

Школа DOM

Данная серия документов подготовлена на основе материалов сайта Школы Консорциума W3C. Этот сайт является экспериментальным сервером, на котором содержание документов хранится в формате XML. Пользователям сайта эти документы доступны в виде HTML (преобразование на стороне клиента с помощью таблицы стилей XSLT) и в виде PDF (преобразование тех же документов в XSL-FO, а затем в формат PDF).

Добро пожаловать в школу DOM

Школа XML DOM¹ ([open link](#))

В школе XML DOM вы узнаете что такое DOM (Document Object Model - Объектная модель документа). XML DOM - это программный интерфейс XML-документов. Она определяет пути, которыми можно обращаться к XML-документу и производить с ним манипуляции. Изучайте DOM!

Описание объектов XML DOM² ([open link](#))

В нашей школе вы найдете полное описание объектов DOM, в нем описываются все объекты, их свойства и методы.

Примеры XML DOM³ ([open link](#))

Учитесь на примерах - их более 50! С помощью нашего редактора вы можете вносить изменения в код и нажимая на кнопку, увидеть результат ваших изменений. Попробуйте!

Ресурсы по DOM⁴ ([open link](#))

Список важных ресурсов по DOM в Интернете.

Школа XML DOM: содержание

Введение в XML DOM⁵ ([open link](#))

Введение в объектную модель документа XML DOM (Document Object Model).

Парсирование DOM⁶ ([open link](#))

Как произвести анализ объектной модели документа XML используя парсер Microsoft XML.

Ошибки парсирования⁷ ([open link](#))

Объект `parseError` и его свойства. Как проверить правильность XML и отобразить ошибки.

Валидатор XML⁸ ([open link](#))

Объект `parseError` применяется для создания валидатора XML-файла.

Получение доступа к DOM⁹ ([open link](#))

Как получать доступ к элементам XML-документа.

- 1: http://xml.nsu.ru/dom/dom_intro.xml
- 2: http://xml.nsu.ru/dom/dom_nodetype.xml
- 3: http://xml.nsu.ru/dom/dom_examples.xml
- 4: http://xml.nsu.ru/dom/dom_resources.xml
- 5: http://xml.nsu.ru/dom/dom_intro.xml
- 6: http://xml.nsu.ru/dom/dom_parser.xml
- 7: http://xml.nsu.ru/dom/dom_errors.xml
- 8: http://xml.nsu.ru/dom/dom_validate.xml
- 9: http://xml.nsu.ru/dom/dom_access.xml

Объект HttpRequest¹⁰ (open link)

Объект httpRequest, его свойства и методы.

Объекты XML DOM**Типы узлов**¹¹ (open link)

Описание типов узлов, с примерами.

Объект узел (Node)¹² (open link)

Свойства и методы объекта node, с примерами.

Объект список узлов (NodeList)¹³ (open link)

Свойства и методы объекта nodeList, с примерами.

Объект документ (Document)¹⁴ (open link)

Свойства и методы объекта document, с примерами.

Объект элемент (Element)¹⁵ (open link)

Свойства и методы объекта element, с примерами.

Объект атрибут (Attr)¹⁶ (open link)

Свойства объекта attr, с примерами.

Объект текст (Text)¹⁷ (open link)

Методы объекта text, с примерами.

Объект CDATASection¹⁸ (open link)

Описание объекта CDATASection, с примерами.

Объект комментарий (Comment)¹⁹ (open link)

Описание объекта comment, с примерами.

Примеры XML DOM**Примеры DOM**²⁰ (open link)

Множество примеров DOM!

Ресурсы по DOM**Ресурсы по DOM**²¹ (open link)

Список важных ресурсов по DOM, имеющихся в Интернете.

- 10: http://xml.nsu.ru/dom/dom_http.xml
- 11: http://xml.nsu.ru/dom/dom_nodetype.xml
- 12: http://xml.nsu.ru/dom/dom_node.xml
- 13: http://xml.nsu.ru/dom/dom_nodelist.xml
- 14: http://xml.nsu.ru/dom/dom_document.xml
- 15: http://xml.nsu.ru/dom/dom_element.xml
- 16: http://xml.nsu.ru/dom/dom_attribute.xml
- 17: http://xml.nsu.ru/dom/dom_text.xml
- 18: http://xml.nsu.ru/dom/dom_cdatasection.xml
- 19: http://xml.nsu.ru/dom/dom_comment.xml
- 20: http://xml.nsu.ru/dom/dom_examples.xml
- 21: http://xml.nsu.ru/dom/dom_resources.xml

Введение в XML DOM

XML DOM - это программный интерфейс XML-документов. Она определяет пути, которыми можно обращаться к XML-документу и производить с ним манипуляции.

Введение

В качестве спецификации W3C, целью разработки XML DOM было создание стандартного программного интерфейса для широкого диапазона приложений. XML DOM создан так, что может использоваться любым языком программирования и любой операционной системой.

С помощью XML DOM программист может создавать XML-документы, реализовывать навигацию по их структуре, добавлять, модифицировать или удалять определенные элементы.

Интерфейс на основе узлов

Как вы увидите в следующем разделе, программа, которая называется XML-парсер, может загрузить XML-документ в память вашего компьютера. После того, как он загружен в память, информацию, содержащуюся в нем можно извлекать или производить с ней манипуляции. Для этого нужно получить доступ к объектной модели этого документа DOM.

DOM представляет собой дерево, отображающее структуру XML-документа. Элемент `documentElement` является верхним уровнем этого дерева. Этот элемент имеет один или несколько дочерних элементов `childNodes`, представляющих ветви дерева.

Модель интерфейса на основе узлов используется для получения доступа к отдельным элементам дерева узлов.

Парсер Microsoft XML поддерживает все необходимые функции управления деревом узлов, получение доступа к отдельным узлам дерева и их атрибутам, позволяет вставлять и удалять узлы, преобразовывать дерево узлов обратно в XML.

Большая часть функций этого парсера, демонстрируемых на этой школе, соответствуют официальной рекомендации консорциума W3C по XML DOM, кроме функций `document load` и `error` (поверите ли, официальная рекомендация по DOM не предусматривает стандартной функции для загрузки XML-документов!).

В этой таблице перечисляются наиболее используемые типы узлов, которые поддерживает парсер Microsoft XML:

Тип узла	Пример
Тип документа	<code><!DOCTYPE food SYSTEM "food.dtd"></code>
Процессуальные инструкции	<code><?xml version="1.0"?></code>
Элемент	<code><drink type="beer">Carlsberg</drink></code>
Атрибут	<code>type="beer"</code>
Текст	<code>Carlsberg</code>

Чтобы просматривать примеры на этом сайте, вам понадобится Internet Explorer 5.0 или выше!

Парсирование DOM

Парсер Microsoft XML - это COM-компонент, который идет вместе с Internet Explorer 5.0. После инсталляции IE 5.0, парсер становится доступным для скриптов, встроенных в HTML-документы и размещенных в ASP-файлах

Парсер Microsoft XML

Чтобы создавать, читать и модифицировать XML-документ, вам необходим XML-парсер

Парсер Microsoft XMLDOM основывается на программной модели, которая:

- Поддерживает языки JavaScript, VBScript, Perl, VB, Java, C++ и другие
- Поддерживает спецификации W3C XML 1.0 и XML DOM
- Поддерживает DTD и проверку правильности документа (валидацию)

Если вы применяете JavaScript в IE 5.0, вы можете создать объект XML-документа с помощью следующего кода:

```
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
```

Если вы применяете VBScript, вы можете создать объект XML-документа с помощью следующего кода:

```
set xmlDoc = CreateObject("Microsoft.XMLDOM")
```

Если вы применяете VBScript в Active Server Page (ASP), вы можете использовать следующий код:

```
set xmlDoc = Server.CreateObject("Microsoft.XMLDOM")
```

Загрузка XML-файла в парсер

Следующий код загружает уже существующий XML-документ (note.xml) XML-парсер:

```
<script type="text/javascript">

var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load("note.xml")
// ..... здесь происходит обработка документа

</script>
```

В первой строке этого кода создается экземпляр парсера Microsoft XML. В третьей строке парсеру дается команда загрузить XML-документ, имеющий название note.xml. Во второй строке указывается, что парсер не продолжает обработку документа, пока он полностью не загрузится в память.

Попробуйте этот механизм в действии:²² ([open editor](#))

Загрузка в парсер чистого XML-текста

Следующий код загружает в XML-парсер текстовые строки:

22: http://www.w3schools.com/dom/tryit.asp?filename=note_parsertest

```
<script type="text/javascript">

var text="<note>"
text=text+"<to>Tove</to><from>Jani</from>"
text=text+"<heading>Reminder</heading>"
text=text+"<body>Don't forget me this weekend!</body>"
text=text+"</note>"

var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.loadXML(text)
// ..... здесь происходит обработка документа

</script>
```

Обратите внимание, что метод "loadXML" (в отличие от метода "load") применяется для загрузки строк текста.

Попробуйте в действии: ²³ (open editor)

Ошибки парсирования

Для получения информации об ошибках у парсера Microsoft XML, можно применять объект `parseError`

Объект `parseError`

При открытии XML-документа, XML-парсер может выдать сообщение об ошибке. Получив доступ к объекту `parseError`, можно получить код ошибки, текст, в котором она содержится и даже номер строки, вызвавшей ошибку.

Объект `parseError` не является частью стандарта W3C DOM.

Файловая ошибка

С помощью этого кода вы можете попробовать загрузить не-существующий файл и отобразить свойства этой ошибки:

```
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load("ksdjf.xml")

document.write("<br>Error Code: ")
document.write(xmlDoc.parseError.errorCode)
document.write("<br>Error Reason: ")
document.write(xmlDoc.parseError.reason)
document.write("<br>Error Line: ")
document.write(xmlDoc.parseError.line)
```

Попробуйте: ²⁴ (open editor)

23: http://www.w3schools.com/dom/tryit.asp?filename=note_parsertest2

24: http://www.w3schools.com/dom/tryit.asp?filename=note_error

Ошибка XML

Этот код загружает в парсер неправильный XML-документ:

```
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load("note_error.xml")

document.write("<br>Error Code: ")
document.write(xmlDoc.parseError.errorCode)
document.write("<br>Error Reason: ")
document.write(xmlDoc.parseError.reason)
document.write("<br>Error Line: ")
document.write(xmlDoc.parseError.line)
```

Вы можете узнать больше о правильных и пригодных XML-документах на нашей школе XML:²⁵ [\(open link\)](#)

Попробуйте этот код в действии:²⁶ [\(open editor\)](#)

Вы можете также просто открыть XML-файл:²⁷ [\(open editor\)](#)

Свойства объекта parseError

Свойство	Описание
<code>errorCode</code>	Возвращает код ошибки в формате long integer
<code>reason</code>	Возвращает строку текста, поясняющую причину ошибки
<code>line</code>	Возвращает номер строки, вызвавшей ошибку в формате long integer
<code>linePos</code>	Returns a long integer representing the line position for the error
<code>srcText</code>	Возвращает строку, содержащую строчку исходного кода, вызвавшую ошибку
<code>url</code>	Возвращает url, указывающий на загруженный документ
<code>filePos</code>	Returns a long integer file position of the error

Валидатор XML

Проверка правильности (валидация) вашего XML

Чтобы помочь вам проверить ваш XML, мы создали валидатор на основе парсера Microsoft. Вставьте XML-код в текстовую область и нажмите кнопку "Проверить":

25: http://xml.nsu.ru/xml/xml_home.xml

26: http://www.w3schools.com/dom/tryit.asp?filename=note_error_two

27: http://xml.nsu.ru/dom/note_error.xml



Данное интерактивное содержание документа может быть представлено только в варианте он-лайн. Он находится на <http://xml.nsu.ru>.

Проверка вашего XML-файла

Еще вы можете проверять свои XML-файлы просто вписав в это поле url своего XML-файла и нажав кнопку "Подача запроса":



Данное интерактивное содержание документа может быть представлено только в варианте он-лайн. Он находится на <http://xml.nsu.ru>.

Если вы хотите проверить XML-файл, заведомо свободный от ошибок, вы можете вставить в поле следующий адрес: http://www.w3schools.com/dom/cd_catalog.xml.

Обратите внимание: Если при попытке доступа к этому файлу вы получили сообщение об ошибке, в котором говорится что-то вроде "Access denied", это произошло потому что в установках безопасности Internet Explorer не позволяется получение доступа между доменами.

Получение доступа к DOM

Отдельные XML-элементы можно выделять из XML-документа двигаясь по дереву узлов, получать к ним доступ через их номер, через их имя

Движение по дереву узлов

Одним из распространенных способов выделения отдельных XML-элементов из XML-документа является движение по дереву узлов данного документа и извлечение текстового значения каждого элемента. Для демонстрации этого способа можно написать небольшой фрагмент программного кода, например, с применением конструктора for/each в языке VBScript:

```
set xmlDoc=CreateObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load("note.xml")

for each x in xmlDoc.documentElement.childNodes
  document.write(x.nodeName)
  document.write(": ")
  document.write(x.text)
next
```

Попробуйте управление деревом в действии:²⁸ ([open editor](#))

Еще один пример, управление деревом узлов CD-каталога:²⁹ ([open editor](#))

Наполнение HTML содержанием из XML-файлов

Одна из наиболее серьезных надежд, возлагаемых на XML связана с возможностью отделить HTML-документы от полезных данных, в них отображаемых. Используя встроенный в браузер XML-парсер, можно конструировать HTML-страницы как статические документы, в которых

28: http://www.w3schools.com/dom/tryit.asp?filename=note_traverse

29: http://www.w3schools.com/dom/tryit.asp?filename=cd_catalog_traverse

JavaScript динамически выводит данные.

Приводимый ниже код JavaScript считывает XML-данные из XML-документа и вписывает XML-данные в HTML-элементы:

```
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load("note.xml")

nodes = xmlDoc.documentElement.childNodes

to.innerText = nodes.item(0).text
from.innerText = nodes.item(1).text
header.innerText = nodes.item(2).text
body.innerText = nodes.item(3).text
```

Попробуйте этот способ в действии: ³⁰ (open editor)

Получение доступа к XML-элементам по имени

Адресация к элементам по номеру - не лучший способ извлечения XML-элементов из XML-документа. Лучше делать это с помощью имен элементов.

Приводимый ниже код JavaScript считывает XML-данные из XML-документа и вписывает XML-данные в HTML-элементы:

```
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load("note.xml")

to.innerText=
xmlDoc.getElementsByTagName("to").item(0).text
from.innerText=
xmlDoc.getElementsByTagName("from").item(0).text
header.innerText=
xmlDoc.getElementsByTagName("heading").item(0).text
body.innerText=
xmlDoc.getElementsByTagName("body").item(0).text
```

Попробуйте этот способ в действии: ³¹ (open editor)

Обратите внимание, что текстовый элемент узла является дочерним узлом элемента. Чтобы извлечь текст (Jani) из элемента, например, `<from>Jani</from>`, вы должны обращаться к дочернему узлу, вот так:

```
getElementsByTagName("from").item(0).text
```

а вот так неправильно:

```
getElementsByTagName("from").text
```

Объект XMLHttpRequest

Объект XMLHttpRequest предоставляет средства коммуникации с сервером на стороне клиента

30: <http://www.w3schools.com/dom/tryit.asp?filename=note>

31: http://www.w3schools.com/dom/tryit.asp?filename=note_getelement

Примеры

readyState:

Получение информации о текущем состоянии документа. Это свойство меняется по мере загрузки документа: ³² (open editor)

responseText:

Получение результата запроса в виде текстовой строки: ³³ (open editor)

status:

Получение статуса операции в виде кода: ³⁴ (open editor)

statusText:

Получение статуса операции в виде строки: ³⁵ (open editor)

Получение XML

Вот как можно получить XML-файл у сервера, используя объект XMLHttpRequest:

```
var xmlHttp = new XMLHttpRequest("Microsoft.XMLHTTP")
xmlHttp.open("GET", "note.xml", false)
xmlHttp.send()
xmlDoc=xmlHttp.responseText
```

Попробуйте в действии: ³⁶ (open editor)

Посылка XML

Кроме того, можно направить XML-документ расположенной на сервере ASP-странице, проанализировать запрос и отправить результат обратно клиенту:

```
var xmlHttp = new XMLHttpRequest("Microsoft.XMLHTTP")
xmlHttp.open("POST", "demo_dom_http.asp", false)
xmlHttp.send(xmlDoc)
document.write(xmlHttp.responseText)
```

ASP-страница, написанная на VBScript:

```
set xmldoc = Server.CreateObject("Microsoft.XMLDOM")
xmldoc.async=false
xmldoc.load(request)

for each x in xmldoc.documentElement.childNodes
    if x.NodeName = "to" then name=x.text
next

response.write(name)
```

Здесь результат посылается обратно клиенту с помощью свойства response.write.

Попробуйте механизм в действии: ³⁷ (open editor)

32: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_readystate

33: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_http

34: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_status

35: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_statustext

36: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_http

Важное замечание

В настоящее время объект Microsoft XMLHttpRequest может использоваться только браузером.

Серверный код, который попытается использовать этот объект для коммуникации с другими серверами, может функционировать не правильно.

Это баг в объекте XMLHttpRequest. Дополнительную информацию по этому вопросу можно получить в информационной базе Microsoft, статья Q237906:³⁸ ([open link](#))

По слухам, Microsoft устранил этот баг в выходящем релизе библиотеки XML. А пока вы можете использовать коммерчески доступный компонент ASPHTTP.

Свойства объекта XMLHttpRequest

Свойство	Описание
<code>readyState</code>	Возвращает состояние документа
<code>responseBody</code>	Возвращает ответ как массив не размеченных байтов
<code>responseStream</code>	Возвращает ответ как IStream
<code>responseText</code>	Возвращает ответ как текстовую строку
<code>responseXML</code>	Возвращает ответ как XML-документ
<code>status</code>	Возвращает код статуса в виде числа
<code>statusText</code>	Возвращает статус в виде текстовой строки

Методы объекта XMLHttpRequest

Метод	Описание
<code>abort()</code>	Прерывает текущий http-запрос
<code>getAllResponseHeaders()</code>	Возвращает значение http-заголовков
<code>getResponseHeader(headerName)</code>	Возвращает значение одного конкретного http-заголовка
<code>open(method, url, async, userid, password)</code>	Открывает http-запрос, указывает дополнительную информацию
<code>send()</code>	Отправляет http-запрос на сервер
<code>setRequestHeader(headerName, headerValue)</code>	Определяет имя http-заголовка

37: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_httpsend

38: <http://support.microsoft.com/support/kb/articles/Q237/9/06.ASP>

Типы узлов DOM

Примеры

Вот XML-файл `note_special.xml`, который используется в этих примерах: ³⁹ ([open xml](#))

NodeType:

Мы движемся по дереву узлов файла `note_special.xml` и получаем свойство `nodeType` (тип узла) его узлов: ⁴⁰ ([open editor](#))

nodeName:

Мы движемся по дереву узлов файла `note_special.xml` и получаем свойство `nodeName` (имя узла) его узлов: ⁴¹ ([open editor](#))

nodeValue:

Мы движемся по дереву узлов файла `note_special.xml` и получаем свойство `nodeValue` (значение узла) его узлов: ⁴² ([open editor](#))

NodeTypeString:

В IE5 вы можете также получить свойство `nodeType` в виде текстовой строки - с помощью свойства `.nodeTypeString`: ⁴³ ([open editor](#))

Типы узлов

Узлы делятся на различные типы. Мы приводим список типов узлов и значение их свойств `.nodeName` и `.nodeValue`. Кроме того, в Internet Explorer 5 можно использовать свойство `.nodeTypeString`, чтобы получить значение свойства `nodeType` в виде текстовой строки.

nodeType	nodeTypeString	nodeName	nodeValue
1	element	имя тэга	null
2	attribute	имя атрибута	значение атрибута
3	text	#text	содержимое узла
4	cdatasection	#cdatasection	содержимое узла
5	entityreference	имя ссылки на сущность	null
6	entity	имя сущности	null
7	processinginstruction	target	содержимое узла
8	comment	#comment	текст комментария
9	document	#document	null
10	documenttype	имя типа документа	null
11	documentfragment	#document fragment	null

39: http://xml.nsu.ru/dom/note_special.xml

40: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodetype

41: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodetype_name

42: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_type_value

43: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodetypestring

12	notation	имя нотации	null
----	----------	-------------	------

Типы узлов - именованные константы

Тип узла	Имя константы
1	ELEMENT_NODE
2	ATTRIBUTE_NODE
3	TEXT_NODE
4	CDATA_SECTION_NODE
5	ENTITY_REFERENCE_NODE
6	ENTITY_NODE
7	PROCESSING_INSTRUCTION_NODE
8	COMMENT_NODE
9	DOCUMENT_NODE
10	DOCUMENT_TYPE_NODE
11	DOCUMENT_FRAGMENT_NODE
12	NOTATION_NODE

Объект Node (узел)

Примеры

Вот XML-файл note.xml, который используется в этих примерах: ⁴⁴ (open xml)

nodeName:

Как получить имя узла: ⁴⁵ (open editor)

nodeValue

Как получить значение узла: ⁴⁶ (open editor)

nextSibling:

Как получить имя узла nextSibling: ⁴⁷ (open editor)

Text:

В IE5 вы можете получить текст из узла и всех его дочерних узлов: ⁴⁸ (open editor)

xml:

В IE5 вы можете получить узел и все его дочерние узлы в виде XML: ⁴⁹ (open editor)

44: <http://xml.nsu.ru/dom/note.xml>

45: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodename

46: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodevalue

47: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nextsibling

48: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_text

appendChild:

Как создать элементный узел, содержащий текстовый узел и прикрепить его в качестве дочернего узла: ⁵⁰ ([open editor](#))

insertBefore:

Как создать текстовый узел и вставить его перед заданным узлом: ⁵¹ ([open editor](#))

Объект node

Объект node представляет любой узел любого дерева узлов. Узел может быть элементным узлом, текстовым узлом или быть люого другого типа из перечисленных в предыдущем разделе. Все эти типы узлов имеют свойства и методы. Ниже перечислены общие для всех типов узлов свойства и методы.

Свойства узла

Имя	Описание
<code>attributes</code>	Возвращает NamedNodeMap (карта именованных узлов), содержащий все атрибуты данного узла
<code>childNodes</code>	Возвращает NodeList (список узлов) содержащий список всех дочерних узлов данного узла
<code>firstChild</code>	Возвращает первый дочерний узел данного узла
<code>lastChild</code>	Возвращает последний дочерний узел данного узла
<code>nextSibling</code>	Возвращает следующий узел, дочерний элемент того же родительского элемента
<code>nodeName</code>	Возвращает nodeName (имя узла), в зависимости от его типа
<code>nodeType</code>	Возвращает nodeType (тип узла) в виде числа
<code>nodeValue</code>	Возвращает или устанавливает значение данного узла, в зависимости от его типа
<code>ownerDocument</code>	Возвращает корневой узел документа
<code>parentNode</code>	Возвращает родительский узел данного узла
<code>previousSibling</code>	Возвращает предыдущий дочерний элемент, обладающий тем же родительским элементом, что и данный

Методы узла

Имя	Описание
<code>appendChild(newChild)</code>	Привязывает новый узел newChild как последний в списке дочерних узлов данного узла
<code>cloneNode(boolean)</code>	Создает точный клон текущего узла. Если булевское значение равно "true", клонированный узел также содержит и все дочерние элементы узла - прототипа
<code>hasChildNodes()</code>	Возвращает значение "true", если узел имеет хотя бы один дочерний узел

49: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_xml

50: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_appendchild

51: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_text_createtext

<code>insertBefore(newNode, refNode)</code>	Вставляет новый узел, newNode, перед существующим узлом refNode
<code>removeChild(nodeName)</code>	Удаляет определенный узел nodeName (имя узла)
<code>replaceChild(newNode, oldNode)</code>	Заменяет старый oldNode новым узлом newNode

Свойства узла в IE5

Объект node обладает некоторыми свойствами и методами, которые определены только в Internet Explorer 5. В этом и следующем параграфе дано их краткое описание.

Имя	Описание
<code>basename</code>	Возвращает nodeName (имя узла) без указания пространства имен
<code>dataType</code>	Возвращает или устанавливает dataType (тип данных) для этого узла
<code>definition</code>	?
<code>nodeTypeString</code>	Возвращает nodeType (тип узла) в виде текстовой строки
<code>nodeTypedValue</code>	?
<code>specified</code>	Указывает, специфицирован ли nodeValue (значение узла) в DTD/Schema или нет
<code>text</code>	Возвращает или устанавливает text для этого узла и всех его дочерних узлов
<code>xml</code>	Возвращает или устанавливает xml для этого узла и всех его дочерних узлов

Методы узла в IE5

Имя	Описание
<code>selectNodes(pattern)</code>	?
<code>selectSingleNode(pattern)</code>	?
<code>transformNode(stylesheet)</code>	Обрабатывает узел и его childNodes (дочерние узлы) посредством заданной таблицы стилей XSL и возвращает результат

Объект NodeList (список узлов)

Примеры

Вот XML-файл note.xml, который используется в этих примерах: ⁵² (open xml)

length:

Как получить число узлов в nodeList: ⁵³ (open editor)

52: <http://xml.nsu.ru/dom/note.xml>

53: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_length

item:

Как получить конкретный узел из nodeList: ⁵⁴ (open editor)

nextNode():

Позволяет вам получить следующий узел из nodeList: ⁵⁵ (open editor)

reset():

Позволяет сбросить указатель на первый узел в nodeList: ⁵⁶ (open editor)

Объект NodeList

Объект nodeList представляет узел и его дочерние узлы в виде дерева. Ниже перечислены свойства и методы объекта nodeList.

Свойства узла NodeList

Имя	Описание
<code>length</code>	Возвращает число узлов в nodeList

Методы узла NodeList

Имя	Описание
<code>item</code>	Возвращает заданный узел из nodeList

Методы объекта NodeList IE5

В Internet Explorer 5 объект nodeList обладает некоторыми дополнительными методами:

Имя	Описание
<code>nextNode()</code>	Возвращает следующий объект в списке узлов
<code>reset()</code>	Устанавливает указатель на первый узел в списке узлов

Объект Document (документ)

Примеры

Вот XML-файл note.xml, который используется в этих примерах: ⁵⁷ (open xml)

54: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_item

55: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nextnode

56: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_reset

documentElement:

Как получить имя узла корневого элемента: [58](#) (open editor)

createCDATASection:

Как создать узел CDATA и затем добавить его к nodeList: [59](#) (open editor)

createComment:

Как создать узел-комментарий и затем добавить его к nodeList: [60](#) (open editor)

createElement:

Как создать элемент и затем добавить его к nodeList: [61](#) (open editor)

createTextNode:

Как создать текстовый узел и затем добавить его к nodeList: [62](#) (open editor)

getElementsByTagName:

Как получить значение определенного узла: [63](#) (open editor)

Объект Document

Объект document - корневой элемент дерева узлов. Все остальные узлы являются childNodes (дочерними узлами) элемента document. Элемент document должен иметься во всех XML-документах. Ниже перечислены свойства и методы объекта Document.

Свойства объекта Document

Имя	Описание
<code>documentElement</code>	Возвращает корневой элемент документа
<code>doctype</code>	Возвращает DTD или Schema для документа
<code>implementation</code>	Возвращает объект implementation для данного конкретного элемента document

Методы объекта Document

Имя	Описание
<code>createAttribute(attributeName)</code>	Создает узел attribute с заданным именем атрибута
<code>createCDATASection(text)</code>	Создает секцию CDATASection, содержащую заданный текст
<code>createComment(text)</code>	Создает узел-комментарий, содержащий заданный текст
<code>createDocumentFragment()</code>	Создает пустой объект documentFragment
<code>createElement(tagName)</code>	Создает элемент с созданным именем тэга

57: <http://xml.nsu.ru/dom/note.xml>

58: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_documentelement

59: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_cdata_create

60: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcomment

61: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement

62: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createtextnode

63: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelebytagname

<code>createEntityReference (referenceName)</code>	Создает ссылку на сущность entityReference с заданным именем ссылки
<code>createProcessingInstruction (target, text)</code>	Создает узел процессуальных инструкций processingInstruction node, содержащий заданную цель и текст
<code>createTextNode (text)</code>	Создает текстовый узел, содержащий заданный текст
<code>getElementsByTagName (tagName)</code>	Возвращает заданный узел и все его дочерние узлы в виде списка узлов nodeList

Объект Element (элемент)

Примеры

Вот XML-файл note.xml, который используется в этих примерах: ⁶⁴ (open xml)

tagName:

Как получить имя тэга-узла: ⁶⁵ (open editor)

getElementsByTagName:

Как получить значение заданного узла: ⁶⁶ (open editor)

getAttribute:

Как получить значение атрибута: ⁶⁷ (open editor)

setAttribute:

Как изменить значение атрибута: ⁶⁸ (open editor)

setAttribute (2):

Как установить новый атрибут и его значение: ⁶⁹ (open editor)

Объект Element

Объект element представляет элементные узлы документа. Если элементный узел содержит текст, этот текст будет представлен в текстовом узле. Ниже перечислены свойства и методы объекта Element.

Свойства объекта Element

Имя	Описание
<code>tagName</code>	Возвращает или устанавливает имя узла

Методы объекта Element

64: <http://xml.nsu.ru/dom/note.xml>

65: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_tagname

66: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelebytagname

67: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getattribute

68: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_changeattribute

69: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_setattribute

Имя	Описание
<code>getAttribute (attributeName)</code>	Возвращает значение заданного атрибута
<code>getAttributeNode (attributeName)</code>	Возвращает заданный узел-атрибут как объект
<code>getElementsByTagName (tagName)</code>	Возвращает заданный узел и все его дочерние узлы в виде списка узлов <code>nodeList</code>
<code>normalize ()</code>	Помещает текстовые узлы данного элемента и всех его дочерних узлов в один текстовый узел. Не возвращает ничего
<code>removeAttribute (attributeName)</code>	Удаляет значение заданного атрибута. Если атрибут имеет значение по умолчанию, оно устанавливается
<code>removeAttributeNode (attributeNode)</code>	Удаляет заданный атрибутный узел. Если атрибутный узел имеет значение по умолчанию, этот атрибут вставляется
<code>setAttribute (attributeName, attributeValue)</code>	Вставляет новый атрибут
<code>setAttributeNode (attributeNodeName)</code>	Вставляет новый атрибутный узел

Объект Attr (атрибут)

Примеры

Вот XML-файл `note.xml`, который используется в этих примерах: ⁷⁰ (open xml)

name:

Как получить имя атрибута: ⁷¹ (open editor)

value:

Как получить значение атрибута: ⁷² (open editor)

specified:

Возвращается значение `true`, если значение устанавливается в документе, `false`, если значение определяется по умолчанию в DTD/Schema: ⁷³ (open editor)

Объект Attr

Объект `attr` возвращает атрибут элементного объекта как атрибутный узел. Объект `attr` имеет те же самые свойства и методы, что и остальные узлы в целом. Свойства, специфичные для объекта `Attr`, перечислены ниже.

Свойства объекта Attr

70: <http://xml.nsu.ru/dom/note.xml>

71: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_name

72: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_value

73: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_specified

Имя	Описание
<code>name</code>	Возвращает или устанавливает имя атрибута
<code>specified</code>	Возвращает булевское значение, показывающее, установлено ли в документе значение узла
<code>value</code>	Возвращает или устанавливает значение атрибута

Объект текст (Text)

Примеры

Вот XML-файл `note.xml`, который используется в этих примерах: ⁷⁴ ([open xml](#))

splitText:

Метод `splitText` разбивает текст на заданном символе и возвращает оставшийся текст: ⁷⁵ ([open editor](#))

createTextNode:

Как создать текстовый узел: ⁷⁶ ([open editor](#))

Объект Text

Текстовый объект представляет текст внутри элемента как узла. Ниже перечислены методы объекта `text`.

Методы объекта Text

Имя	Описание
<code>splitText(number)</code>	Разбивает текст на заданном символе и возвращает оставшийся текст

Объект CDATASection

Примеры

Вот XML-файл `note.xml`, который используется в этих примерах: ⁷⁷ ([open xml](#))

createCDATASection:

Как создать узел типа `CDATASection`: ⁷⁸ ([open editor](#))

74: <http://xml.nsu.ru/dom/note.xml>

75: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_splittext

76: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_text_createtext

77: <http://xml.nsu.ru/dom/note.xml>

Объект CDATASection

Объект CDATASection представляет в документе узлы типа CDATASection. Узел CDATASection используется для того, чтобы включить в документ не-парсируемый текст, который обычно трактуется как разметка.

Объект Comment (комментарий)

Примеры

Вот XML-файл note.xml, который используется в этих примерах: [79](#) (open xml)

createComment:

Как создать узел-комментарий: [80](#) (open editor)

Объект Comment

Объект comment представляет в документе узлы-комментарии. Узел этого типа не имеет свойства nodeName (имя объекта), но имеет свойство nodeValue (значение узла) - оно равно тексту комментария.

Примеры XML DOM

Вот XML-файл note.xml, который используется в этих примерах: [81](#) (open xml)

Парсер XML

Загрузка XML-файла в парсер: [82](#) (open editor)

Загрузка в парсер чистого XML-текста: [83](#) (open editor)

Получение доступа к DOM

Движение по дереву узлов документа note.xml: [84](#) (open editor)

Наполнение HTML содержимым документа note.xml: [85](#) (open editor)

Получение доступа к XML-элементам по имени: [86](#) (open editor)

Вот XML-файл note_error.xml, который используется в этих примерах: [87](#) (open xml)

Ошибки парсирования

78: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_cdata_create

79: <http://xml.nsu.ru/dom/note.xml>

80: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_comment_create

81: <http://xml.nsu.ru/dom/note.xml>

82: http://www.w3schools.com/dom/tryit.asp?filename=note_parsertest

83: http://www.w3schools.com/dom/tryit.asp?filename=note_parsertest2

84: http://www.w3schools.com/dom/tryit.asp?filename=note_traverse

85: <http://www.w3schools.com/dom/tryit.asp?filename=note>

86: http://www.w3schools.com/dom/tryit.asp?filename=note_getelement

87: http://xml.nsu.ru/dom/note_error.xml

Не правильный XML-документ: ⁸⁸ (open editor)

Вот XML-файл note_special.xml, который используется в этих примерах: ⁸⁹ (open xml)

Типы узлов

Движение по XML-файлу и получение типа узлов (nodeType): ⁹⁰ (open editor)

Движение по XML-файлу и получение имени тех же узлов (nodeName): ⁹¹ (open editor)

Движение по XML-файлу и получение значения тех же узлов (nodeValue): ⁹² (open editor)

Получение типа узла в виде строки текста (свойство TypeString в IE5): ⁹³ (open editor)

Вот XML-файл note.xml, который используется в этих примерах: ⁹⁴ (open xml)

Объект HttpRequest

Получение состояния документа (readyState): ⁹⁵ (open editor)

Получение результата запроса в виде текста (responseText): ⁹⁶ (open editor)

Получение статуса операции в виде кода (status): ⁹⁷ (open editor)

Получение статуса операции в виде текстовой строки (statusText): ⁹⁸ (open editor)

Объект узел (Node)

Получение имени узла (nodeName): ⁹⁹ (open editor)

Получение значения узла (nodeValue): ¹⁰⁰ (open editor)

Получение следующего дочернего ула того же родительского узла (nextSibling): ¹⁰¹ (open editor)

Получение текста из узла и всех его дочерних узлов (свойство text в IE5): ¹⁰² (open editor)

Получение XML из узла и всех его дочерних узлов (свойство xml в IE5): ¹⁰³ (open editor)

Создание текстового узла и привязка его в виде дочернего узла (appendChild): ¹⁰⁴ (open editor)

Создание текстового узла и вставка его перед заданным узлом (insertBefore): ¹⁰⁵ (open editor)

Объект список узлов (NodeList)

Получение номера узла в списке узлов nodeList (length): ¹⁰⁶ (open editor)

Получение заданного узла из списка узлов nodeList (item): ¹⁰⁷ (open editor)

88: http://www.w3schools.com/dom/tryit.asp?filename=note_error_two

89: http://xml.nsu.ru/dom/note_special.xml

90: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodetype

91: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodetype_name

92: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_type_value

93: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodetypestring

94: <http://xml.nsu.ru/dom/note.xml>

95: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_readystate

96: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_http

97: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_status

98: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_statustext

99: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodename

100: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodevalue

101: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nextsibling

102: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_text

103: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_xml

104: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_appendchild

105: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_text_createtext

106: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_length

107: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_item

Получение следующего узла в списке узлов (метод `nextNode()` в IE5): ¹⁰⁸ (open editor)

Сброс указателя на первый узел в списке узлов `nodeList` (метод `reset()` в IE5): ¹⁰⁹ (open editor)

Объект документ (Document)

Получение имени узла корневого элемента (`documentElement`): ¹¹⁰ (open editor)

Создание узла типа CDATA и его вставка в список узлов `nodeList` (`createCDATASection`): ¹¹¹ (open editor)

Создание узла-комментария и вставка его в список узлов `nodeList` (`createComment`): ¹¹² (open editor)

Создание элементного узла и вставка его в список узлов `nodeList` (`createElement`): ¹¹³ (open editor)

Создание текстового узла и вставка его в список узлов `nodeList` (`createTextNode`): ¹¹⁴ (open editor)

Получение значения заданного узла (`getElementsByTagName`): ¹¹⁵ (open editor)

Объект элемент (Element)

Получение имени тэга узла (`tagName`): ¹¹⁶ (open editor)

Получение значения заданного узла (`getElementsByTagName`): ¹¹⁷ (open editor)

Получение значения атрибута (`getAttribute`): ¹¹⁸ (open editor)

Изменение значения атрибута (`setAttribute`): ¹¹⁹ (open editor)

Создание нового атрибута и установка его значения (`setAttribute`): ¹²⁰ (open editor)

Объект атрибут (Attr)

Получение имени атрибута (`name`): ¹²¹ (open editor)

Получение значения атрибута (`value`): ¹²² (open editor)

Проверка, установлено ли значения атрибута в документе или определено по умолчанию в DTD/Schema (`specified`): ¹²³ (open editor)

Объект текст (Text)

Разбиение текста и получение оставшейся части текста (`splitText`): ¹²⁴ (open editor)

Создание текстового узла (`createTextNode`): ¹²⁵ (open editor)

- 108: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nextnode
- 109: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_reset
- 110: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_documentelement
- 111: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_cdata_create
- 112: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcomment
- 113: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement
- 114: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createtextnode
- 115: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelebytagname
- 116: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_tagname
- 117: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelebytagname
- 118: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getattribute
- 119: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_changeattribute
- 120: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_setattribute
- 121: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_name
- 122: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_value
- 123: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_specified
- 124: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_splittext
- 125: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_text_createtext

Объект CDATASection

Создание узла типа CDATA (createCDATASection): ¹²⁶ (open editor)

Объект комментариев (Comment)

Создание узла-комментария (createComment): ¹²⁷ (open editor)

Интернет-ресурсы по DOM

Список имеющихся в Интернете важных ресурсов по DOM

W3C-ресурсы по DOM

Консорциум W3C был создан в 1994 с целью упорядочивания и стандартизации WWW-протоколов, таких, как HTML, CSS и XML.

Среди сервисных служб, поддерживаемых W3C имеются: хранилище информации, относящейся к WWW для разработчиков и пользователей, реализация кодовых ссылок для встраивания в документы и продвижения стандартов, разработка различных прототипов для демонстрации новых технологий.

W3C-ресурсы:

- [The DOM Activity](#)¹
- [DOM Level 2 Core Specification](#)²
- [DOM Level 2 HTML Specification](#)³
- [DOM Level 2 Views Specification](#)⁴
- [DOM Level 2 Style Specification](#)⁵
- [DOM Level 2 Events Specification](#)⁶
- [DOM Level 2 Traversal-Range Specification](#)⁷

Developed by [Metaphor](#) (c) 2002

126: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_cdata_create

127: http://www.w3schools.com/dom/tryit.asp?filename=try_dom_comment_create

1: <http://www.w3.org/DOM/Activity>

2: <http://www.w3.org/TR/2000/PR-DOM-Level-2-Core-20000927/>

3: <http://www.w3.org/TR/2000/PR-DOM-Level-2-HTML-20000927/>

4: <http://www.w3.org/TR/2000/PR-DOM-Level-2-Views-20000927/>

5: <http://www.w3.org/TR/2000/PR-DOM-Level-2-Style-20000927/>

6: <http://www.w3.org/TR/2000/PR-DOM-Level-2-Events-20000927/>

7: <http://www.w3.org/TR/2000/PR-DOM-Level-2-Traversal-Range-20000927/>