

Лабораторная работа № 12

СОЗДАНИЕ И ВАЛИДАЦИЯ XML-ДОКУМЕНТОВ

Цель работы: изучить правила создания XML-документов, получить навыки создания валидных документов и их проверка.

Теоретические сведения для выполнения работы

Основы XML

XML (eXtensible Markup Language) — расширяемый язык разметки, созданный для хранения, транспортировки и обмена данными. Он включает в себя элементы SGML и предназначен для определения HTML-подобных языков. В нем отсутствуют элементы SGML, не применимые к языкам типа HTML, а другие элементы упрощены, чтобы облегчить их понимание и использование. В языке XML нет предопределенных тегов, автор определяет свои языковые теги и свою структуру документа. Формально XML представляет собой набор правил для создания собственных языков разметки, а также чтения и написания документов на языке разметки. Пример XML-документа представлен на рис. 12.1

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Анна</to>
  <from>Дмитрий</from>
  <heading>Напоминание</heading>
  <body>Не забудь обо мне в эти выходные!</body>
</note>
```

Рис. 12.1 Пример структуры XML-документа

Согласно примеру, XML-документ состоит из элементов, которые представляют собой контент между начальным и конечным тегом. Первые строки называются прологом и объявляют процессору XML, что данный документ размечен в XML. Пролог включает объявление (XML-декларация), которое начинается с разделителя, состоящего из пяти символов `<?xml`, за которым следует некоторое

количество определений свойств, каждое из которых состоит из имени свойства и его значения, заключенного в кавычки. Объявление завершается закрывающим разделителем из двух символов `?>`. Варианты объявлений представлены на рис. 12.2

```
<?xml version="1.0"?>  
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>  
<?xml version = '1.0' encoding= 'UTF-8' standalone ="no"?>
```

Рис. 12.2 Варианты объявлений XML-документа

На рис. 12.1 элемент документа `<note>` является корневым. Следующие 4 строки описывают дочерние элементы корневого элемента: `<to>Анна</to>`, `<from>Дмитрий</from>`, `<heading>Напоминание</heading>`, `<body>Не забудь обо мне в эти выходные!</body>`. Последняя строка определяет конец корневого элемента: `</note>`. В общем виде структуру XML-документа после пролога можно представить в следующем виде:

```
<корневой>  
  <потомок>  
    <подпотомок>.....</подпотомок>  
  </потомок>  
</корневой>
```

Рис. 12.3 Структура XML-документа

При открытии в браузерах, например Mozilla или Opera, будет отображаться согласно рис. 12.4

С этим XML-файлом не связана ни одна таблица стилей. Ниже показано дерево элементов.

```
<note>  
  <to>Анна</to>  
  <from>Дмитрий</from>  
  <heading>Напоминание</heading>  
  <body>Не забудь обо мне в эти выходные!</body>  
</note>
```

Рис. 12.4 Пример XML-документа в браузере

Таким образом, необходимо подключение таблицы стилей для создания конечного продукта после объявления XML-документа. Пример XML-документа с подключением CSS представлен на рис. 12.5

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="forXML.css"?>
<books>
<head>
  <author>Автор</author>
  <title>Название книги</title>
</head>
<book>
  <author>С.В. Одиноккина</author>
  <title>Основы технологий XML</title>
</book>
<book>
  <author>Эрик Рэй</author>
  <title>Изучаем XML</title>
</book>
</books>
```

а

```
books {
  display:table;
  border-collapse: collapse;
  margin:10px; }
book, head {
  display:table-row; }
ntitle, nauthor {
  display: table-cell;
  border:1px solid black;
  padding: 5px;
  text-align: center;
  background: #ffff00; }
title, author {
  display: table-cell;
  border:1px solid black;
  padding: 5px; }
```

б

Автор	Название книги
С.В. Одиноккина	Основы технологий XML
Эрик Рэй	Изучаем XML

в

а – XML-документ, б – CSS-файл,
в – результат в браузере с подключением CSS

Рис. 12.5 Пример оформления XML-документа

Также следует различать синтаксически верный документ и валидные документы. К синтаксически верным (корректным) относятся XML-документы, которые соответствуют следующим правилам синтаксиса XML:

1. Все XML элементы должны иметь закрывающий тег.
2. Теги XML являются регистрозависимыми.
3. Перед закрывающей угловой скобкой в пустых элементах XML требуется ставить косую черту.
4. Значения должны быть заключены в одинарные или двойные кавычки.
5. Все элементы обязаны соблюдать корректную вложенность.

6. XML документ должен содержать один корневой элемент, который будет родительским для всех других элементов.

7. Учитываются все символы форматирования (пробелы, переводы строк, табуляции не игнорируются, как в HTML).

Имена элементов могут начинаться только с букв и символов подчеркивания и могут содержать только буквы, цифры, дефисы, точки и символы подчеркивания и не могут начинаться с сочетания «xml». В качестве имен можно использовать любые слова, зарезервированных слов нет.

Валидные XML-документы

Валидный XML-документ должен быть синтаксически верным и соответствовать одному из типов определения документов. Правила, определяющие допустимые элементы и атрибуты для XML документа, называются определениями документа или схемами документа.

С XML можно использовать различные типы определений документа:

- оригинальное определение типа документа (Document Type Definition, DTD).

- XML схема — тип определений, основанный на XML схема.

DTD (Document Type Definition, определение типа документа) — это язык описания структуры XML-документа, который используется для проверки грамматики XML-документа и его соответствия определенному типу. Цель DTD состоит в том, чтобы определить структуру XML документа. Это делается путем определения списка допустимых элементов. В прологе декларация объявления элементов может быть внутренняя или внешняя

Таблица 12.1

Виды деклараций DTD

Внутренняя декларация DTD	Внешняя декларация DTD
<pre><?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE note [<!ELEMENT note (to, from, Sbj, msg)> <!ELEMENT to (#PCDATA)> <!ELEMENT from (#PCDATA)></pre>	<pre><?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE note SYSTEM "note.dtd"> <note> <to>Sunny</to> <from>Oliver</from> <Sbj>Hello</Sbj></pre>

Внутренняя декларация DTD	Внешняя декларация DTD
<pre> <!ELEMENT Sbj (#PCDATA)> <!ELEMENT msg (#PCDATA)>]> <note> <to>Sunny</to> <from>Oliver</from> <Sbj>Hello</Sbj> <msg>This is a good day!</msg> </note> </pre>	<pre> <msg>This is a good day!</msg> </note> </pre> <p>файл note.dtd содержит:</p> <pre> <!ELEMENT note (to,from,heading,body)> <!ELEMENT to (#PCDATA)> <!ELEMENT from (#PCDATA)> <!ELEMENT heading (#PCDATA)> <!ELEMENT body (#PCDATA)> </pre>

DTD в приведенном примере интерпретируется следующим образом:

1. !DOCTYPE note определяет, что корневым элементом документа является note.
2. !ELEMENT note определяет, что элемент note содержит четыре элемента: to, from, heading, body.
3. !ELEMENT to определяет, что элемент to должен быть типа "#PCDATA".
4. !ELEMENT from определяет, что элемент from должен быть типа "#PCDATA".
5. !ELEMENT heading определяет, что элемент heading должен быть типа "#PCDATA".
6. !ELEMENT body определяет, что элемент body должен быть типа "#PCDATA".

#PCDATA означает разбираемые текстовые данные. Кроме инструкции ELEMENT существуют следующие:

- ATTLIST для перечисления и объявления атрибутов, которые могут принадлежать элементу.
- ENTITY для определения сущностей в DTD с целью их использования как в связанном с DTD XML-документе, так и собственно в DTD.

```

<?xml version="1.0"?>
<!DOCTYPE note [
<!ENTITY name "Hello, world!">
]>
<element>&name;</element>

```

Рис. 12.6 Пример использования сущности

Сущность является заместителем содержания, которую можно однажды объявить и многократно использовать почти в любом месте документа. Встроенные сущности приведены в таблице 12.2

Таблица 12.2

Многократно используемые встроенные сущности

Сущность	Ссылка на сущность	Значение
lt	<	< (меньше чем)
gt	>	> (больше чем)
amp	&	& (амперсанд)
apos	'	' (апостроф или одиночная кавычка)
quot	"	" (двойная кавычка)

К основным недостаткам DTD схем относят синтаксические отличия от языка XML, а также отсутствие возможностей работы с типами данных. Для решения этих недостатков предлагается использовать XML Schema, который представляет из себя язык XML Schema Definition Language и предназначен для создания файлов с расширением .xsd. Все элементы XSD начинаются с префикса **xsd:**, который указывается для пространства имен XSD, объявленного в корневом элементе экземпляра схемы.

Пример использования XML Schema представлен в таблице 12.3.

Таблица 12.3

Пример XML Schema

XML-документ	XML Schema
<pre><?xml version="1.0" encoding="UTF-8"?> <note xmlns:xsi="http://www.w3.org /2001/XMLSchema-instance" xsi:noNamespaceSchemaLocat ion="note.xsd"> <to>Sunny</to> <from>Oliver</from> <Sbj>Hello</Sbj> <msg>This is a good day!</msg> </note></pre>	<pre><?xml version="1.0" encoding="UTF-8" ?> <xsd:schema xmlns:xs="http://www.w3.org/2001 /XMLSchema"> <xsd:element name="note"> <xsd:complexType> <xsd:sequence> <xsd:element name="to" type="xs:string"/> <xsd:element name="from" type="xs:string"/> <xsd:element name="heading" type="xs:string"/> <xsd:element name="body" type="xs:string"/> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:schema></pre>

Приведенная выше схема интерпретируется следующим образом:

1. `<xs:element name="note">` определяет элемент "note";
2. `<xs:complexType>` у элемента "note" комплексный тип;
3. `<xs:sequence>` последовательность элементов элемента ;
4. `<xs:element name="to" type="xs:string">` у элемента "to" строковый тип (текст);
5. `<xs:element name="from" type="xs:string">` у элемента "from" строковый тип;
6. `<xs:element name="heading" type="xs:string">` у элемента "heading" строковый тип;
7. `<xs:element name="body" type="xs:string">` у элемента "body" строковый тип.

Как видно из примера, каждая XML схема состоит с корневого элемента «schema» и обязательного пространства имен «<http://www.w3.org/2001/XMLSchema>», а далее идет описание схемы.

Следует отметить, что XML Schema описывают структуру XML-документа. XML-документ, прошедший проверку по XML Schema, является синтаксически верным и валидным, а также имеет ряд преимуществ перед DTD:

- XML схема пишется на XML;
- XML схема легко расширяется;
- XML схема поддерживает типы данных;
- XML схема поддерживает пространства имен.

Задания к лабораторной работе № 12

Задание 1 Создать корректный XML-документ, который будет содержать информацию минимум о 5 товарах интернет-магазина. При выполнении задания для внешнего оформления используйте css.

Задание 2 Преобразуйте XML-документ из задания 1 в XML DTD, предварительно сделав его копию. Должно быть внутреннее объявление DTD, в котором должны быть использованы сущности, которые затем применяются в XML-документе.

Задание 3 Преобразуйте XML-документ из задания 1 в XML Schema, предварительно сделав его копию.

Задания 4 Провести проверку валидности XML-документов, подтвердив скриншотами результатов.\

Контрольные вопросы

1. Что такое XML?
2. Чем является первая строка в коде XML-документа?
3. Назовите правила синтаксиса XML-документа?
4. Какие должны быть имена элементов?
5. Как вы понимаете правило соблюдения корректной вложенности?
6. Какие элементы являются корневыми в XML-документе?
7. Для чего необходима валидация XML-документа?
8. В чем заключается разница между простыми и комплексными типами элементов в XML Schema?
9. В чем заключается преимущество Schema перед DTD?
10. Какие типы элементов XML Schema вы знаете?
11. Что означает *xs:element*?
12. Что означает *xs:complexType*?
13. В чем заключается предназначение *xs:sequence*?
14. Для чего предназначен *xs:attribute*?
15. Для чего используется ENTITY?
16. Что такое сущности? Каким образом вы использовали сущности?
17. Какие встроенные сущности вы знаете?
18. Для чего используется ATTLIST?
19. Что такое валидный XML-документ? Как типы вы знаете?
20. Что означает следующая запись `<!ENTITY name "Hello, world!">`?
21. Что означает следующая запись `<!ELEMENT to (#PCDATA)>`?
22. Какие параметры и значения имеет инструкция ATTLIST?
23. Какие параметры имеет инструкция ELEMENT?
24. Что находится на <http://www.w3.org/2001/XMLSchema>?
25. Каким образом можно объявить DTD? Как расшифровывается аббревиатура?