

Лабораторная работа №8 Работа с XML через JavaScript

Цель: научиться использовать работать с XML и использовать XMLHttpRequest() .

Следующий пример позволяет прочитать два уровня иерархии на примере созданного в рамках предыдущей лабораторной работы XML2.xml:

```
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
th, td {
    padding: 5px;
}
</style>
</head>
<body>

<script>
    if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    }
    else { // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.open("GET", "XML2.xml", false);
    xmlhttp.send();
    xmlDoc = xmlhttp.responseXML;

    //начинаем движение по элементам первого уровня иерархии - Авторы
    document.write("<table><tr><th>Имя автора</th><th>Список книг</th></tr>");
    var Author = xmlDoc.getElementsByTagName("Author");
    for (i = 0; i < Author.length; i++) {
        document.write("<tr><td>");

document.write(Author[i].getElementsByTagName("Name_author")[0].childNodes[0].nodeValue);
        document.write("</td><td>");

        //начинаем движение по подчиненным элементам - Книги текущего автора
        var Book = Author[i].getElementsByTagName("Book");
        document.write("<table><tr><th>Название                книги</th><th>Количество
страниц</th></tr>");
        for (j = 0; j < Book.length; j++) {
            document.write("<tr><td>");

document.write(Book[j].getElementsByTagName("Name")[0].childNodes[0].nodeValue);
            document.write("</td><td>");

document.write(Book[j].getElementsByTagName("Pages")[0].childNodes[0].nodeValue);
            document.write("</td></tr>");
        }
    }
</script>
</body>
</html>
```

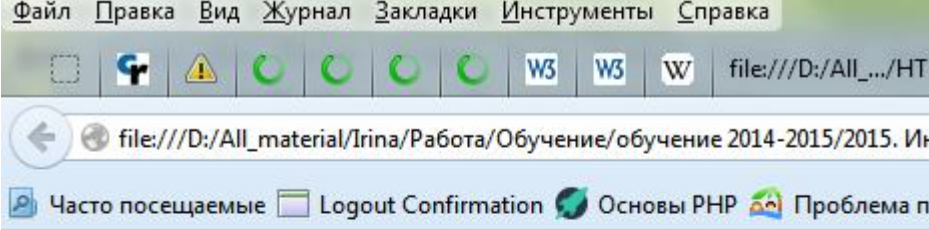
```

document.write("</table>"); //закрыли таблицу книг текущего автора
document.write("</td></tr>"); //закрыли столбец и строку текущего автора
}
document.write("</table>"); //закрыли таблицу авторов
</script>

</body>
</html>

```

Результат:



The screenshot shows a web browser window with a table rendered from an HTML document. The table has two main sections, one for 'Пушкин А.С.' and one for 'Толстой Л.Н.'. Each section contains a list of books with their titles and page counts.

Имя автора	Список книг	
Пушкин А.С.	Название книги	Количество страниц
	Руслан и Людмила	150
	Сказка о рыбаке и рыбке	25
Толстой Л.Н.	Название книги	Количество страниц
	Война и мир	1000

Индивидуальное задание 1.

Вариант 1. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности по примеру выше.

Вариант 2. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности по примеру выше.

Вариант 3. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности по примеру выше.

Вариант 4. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности по примеру выше.

Индивидуальное задание 2.

Вариант 1. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности в виде многоуровневого списка.

Вариант 2. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности в виде многоуровневого списка.

Вариант 3. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности в виде многоуровневого списка.

Вариант 4. Используя XML из второго индивидуального задания, создать HTML документ, который будет выводить данные всех трех уровней вложенности в виде многоуровневого списка.

Пример оформления списка для всех вариантов:

- данные 1 уровня иерархии
 - + данные 2 уровня иерархии
 - ...
 - * данные 3 уровня иерархии
 - ...
- данные 1 уровня иерархии
 - ...

Индивидуальное задание 3.

Используя ресурс <http://www.cbr.ru/scripts/Root.asp?Prtid=SXML>

Создать html-файл, который парсит и выводит данные XML в виде таблицы по следующим данным:

Вариант 1:

Получение котировок на день сдачи лабораторной работы.

Вариант 2:

Коды валют устанавливаемые ежедневно.

Вариант 3:

Пример получения динамики котировок доллара США.

Вариант 4:

Пример получения динамики котировок драгоценных металлов.

Кусок кода как пример:

```
function get_currency_list()
{
    if(Request.Get("http://www.cbr.ru/scripts/XML_daily.asp"))
    {
        // work with list
        CurrencyList.RegisterColumn(currencyCode,"Code",false);
        CurrencyList.RegisterColumn(currencyName,"Currency",false);
        CurrencyList.RegisterColumn(nominal,"Nominal",false);
        CurrencyList.RegisterColumn(value,"Value",true);

        var str = Request.HTMLText;

        var re =
        /<CharCode>(.*?)</CharCode>\s*<Nominal>(.*?)</Nominal>\s*<Name>(.*?)</Name>\s*<Value>(.*?)</Value>/im;
```

```
var arr;

while ((arr = re.exec(str)) != null)
{
    str = str.substring(arr.lastIndex, str.length);

    if(arr.length > 4)
    {
        var currency = CurrencyList.AddCurrency(arr[1]);
        if(currency)
        {
            currency.CurrencyName = arr[3];
            currency.SetValue(currencyCode, arr[1]);
            currency.SetValue(currencyName, arr[3]);
            currency.SetValue(nominal, arr[2]);
            currency.SetValue(value, arr[4]);
        }
    } // if(arr.length > 4)

} // while

return CurrencyList.CurrenciesCount > 0;

} // if(Request.Get(pageURL))

return false;
}
```

Навигация по XML

Вывод первого элемента из XML

Следующий пример позволяет получить XML данные о первом CD элементе и отображать в HTML элементе с id="showCD". Функция displayCD() вызывается, когда страница загружается:

```
<!DOCTYPE html>
<html>
<head>

<script>
  if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp = new XMLHttpRequest();
  }
  else { // code for IE6, IE5
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
  }
  xmlhttp.open("GET", "cd_catalog.xml", false);
  xmlhttp.send();
  xmlDoc = xmlhttp.responseXML;

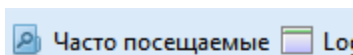
  x = xmlDoc.getElementsByTagName("CD");
  i = 0;

  function displayCD() {
    artist = (x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
    title = (x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue);
    year = (x[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue);
    txt = "Artist: " + artist + "<br>Title: " + title + "<br>Year: " + year;
    document.getElementById("showCD").innerHTML = txt;
  }
</script>
</head>
<body onload="displayCD()">

<div id='showCD'></div>

</body>
</html>
```

Результат:



Artist: Bob Dylan
Title: Empire Burlesque
Year: 1985

Перемещение между элементами XML

Пример перемещения между записями "cd_catalog.xml"

```
<!DOCTYPE html>
<html>
<head>

<script>
  if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp = new XMLHttpRequest();
  }
  else { // code for IE6, IE5
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
  }
  xmlhttp.open("GET", "cd_catalog.xml", false);
  xmlhttp.send();
  xmlDoc = xmlhttp.responseXML;

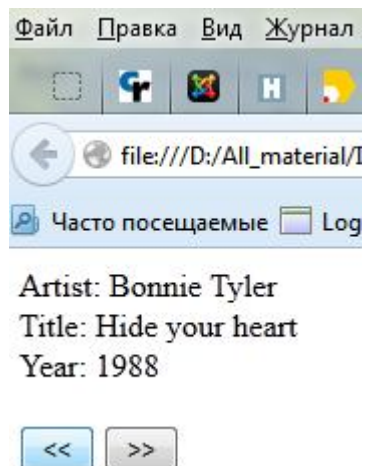
  x = xmlDoc.getElementsByTagName("CD");
  i = 0;

  // вывод данных текущего элемента
  function displayCD() {
    artist = (x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
    title = (x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue);
    year = (x[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue);
    txt = "Artist: " + artist + "<br>Title: " + title + "<br>Year: " + year;
    document.getElementById("showCD").innerHTML = txt;
  }
  // переход на следующий элемент
  function next() {
    if (i < x.length - 1) {
      i++;
      displayCD();
    }
  }
  // переход на предыдущий элемент
  function previous() {
    if (i > 0) {
      i--;
      displayCD();
    }
  }
</script>
</head>
<body onload="displayCD()">

<div id='showCD'></div><br>
<input type="button" onclick="previous()" value="<<" />
<input type="button" onclick="next()" value=">>" />

</body>
</html>
```

Результат:



Просмотр подробной информации при нажатии на конкретную строку

Следующий пример показывает, как вы можете получить подробную информацию об музыкальном альбоме, когда кликаете по названию CD.

Пример HTML файла с навигацией по ["cd_catalog.xml"](#).

```
<!DOCTYPE html>
<html>
<head>
<script>
    if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera,
Safari
        xmlhttp = new XMLHttpRequest();
    }
    else { // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.open("GET", "cd_catalog.xml", false);
    xmlhttp.send();
    xmlDoc = xmlhttp.responseXML;
    x = xmlDoc.getElementsByTagName("CD");

    function displayCDInfo(i) {
        artist =
            (x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
        title =
            (x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue);
        year =
            (x[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue);
        country =
            (x[i].getElementsByTagName("COUNTRY")[0].childNodes[0].nodeValue);
        company =
            (x[i].getElementsByTagName("COMPANY")[0].childNodes[0].nodeValue);
```

```

        price =
            (x[i].getElementsByTagName("PRICE")[0].childNodes[0].nodeValue);
        txt = "Artist: " + artist + "<br>Title: " + title + "<br>Year: " +
            year + "<br>Country: " + country + "<br>Company: " + company +
            "<br>Price: " + price;
        document.getElementById("showCD").innerHTML = txt;
    }
</script>
</head>

<body>
<h2>Click on a CD to display album information.</h2>

<div id='showCD'>Click on a CD to display album information.</div><br>
<script>
    document.write("<table border='1'>");
    for (var i = 0; i < x.length; i++) {
        document.write("<tr onclick='displayCDInfo(" + i + ")'>");
        document.write("<td>");

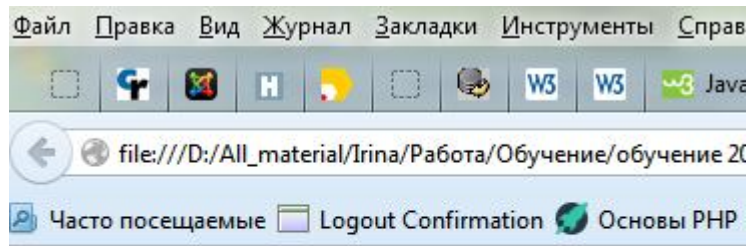
document.write(x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue
);
        document.write("</td><td>");

document.write(x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue)
;
        document.write("</td></tr>");
    }
    document.write("</table>");
</script>

</body>
</html>

```

Результат:



Click on a CD to display album info

Artist: Bonnie Tyler
Title: Hide your heart
Year: 1988
Country: UK
Company: CBS Records
Price: 9.90

Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Joe Cocker	Unchain my heart

Обзор возможностей работы с XML DOM

В данном разделе приводятся сведения о работе некоторых методов в XML DOM.

Получение значения элемента

Для работы примеров используется файл books.xml.

Пример books.xml ниже:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
<title lang="en">XQuery Kick Start</title>
<author>James McGovern</author>
<author>Per Bothner</author>
```

```

<author>Kurt Cagle</author>
<author>James Linn</author>
<author>Vaidyanathan Nagarajan</author>
<year>2003</year>
<price>49.99</price>
</book>
<book category="WEB">
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>

</bookstore>

```

Следующий пример возвращает значение первого <title> элемента благодаря работе оператора:

```
txt=xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
```

```

<!DOCTYPE html>
<html>
<body>
  <script>
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

txt=xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
document.write(txt);
  </script>
</body>
</html>

```

Получение значения атрибута

Следующий пример возвращает значение атрибута "lang" первого <title> элемента с помощью оператора:

```
txt=xmlDoc.getElementsByTagName("title")[0].getAttribute("lang");
```

```

<!DOCTYPE html>
<html>
<body>
  <script>
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else

```

```

    {// code for IE6, IE5
      xmlhttp=new XMLHttpRequest("Microsoft.XMLHTTP");
    }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

txt=xmlDoc.getElementsByTagName("title")[0].getAttribute("lang");
document.write(txt);
    </script>
</body>
</html>

```

Изменение значения элемента

Следующий пример изменяет значение первого <title> элемента с помощью операторов:

```

x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
x.nodeValue="Easy Cooking";

```

```

<!DOCTYPE html>
<html>
<body>
    <script>
if (window.XMLHttpRequest)
    {// code for IE7+, Firefox, Chrome, Opera, Safari
      xmlhttp=new XMLHttpRequest();
    }
else
    {// code for IE6, IE5
      xmlhttp=new XMLHttpRequest("Microsoft.XMLHTTP");
    }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
x.nodeValue="Easy Cooking";

x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
txt=x.nodeValue;
document.write(txt);
    </script>
</body>
</html>

```

Создание нового атрибута

XML DOM setAttribute() метод может использоваться для изменения значения существующего атрибута или создания нового атрибута:

Следующий пример добавляет новый атрибут (edition="first") для каждого <book> элемента:

```
x=xmlDoc.getElementsByTagName("book");
```

```
for(i=0;i<x.length;i++)  
{  
  x[i].setAttribute("edition","first");  
}
```

```
<!DOCTYPE html>  
<html>  
<body>  
<script>  
if (window.XMLHttpRequest)  
  {  
    // code for IE7+, Firefox, Chrome, Opera, Safari  
    xmlhttp=new XMLHttpRequest();  
  }  
else  
  {  
    // code for IE6, IE5  
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  
  }  
xmlhttp.open("GET","books.xml",false);  
xmlhttp.send();  
xmlDoc=xmlhttp.responseXML;  
  
x=xmlDoc.getElementsByTagName("book");  
for(i=0;i<x.length;i++)  
{  
  x[i].setAttribute("edition","first");  
}  
  
//Output all attribute values  
for (i=0;i<x.length;i++)  
{  
  document.write("Category: " + x[i].getAttribute('category') + "<br>");  
  document.write("Edition: " + x[i].getAttribute('edition') + "<br>");  
}  
</script>  
</body>  
</html>
```

Создание нового элемента

XML DOM createElement() метод создает новый узел элемента.

XML DOM createTextNode() метод создает новый текстовый узел.

XML DOM appendChild() метод добавляет дочерний узел к узлу (после последнего дочернего).

Чтобы создать новый элемент с текстовым контентом, необходимо сначала создать новый узел элемента и новый текстовый узел, затем добавить его к существующему узлу.

Следующий пример создает новый элемент <edition> с текстом: «First» и добавляет его к первому <book> элементу:

```
newel=xmlDoc.createElement("edition");  
newtext=xmlDoc.createTextNode("First");  
newel.appendChild(newtext);
```

```
x=xmlDoc.getElementsByTagName("book");
x[0].appendChild(newel);
```

```
<!DOCTYPE html>
<html>
<body>
<script>
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

newel=xmlDoc.createElement("edition");
newtext=xmlDoc.createTextNode("First");
newel.appendChild(newtext);
x=xmlDoc.getElementsByTagName("book");
x[0].appendChild(newel);

for (i=0;i<x[0].childNodes.length;i++)
{
if (x[0].childNodes[i].nodeType==1)
  {
document.write(x[0].childNodes[i].nodeName);
document.write(": ");
document.write(x[0].childNodes[i].childNodes[0].nodeValue);
document.write("<br>");
  }
}
</script>
</body>
</html>
```

Удаление элемента

Следующий пример удаляет первый узел в первом <book> элементе:

```
x=xmlDoc.getElementsByTagName("book")[0];
x.removeChild(x.childNodes[0]);
```

```
<!DOCTYPE html>
<html>
<body>
  <script>
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
```

```
xmlhttp=new XMLHttpRequest("Microsoft.XMLHTTP");
}
xmlhttp.open("GET", "books.xml", false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

var x=xmlDoc.getElementsByTagName("book")[0];
document.write("Child nodes before removal: ");
document.write(x.childNodes.length);
x.removeChild(x.childNodes[0]);

document.write("<br>Child nodes after removal: ");
document.write(x.childNodes.length);

</script>
</body>
</html>
```

Примечание: Результат выполнения примера может быть различным в зависимости от того, какой браузер вы используете. Firefox обрабатывает новые линии как пустые текстовые узлы, Internet Explorer - нет.

Индивидуальное задание 4:

По вашему варианту XML из предыдущей работы сделать html-файл, который содержит кнопки с функциями:

- Вывод данных элемента по заданному номеру узла и названию (вводится в поле).
- Вывод данных атрибута по заданному номеру узла и названию (вводится в поле).
- Добавляет элемент к заданному по номеру (вводится в поле).
- Удаляет элемент по заданному номеру (вводится в поле).