Focused forward

Netcracker

Лекция 6.
XML. XML Schema. SVG. WSDL. SOAP.

Web System Design and Development

План лекции

- Что такое XML?
- Создание и просмотр XML документов.
- Правила и синтаксис.
- Структура XML документа.
- Связь с другими технологиями.
- Применение XML.
- Понятие и особенности XML Schema.
- SVG.
- Понятие WSDL.
- Структура WSDL.
- Протокол SOAP.
- Основные понятия SOAP.
- Взаимодействие технологий.

XML

Что такое XML?

XML (eXtensible Markup Language) – расширяемый язык разметки

```
<?xml?>
```

Стандарт построения языков разметки иерархически структурированных данных для обмена между разными приложениями, в частности через Интернет

XML ничего не делает. Он создан для структурирования, хранения и передачи информации, а не для отображения.

Теги не предопределены, нужно создавать свои собственные теги.

XML образует целое семейство технологий.

XML свободный, платформенно-независимый, имеет широкую поддержку.

Что такое XML?

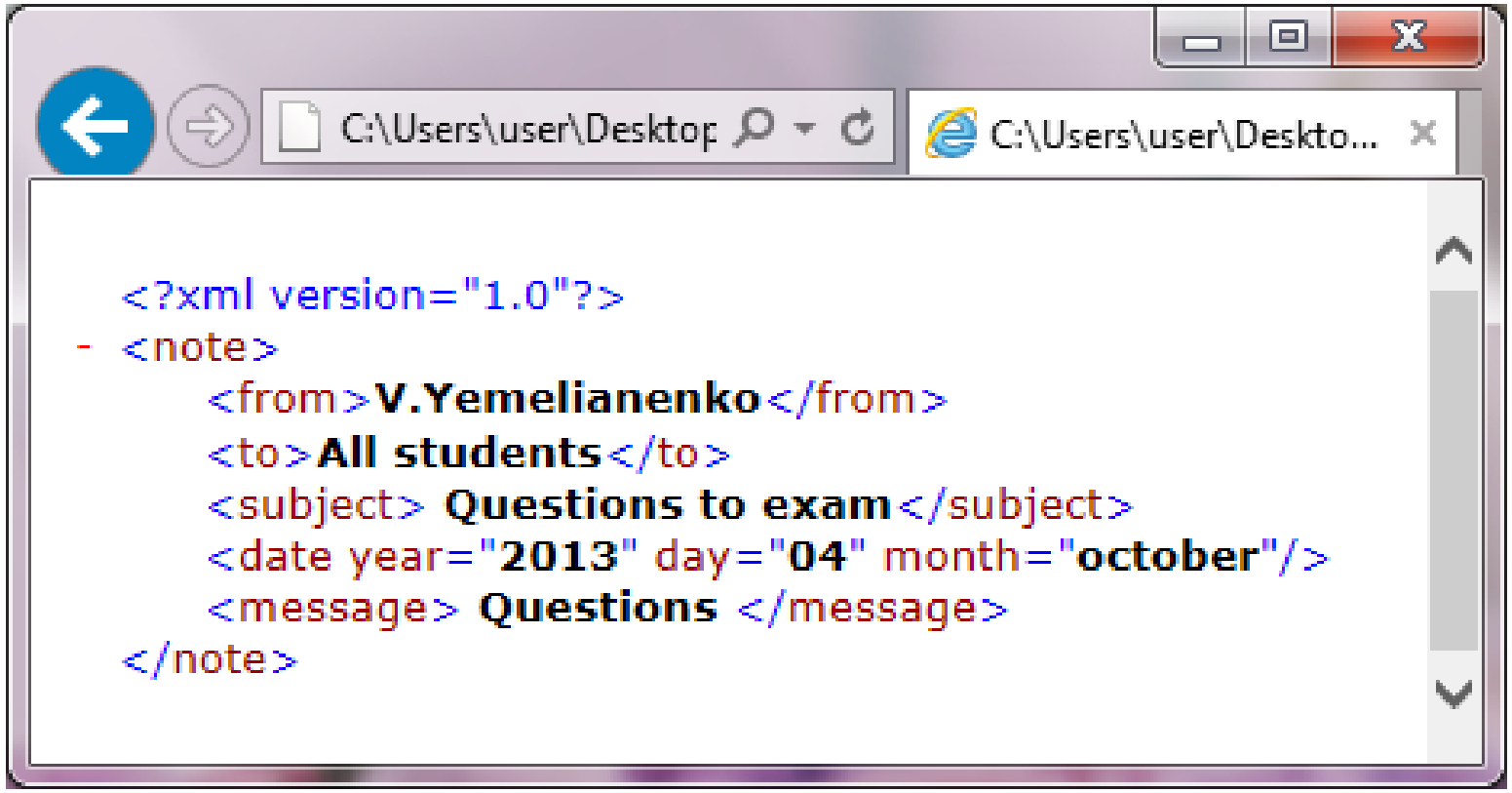
HTML

```
<html>
<h1> note </h1>
<p>From: V.Yemelianenko </p>
<p>to: All students </p>
<p>Subject: Questions to exam </p>
<p>Date: 04.10.2013</p>
<message> </message>
</html>
```

XML

```
<note>
<from>V.Yemelianenko</from>
<to>All students</to>
<subject> Questions to exam</subject>
<date month="october" day="04" year="2013" />
<message> Questions </message>
</note>
```

Создание и просмотр XML документов

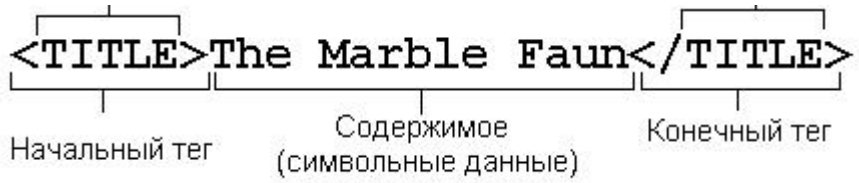


Результат отображения XML-кода в браузере

Правила и синтаксис

Документ должен иметь **только один элемент верхнего уровня** (элемент Документ, или корневой элемент). **Все** другие элементы **должны быть вложены в элемент верхнего уровня.**

Границы элементов представлены **начальным** и **конечным тегами**.
Имя элемента в начальном и конечном тегах элемента должно **совпадать**.



Тег — конструкция разметки, которая содержит **имя элемента.**

Начальный тег: **<element1>**
Конечный тег: **</element1>**

Тег пустого элемента: **<empty_element1 />**

В элементе **атрибуты** могут использоваться только **в начальном теге** и **теге пустого элемента.**

Правила и синтаксис

Элементы должны быть вложены **упорядоченным образом**. Если элемент начинается внутри другого элемента, он должен и заканчиваться внутри этого элемента.

Имена элементов чувствительны к регистру, в котором они набраны. В действительности весь текст внутри XML-разметки является чувствительным к регистру.

```
<TITLE>Leaves of grass<Title> <!-- некорректный элемент -->
```

Корректный документ (well-formed) соответствует всем общим правилам синтаксиса XML, применимым к любому XML-документу: **правильная структура документа, совпадение имен в начальном и конечном теге элемента и т. п.**
Документ, который неправильно построен, не может считаться документом XML.

Правила и синтаксис

Символы разметки

Разметка всегда начинается символом **<** и заканчивается символом **>**. Наряду с символами < и >, специальную роль для разметки играет также символ &.

Символ	Замена
<	<
>	>
&	&amp;

Символ	Замена
'	&apos;
"	&quot;

Правила и синтаксис

Пример кулинарного рецепта, размеченного с помощью XML:

```
<recipe name="хлеб" preptime="5" cooktime="180">
  <title>Простой хлеб</title>
  <composition>
    <ingredient amount="3" unit="стакан">Мука</ingredient>
    <ingredient amount="0.25" unit="грамм">Дрожжи</ingredient>
    <ingredient amount="1.5" unit="стакан">Тёплая вода</ingredient>
    <ingredient amount="1" unit="чайная ложка">Соль</ingredient>
  </composition>
  <instructions>
    <step>Смешать все ингредиенты и тщательно замесить.</step>
    <step>Закрыть тканью и оставить на один час в тёплом помещении.</step>
    <!-- <step>Почитать вчерашнюю газету.</step> - это сомнительный шаг... -->
    <step>Замесить ещё раз, положить на противень и поставить в духовку.</step>
  </instructions>
</recipe>
```

Правила и синтаксис

Правила и синтаксис

Объявление XML

Хотя парсер понимает, что отображаемый документ XML-документом хорошим тоном считается указывать, что это документ XML.

```
<?xml version="1.0"?>
```

Кроме версии XML, объявление может также содержать информацию о кодировке документа.

```
<?xml version="1.1" encoding='UTF-8' ?>
или
<?xml version="1.0" encoding="windows-1251"?>
```

Правила и синтаксис

Комментарии в XML

- Комментарии не относятся к символьным данным документа.
- Комментарий начинается последовательностью «<!--» и заканчивается последовательностью «-->», внутри не может встречаться комбинация символов «--».
- Символ & не используется внутри комментария в качестве разметки.

```
<!-- это комментарий -->
```

Правила и синтаксис

Определение имен

- **Имя элемента** должно начинаться **с буквы, знака подчеркивания (_) или двоеточия (:).**
- **После первого символа** в имени элемента могут быть **буквы, цифры, знаки переноса (-), знаки подчеркивания (_), точка или двоеточие (:).**
- Имена элементов не могут начинаться с букв XML или вариаций на эту тему, поскольку все подобные имена защищены правами на интеллектуальную собственность консорциума W3C.

Совет. Когда Вы присваиваете имена в XML-документе, старайтесь делать их по возможности наиболее информативными. Одним из преимуществ XML-документа является то, что каждому фрагменту информации может быть присвоено информативное описание.

Правила и синтаксис

Атрибуты

<имя_элемента имя_атрибута=“значение”> Содержимое элемента </имя_элемента>

```
<account type= "checking" currency= "Gryvnja">
    <name>Івченко</name>
    <balance>18623,12</balance>
</account>
```

Все значения атрибутов должны быть обязательно в кавычках.
При отсутствии хотя бы одной из кавычек, парсер выдает такое замечание:

This page contains the following errors:

error on line 2 at column 14: attributes construct error

Below is a rendering of the page up to the first error.

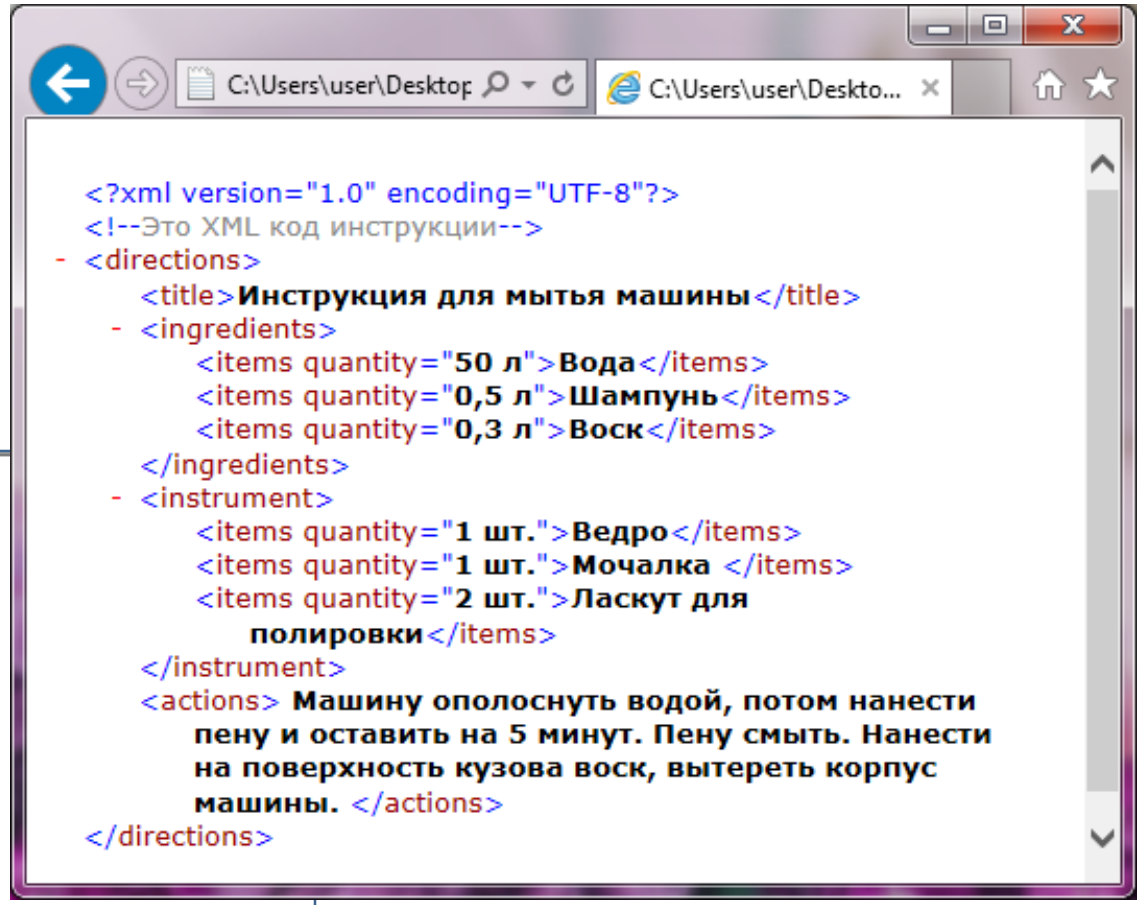
Правила и синтаксис

Пустые элементы

```
<date month="September" day="19" year="2009" />
```

Правила и синтаксис

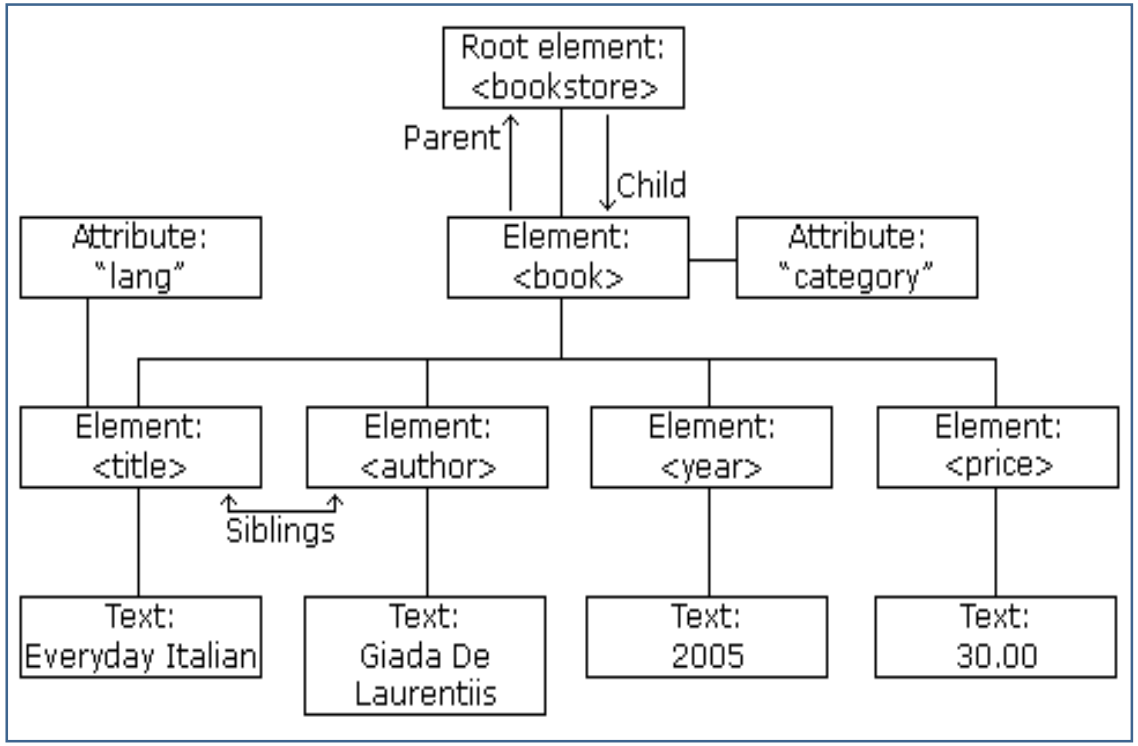
```
<?xml version= "1.0" encoding= "windows-1251">
<!--Это XML код инструкции-->
<directions>
<title>Инструкция для мытья машины</title>
<ingredients>
    <items quantity= "50 л">Вода</items>
    <items quantity= "0,5 л">Шампунь</items>
    <items quantity= "0,3 л">Воск</items>
</ingredients>
<instrument>
<items quantity= "1 шт.">Ведро</items>
<items quantity= "1 шт." >Мочалка </items>
<items quantity= "2 шт." >Ласкут для полировки</items>
</instrument>
<actions> Машину ополоснуть водой, потом нанести пену и оставить на 5 минут.
Пену смыть. Нанести на поверхность кузова воск, вытереть корпус машины.
</actions>
</directions>
```



Структура XML документа

eXtensible Markup Language

```
<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">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```



Связь с другими технологиями

CSS можно использовать для форматирования XML документов.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy -->
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
    <CD>
        <TITLE>Empire Burlesque</TITLE>
        <ARTIST>Bob Dylan</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Columbia</COMPANY>
        <PRICE>10.90</PRICE>
        <YEAR>1985</YEAR>
    </CD>
    <CD>
        <TITLE>Hide your heart</TITLE>
        <ARTIST>Bonnie Tyler</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>CBS Records</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1988</YEAR>
    </CD>
    <CD>
        <TITLE>Greatest Hits</TITLE>
        <ARTIST>Dolly Parton</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>RCA</COMPANY>
```

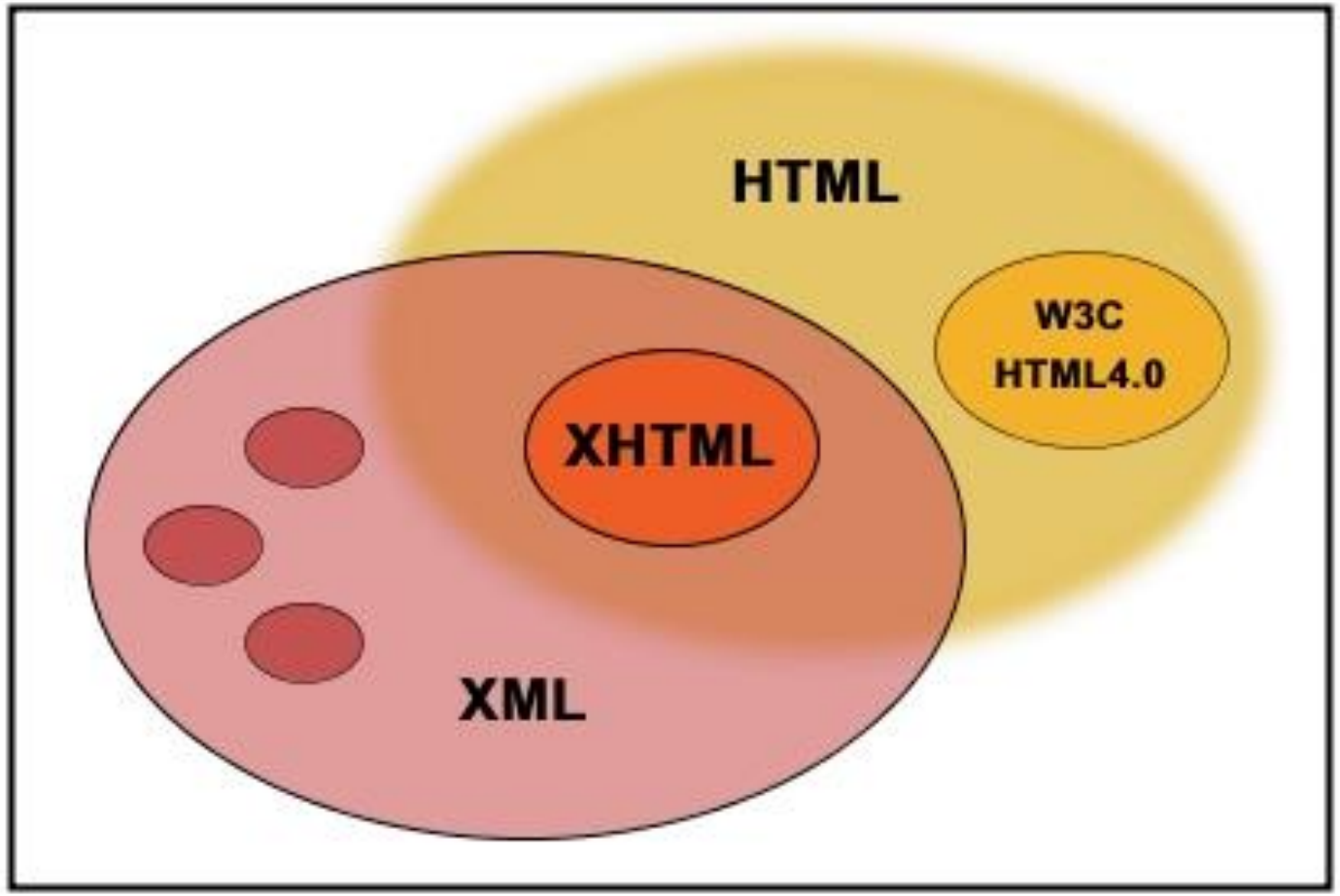
```
CATALOG
{
background-color: #ffffff;
width: 100%;
}
CD
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
TITLE
{
color: #FF0000;
font-size: 20pt;
}
ARTIST
{
color: #0000FF;
font-size: 20pt;
}
COUNTRY,PRICE,YEAR,COMPANY
{
display: block;
color: #000000;
margin-left: 20pt;
}
```

Empire Burlesque Bob Dylan
USA
Columbia
10.90
1985

Hide your heart Bonnie Tyler
UK
CBS Records
9.90
1988

Greatest Hits Dolly Parton
USA
RCA
9.90
1982

Связь с другими технологиями



Применение XML

XML может применяться для решения нескольких задач:

- Обмен данными между разными программами.
- Обмен данными с удаленными филиалами организации.
- Обмен данными между разными организациями.
- Обмен данными между базой данных и Интернет-приложением.

XML Schema

Понятие и особенности XML Schema

XML Schema
- язык описания структуры XML-документа

- XML Schema была задумана **для определения правил, которым должен подчиняться документ.**
- В отличие от других языков, XML Schema была разработана так, чтобы её можно было **использовать в создании программного обеспечения для обработки документов XML**.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Понятие и особенности XML Schema

После проверки документа на соответствие XML Schema читающая программа может создать **модель данных документа**, которая включает:

- **словарь** (названия элементов и атрибутов);
- **модель содержания** (отношения между элементами и атрибутами и их структура);
- **типы данных**.

- **Каждый элемент** в этой модели **ассоциируется с определённым типом данных**, позволяя строить в памяти объект, соответствующий структуре XML-документа.
- Языкам объектно-ориентированного программирования гораздо легче иметь дело с таким объектом, чем с текстовым файлом.

Понятие и особенности XML Schema

- **Один словарь может ссылаться на другой**, и, таким образом, разработчик может использовать уже существующие словари и легче устанавливать и распространять стандарты XML структуры для определённых задач (например, словарь протокола SOAP).

Файл, содержащий XML Schema,
обычно имеет расширение «.xsd» (XML Schema definition).

Понятие и особенности XML Schema

Пример схемы на XML Schema, расположенной в файле "country.xsd" и описывающей данные о населении страны:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="country_name" type="xs:string"/>
        <xs:element name="population" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Пример документа, соответствующего этой схеме:

```
<?xml version="1.0" encoding="utf-8"?>
<country>
    <country_name>France</country_name>
    <population>59.7</population>
</country>
```

Понятие и особенности XML Schema

XML Schema

- определяет элементы и атрибуты, которые могут быть в документе
- определяет, какие элементы являются дочерними
- определяет порядок и количество дочерних элементов
- определяет, является ли элемент пустым или может включать в себя текст
- определяет тип данных элементов и атрибутов
- определяет фиксированные значения и значения по умолчанию для элементов и атрибутов

Понятие и особенности XML Schema

Элемент <schema> является корневым элементом каждой XML Schema:

```
<?xml version="1.0"?>

<xs:schema>
...
...
</xs:schema>
```

Элемент <schema> может содержать атрибуты:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    ...
</xs:schema>
```

Понятие и особенности XML Schema

В схемах XSD дескрипторы, используемые в документах XML, разделяются на две категории:

Сложные типы. Элементы сложных типов могут содержать другие элементы, а также обладают определенными атрибутами, т.е. иметь смешанный содержание.

```
<note    time= '08:15:00'
         date= '15-10-2013'>
         <text> Завтра на 8.15 в университет </text>
</note>
```

Простые типы.

```
<text>Завтра на 8.15 в университет</text>
```

Простые и сложные типы элементов
- это уникальные характеристики языка XML Schema.

Понятие и особенности XML Schema

Описание элементов в XML Schema:

Простые элементы

```
<lastname>Refsnes</lastname>
<age>36</age>
<dateborn>1970-03-27</dateborn>
```

```
<xs:element name="lastname" type="xs:string"/>
<xs:element name="age" type="xs:integer"/>
<xs:element name="dateborn" type="xs:date"/>
```

Префикс **xs** означает, что **это базовый тип**, который уже **описан в стандарте XML Schema.**

Понятие и особенности XML Schema

Описание элементов в XML Schema:

Сложные элементы

```
<employee>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</employee>
```

```
<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Понятие и особенности XML Schema

Основные типы данных

- **xs:string**
- **xs:decimal**
- **xs:integer**
- **xs:boolean**
- **xs:date**
- **xs:time**

Понятие и особенности XML Schema

Описание атрибутов

```
<lastname lang="EN">Smith</lastname>
```

```
<xs:attribute name="lang" type="xs:string"/>
```

Некоторые свойства атрибутов

```
<xs:attribute name="lang" type="xs:string" default="EN"/>
```

```
<xs:attribute name="lang" type="xs:string" fixed="EN"/>
```

SVG

SVG - **Scalable Vector Graphics**

язык разметки масштабируемой векторной графики и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML.

What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define vector-based graphics for the Web
- SVG defines the graphics in XML format
- SVG graphics do NOT lose any quality if they are zoomed or resized
- Every element and every attribute in SVG files can be animated
- SVG is a W3C recommendation
- SVG integrates with other W3C standards such as the DOM and XSL

SVG - **Scalable Vector Graphics**

SVG Advantages

Advantages of using SVG over other image formats (like JPEG and GIF) are:

- SVG images can be created and edited with any text editor
- SVG images can be searched, indexed, scripted, and compressed
- SVG images are scalable
- SVG images can be printed with high quality at any resolution
- SVG images are zoomable (and the image can be zoomed without degradation)
- SVG is an open standard
- SVG files are pure XML

SVG - **Scalable Vector Graphics**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg version = "1.1"
     baseProfile="full"
     xmlns = "http://www.w3.org/2000/svg"
     xmlns:xlink = "http://www.w3.org/1999/xlink"
     xmlns:ev = "http://www.w3.org/2001/xml-events"
     height = "400px"  width = "400px">
     <rect x="0" y="0" width="400" height="400"
          fill="none" stroke="black" stroke-width="5px" stroke-opacity="0.5"/>
     <g fill-opacity="0.6" stroke="black" stroke-width="0.5px">
        <circle cx="200px" cy="200px" r="104px" fill="red"   transform="translate(  0,-52)" />
        <circle cx="200px" cy="200px" r="104px" fill="blue"  transform="translate( 60, 52)" />
        <circle cx="200px" cy="200px" r="104px" fill="green" transform="translate(-60, 52)" />
     </g>
</svg>
```

WSDL

Язык WSDL

WSDL (Web Services Description Language) — язык описания веб-сервисов и доступа к ним, основанный на языке XML.

WSDL **расширяем**, что позволяет описывать услуги (сервисы) и их сообщения независимо от того, какие форматы сообщений или сетевые протоколы используются для транспорта, однако, чаще всего используется WSDL 1.1 вместе с SOAP 1.1, HTTP GET/POST и MIME.

Язык WSDL

WSDL предназначен для точного описания веб-сервиса и его программных интерфейсов. В описании можно найти такую информацию, как **адрес сервера, протокол, номер используемого порта, формат запроса и многое другое.**

Стандартизированное описание упрощает понимание и применение. Допустим, что вы нашли сервис, который решает необходимые вам задачи, и хотите его использовать в своих решениях. Самый простой способ получить информацию о чужой разработке и ее возможностях — взглянуть на WSDL-описание.

Рекомендован консорциумом W3C

Структура WSDL

WSDL документ описывает веб-сервис, с помощью четырех основных элементов:

<types>	**- определение типов данных** - контейнер для определения типов данных, используемых веб-сервисом
<message>	- **элементы данных** - сообщения, используемые web-сервисом
<portType>	- **абстрактные операции** - список операций, которые могут быть выполнены с сообщениями
<binding>	- **связывание сервисов** - способ, которым сообщение будет доставлено

Структура WSDL

```
<definitions>

<types>
  data type definitions........
</types>

<message>
  definition of the data being communicated...
</message>

<portType>
  set of operations......
</portType>

<binding>
  protocol and data format specification....
</binding>

</definitions>
```

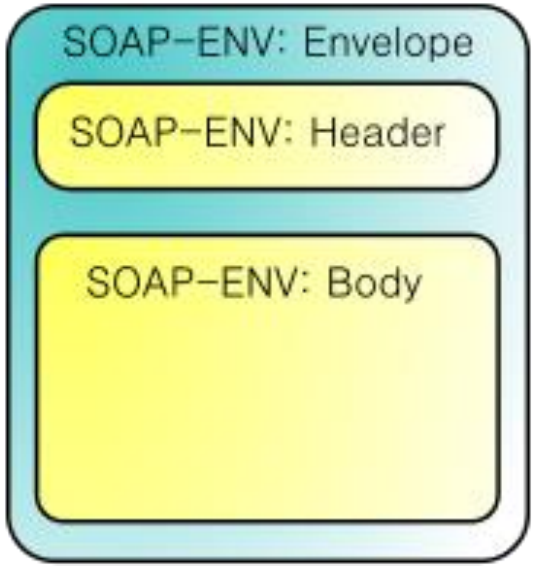
Основная структура документа WSDL выглядит следующим образом:

Тег <definitions> – это корневой тег всех WSDL-документов

SOAP

Протокол SOAP

SOAP
(Simple Object Access Protocol
— простой протокол доступа к объектам;
вплоть до спецификации 1.2)
— протокол обмена структурированными сообщениями в распределённой вычислительной среде.



- Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур (RPC).
- Сейчас протокол используется для обмена произвольными сообщениями в формате XML, а не только для вызова процедур.
- SOAP является расширением протокола XML-RPC.

SOAP - протокол для доступа к веб-сервисам.

Протокол SOAP

- SOAP может использоваться с любым протоколом прикладного уровня: SMTP, FTP, HTTP, HTTPS и др. Однако его взаимодействие с каждым из этих протоколов имеет свои особенности, которые должны быть определены отдельно.
- Чаще всего SOAP используется поверх HTTP.

- SOAP является одним из стандартов, на которых базируются технологии веб-служб.

Использование **SOAP для передачи сообщений увеличивает их объём и снижает скорость обработки**. В системах, где скорость важна, чаще используется пересылка XML-документов через HTTP напрямую, где параметры запроса передаются как обычные HTTP-параметры.

Хотя **SOAP** является стандартом, некоторые программы часто генерируют сообщения в несовместимом формате.

Протокол SOAP

SOAP

- Предназначен для связи между приложениями
- Является форматом для отправки сообщений
- Не зависит от языков и платформ
- Основан на XML
- Простой и расширяемый
- Позволяет обойти брандмауэры
- Рекомендован W3C

Основные понятия SOAP

Спецификация SOAP определяет

- XML-«конверт» для передачи сообщений,
- метод для кодирования программных структур данных в формате XML,
- а также средства связи по протоколу HTTP.

SOAP-сообщения бывают двух типов:

- **запрос (Request)** - вызывает метод удаленного объекта;
- **ответ (Response)** - возвращает результат выполнения данного метода.

SOAP предоставляет возможность обмена данными между приложениями, работающими на различных операционных системах, с помощью различных технологий и языков программирования.

Основные понятия SOAP

SOAP-сообщение состоит из трех основных элементов: конверт, заголовок и тело.

Конверт <soap:Envelope>	Является обязательной составляющей, в него вложены остальные элементы.
Заголовок <soap:Header>	Если он есть, обязательно является первым элементом, вложенным в конверт. Заголовок предоставляет информацию о самом сообщении, как правило, эта информация адресуется тем узлам, через которые проходит сообщение (SOAP-сообщение может пройти через насколько SOAP-серверов или несколько приложений на одном сервере).
Тело сообщения <soap:Body>	Является необязательным, но, если оно есть, оно следует непосредственно за заголовком. В элемент <Body> могут быть вложены любые элементы, определяющие информацию сообщения, которая адресуется приложению, выполняющему функцию Web-сервиса.

Основные понятия SOAP

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
...
</soap:Header>

<soap:Body>
...
  <soap:Fault>
  ...
  </soap:Fault>
</soap:Body>

</soap:Envelope>
```

Структура SOAP сообщения

Основные понятия SOAP

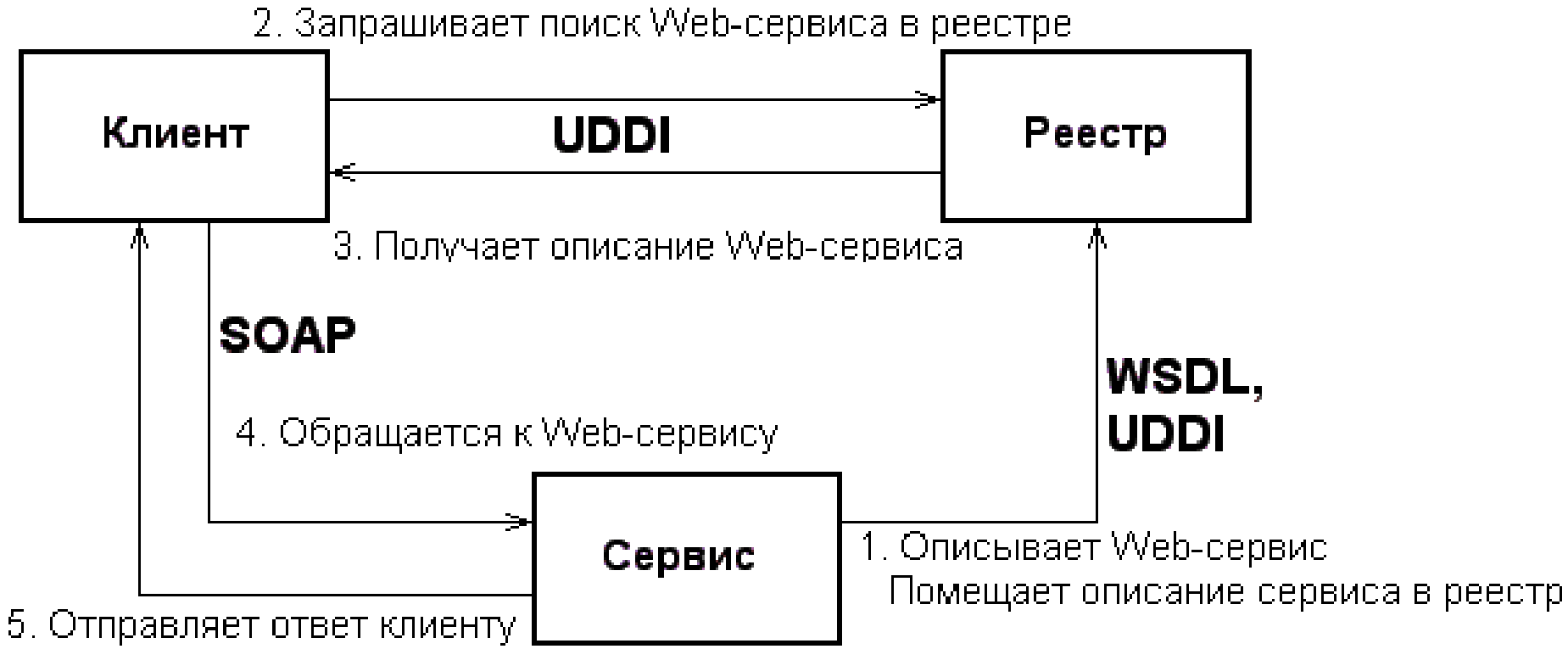
Пример SOAP-запроса на сервер интернет-магазина:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <getProductDetails xmlns="http://warehouse.example.com/ws">
         <productID>12345</productID>
      </getProductDetails>
   </soap:Body>
</soap:Envelope>
```

Пример ответа

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
         <getProductDetailsResult>
            <productID>12345</productID>
            <productName>Стакан граненый</productName>
            <description>Стакан граненый. 250 мл.</description>
            <price>9.95</price>
            <currency>
                <code>840</code>
                <alpha3>USD</alpha3>
                ....
            </currency>
            <inStock>true</inStock>
         </getProductDetailsResult>
      </getProductDetailsResponse>
   </soap:Body>
</soap:Envelope>
```

Взаимодействие технологий



UDDI — Universal Description, Discovery and Integration
Задача UDDI — предоставить механизм для обнаружения веб-сервисов. UDDI задает бизнес-реестр, в котором провайдеры веб-сервисов могут регистрировать сервисы, а разработчики — искать необходимые им сервисы.

Дополнительные материалы

- http://www.w3.org/
- http://www.w3.org/TR/wsdl
- http://www.w3.org/2002/07/soap-translation/russian/part0.html
- http://www.w3schools.com/
- http://khpi-iip.mipk.kharkiv.edu/library/extent/ii/itci/04.html
- https://inkscape.org/en/
- www.google.com ☺

… и другие учебники XML.

Q&A

Thank You

Focused forward

Netcracker