать инструкции xsl:if, xsl:apply-templates, xsl:sort, в XML файле применить атрибуты тегов.

Подготовить файл с XML-схемой (файл xsd) и применить его для валидации созданных XML документов используя online-сервис. Показать, что документ прошел проверку.

Подготовить xsd файл для проверки созданного xml файла и выполнить проверку xml файла используя online-сервис.

Подготовить xsl файл для отображения созданного xml файла и отобразить его на сайте

Тема сайта: роботы игрушки

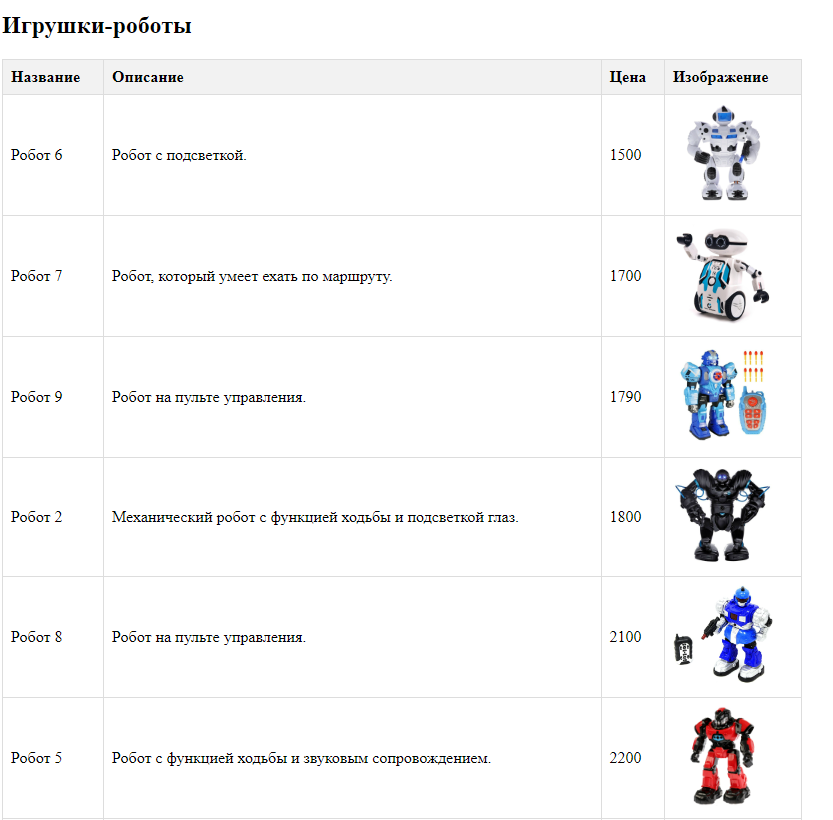


Рисунок 1 – Содержание страницы

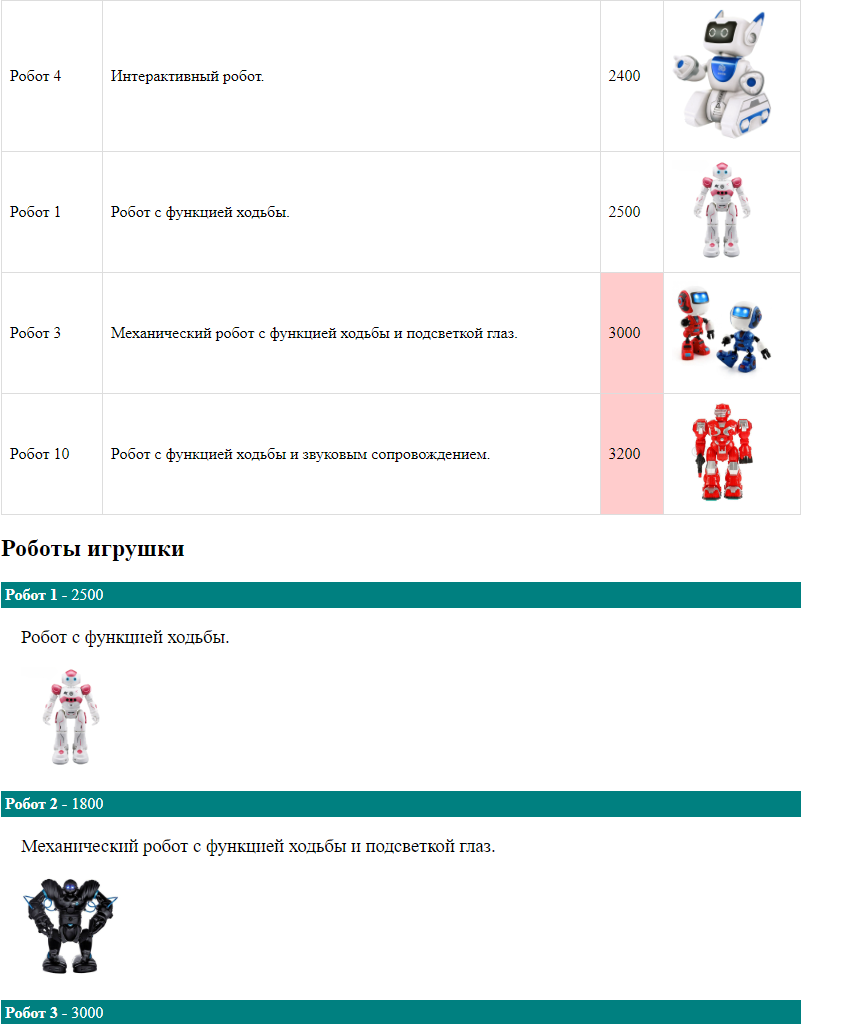


Рисунок 2 – Содержание страницы

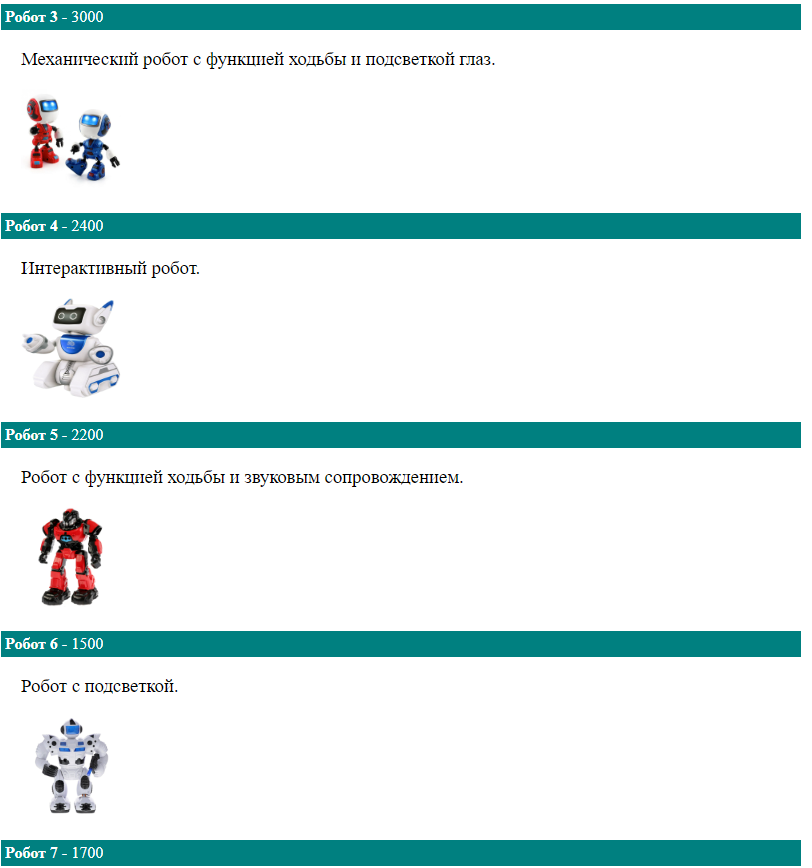


Рисунок 3 – Содержание страницы

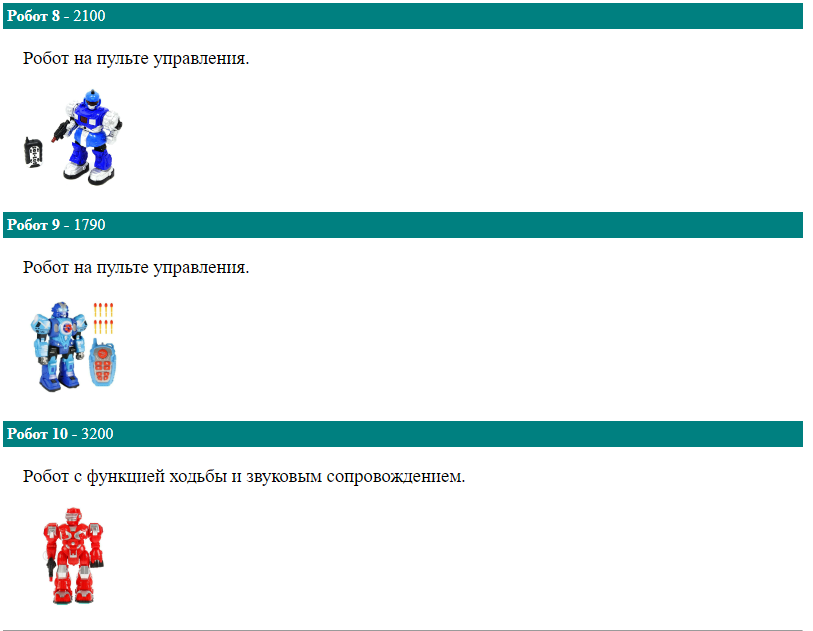


Рисунок 4 – Содержание страницы

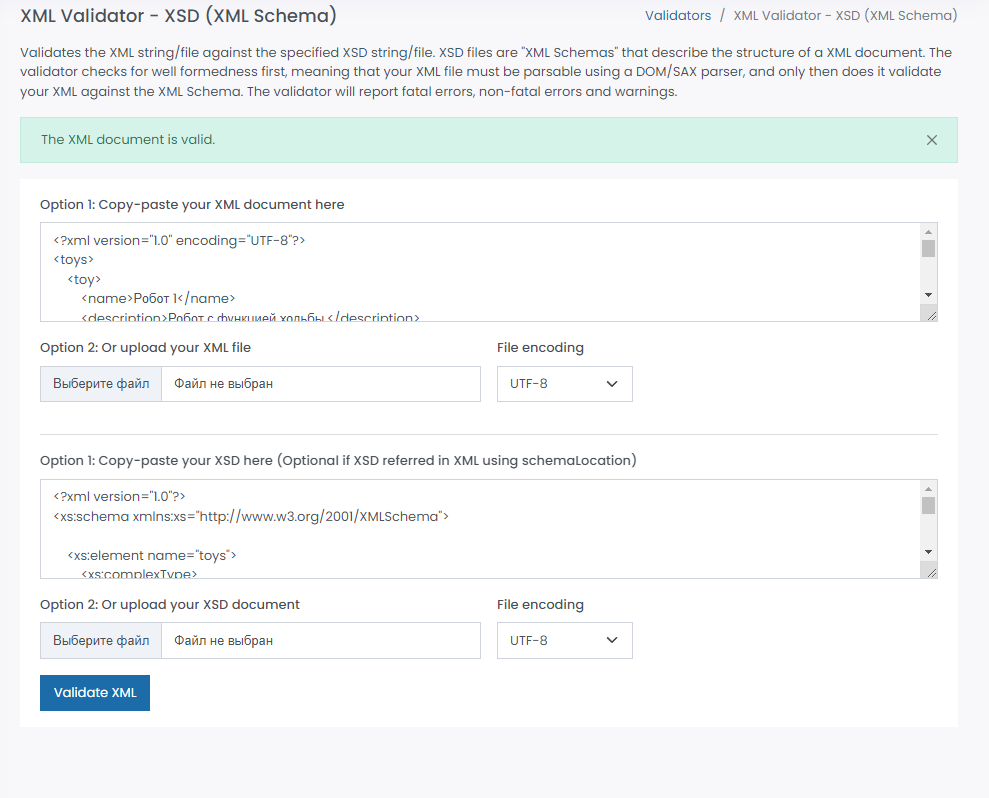


Рисунок 5 – Результат проверки XML

**Листинг кода**

**Toys.xml**

<?xml version="1.0" encoding="UTF-8"?>

<toys>

<toy>

<name>Робот 1</name>

<description>Робот с функцией ходьбы.</description>

<price>2500</price>

<image>robot1.jpg</image>

</toy>

<toy>

<name>Робот 2</name>

<description>Механический робот с функцией ходьбы и подсветкой глаз.</description>

<price>1800</price>

<image>robot2.jpg</image>

</toy>

<toy>

<name>Робот 3</name>

<description>Механический робот с функцией ходьбы и подсветкой глаз.</description>

<price>3000</price>

<image>robot3.jpg</image>

</toy>

<toy>

<name>Робот 4</name>

<description>Интерактивный робот.</description>

<price>2400</price>

<image>robot4.jpg</image>

</toy>

<toy>

<name>Робот 5</name>

<description>Робот с функцией ходьбы и звуковым сопровождением.</description>

<price>2200</price>

<image>robot5.jpg</image>

</toy>

<toy>

<name>Робот 6</name>

<description>Робот с подсветкой.</description>

<price>1500</price>

<image>robot6.jpg</image>

</toy>

<toy>

<name>Робот 7</name>

<description>Робот, который умеет ехать по маршруту.</description>

<price>1700</price>

<image>robot7.jpg</image>

</toy>

<toy>

<name>Робот 8</name>

<description>Робот на пульте управления.</description>

<price>2100</price>

<image>robot8.jpg</image>

</toy>

<toy>

<name>Робот 9</name>

<description>Робот на пульте управления.</description>

<price>1790</price>

<image>robot9.jpg</image>

</toy>

<toy>

<name>Робот 10</name>

<description>Робот с функцией ходьбы и звуковым сопровождением.</description>

<price>3200</price>

<image>robot10.jpg</image>

</toy>

</toys>

**Toys.xsl**

<?xml version="1.0" encoding="UTF-8"?>

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

<xsl:template match="/">

<h2>Игрушки-роботы</h2>

<table border="1">

<tr bgcolor="#9acd32">

<th style="text-align:left">Название</th>

<th style="text-align:left">Описание</th>

<th style="text-align:left">Цена</th>

<th style="text-align:left">Изображение</th>

</tr>

<xsl:apply-templates select="toys/toy">

<xsl:sort select="price"/>

</xsl:apply-templates>

</table>

</xsl:template>

<xsl:template match="toy">

<xsl:variable name="price" select="price"/>

<tr>

<td>

<xsl:value-of select="name"/>

</td>

<td>

<xsl:value-of select="description"/>

</td>

<td>

<xsl:if test="$price &gt; 2500">

<xsl:attribute name="bgcolor">#FFCCCC</xsl:attribute>

</xsl:if>

<xsl:value-of select="price"/>

</td>

<td>

<img src="{image}" alt="{name}" style="width: 100px; height: auto;"/>

</td>

</tr>

</xsl:template>

</xsl:stylesheet>

**Toysline.xsl**

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

<xsl:template match="/">

<h2>Роботы игрушки</h2>

<xsl:for-each select="toys/toy">

<div style="background-color: teal; color: white; padding: 4px">

<span style="font-weight: bold">

<xsl:value-of select="name"/>

-

</span>

<xsl:value-of select="price"/>

</div>

<div style="margin-left: 20px; margin-bottom: 1em; font-size: 14pt">

<p>

<xsl:value-of select="description"/>

</p>

<img src="{image}" alt="{name}" style="width: 100px; height: 100px;"/>

</div>

</xsl:for-each>

</xsl:template>

</xsl:stylesheet>

**Xsdfile.xsd**

<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="toys">

<xs:complexType>

<xs:sequence>

<xs:element name="toy" maxOccurs="unbounded">

<xs:complexType>

<xs:sequence>

<xs:element name="name" type="xs:string"/>

<xs:element name="description" type="xs:string"/>

<xs:element name="price" type="xs:decimal"/>

<xs:element name="image" type="xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="styles.css">

<script>

function loadXMLDoc(filename) {

if (window.ActiveXObject) {

xhttp = new ActiveXObject("Msxml2.XMLHTTP");

} else {

xhttp = new XMLHttpRequest();

}

xhttp.open("GET", filename, false);

try {

xhttp.responseType = "msxml-document"

} catch (err) {

} // Helping IE11

xhttp.send("");

return xhttp.responseXML;

}

function displayResult() {

xml = loadXMLDoc("toys.xml");

xsl = loadXMLDoc("toys.xsl");

xlsLine = loadXMLDoc("toysline.xsl");

// код для IE

if (window.ActiveXObject || xhttp.responseType == "msxml-document") {

ex = xml.transformNode(xsl);

document.getElementById("example").innerHTML = ex;

}

// код для Chrome, Firefox, Opera и др.

else if (document.implementation && document.implementation.createDocument) {

xsltProcessor = new XSLTProcessor();

xsltProcessor.importStylesheet(xsl);

resultDocument = xsltProcessor.transformToFragment(xml, document);

document.getElementById("example").appendChild(resultDocument);

xsltProcessor.importStylesheet(xlsLine);

resultDocument = xsltProcessor.transformToFragment(xml, document);

document.getElementById("lineExample").appendChild(resultDocument);

}

}

</script>

<style>

tbody img {

width: 50px;

height: auto;

filter: grayscale(0%);

}

</style>

</head>

<body onload="displayResult()">

<div class="cont" style="width:800px">

<main>

<div id="lineExample"/>

<div id="example"/>

</main>

<hr>

</div>

</body>

</html>