Web технологии 3

Схема документа на языке XSD
(XML Schema Defenition Language)

- В основе корректный (well formed) XML
- Префикс по соглашению xsd
- Структура

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="
http://www.w3.org/2001/XMLSchema">
<!-- описание схемы документа -->
</xsd:schema>
```

- Простой элемент xml — не содержит атрибутов и вложенных элементов
- Сложный элемент xml — содержит атрибуты или элементы

Типы данных

Вещественные числа

- decimal — простое вещественное число ('-1.23', '12678967.543233', '+100000.00', '210')
- float — 32битное вещественное число
- double — 64битное вещественное число

Целые числа

- byte/unsignedInt/short/long/unsignedLong/...
- integer — целое число
- positiveInteger/negativeInteger — положительные/отрицательные

Строки

- string — произвольная строка, в т.ч. пустая
- token — строка без пробелов
- language — Код языка, скажем, en-US.
- ID — идентификатор (без пробелов)
- name — имя элемента XML

Время и дата

- time — время «hh:mm:ss zone»
  13:27:45
- date — дата «yyyy-mm-dd zone»
  0107-04-17 z-02:00
- dateTime — дата «yyyy-mm-ddThh:mm:ss zone»
- duration — длительность «P*Y*M*DT*H*M*S»
  P1Y2M3DT10H30M45S

Двоичный тип

- binary - двоичные числа
  00101110
- hexBinary — 16-ричные числа
- base64Binary — base64 кодированные данные

Разное

- boolean — Значение true или false. Допускаются 1 или 0.
- anyURI — URI адрес

Определение элементов

простой элемент (простой тип)

```
<xsd:element name='age' type='xsd:integer'/>
<xsd:element name='price' type='xsd:decimal'/>
```

Пример:

```
<price>45.50</price>
```

элемент с атрибутом (сложный тип)

```
<xsd:element name='price'>
  <xsd:complexType base='xsd:decimal'>
    <xsd:attribute name='currency' type='xsd:string'/>
  </xsd:complexType>
</xsd:element>
```

Пример:

```
<price currency='US'>45.50</price>
```

Элемент с вложенными элементами

- XML документ

```
<Book>
   <Title>Cool XML<Title>
   <Author>Cool Guy</Author>
</Book>
```

- DTD

```
<!ELEMENT Book (Title, Author)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
```

- Схема

```
<xsd:element name='Book' type='BookType'/>
<xsd:complexType name='BookType'>
    <xsd:element name='Title' type='xsd:string'/>
    <xsd:element name='Author' type='xsd:string'/>
</xsd:complexType>
```

Ссылки на другие элементы

```
<xsd:element name="note">
  <xsd:complexType>
      <xsd:element ref="to"/>
      <xsd:element ref="from"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="to" type="xsd:string"/>
<xsd:element name="from" type="xsd:string"/>
```

Грани

Ограничения на содержимое XML-элементов называются гранями

Ограничения на значение элемента (restriction)

```
<xsd:element name="age">
<xsd:simpleType>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="16"/>
    <xsd:maxInclusive value="34"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

simpleType — определение простого типа

Варианты граней

- enumeration Задает список значений
- length Задает длину
- maxLength Задает максимальную длину
- minLength Задает минимальную длину
- maxExclusive Задает максимальное значение
- maxInclusive Задает максимальное значение включительно
- minExclusive Задает минимальное значение
- minInclusive Задает минимальное значение включительно
- fractionDigits Задает число цифр в дроби
- totalDigits Задает число цифр
- pattern Задает шаблон содержимого элементов
- whiteSpace Задает значение пробелов в содержимом элементов

Ограничения на значения из списка (list)

```
<xsd:element name="car">
<xsd:simpleType>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Audi"/>
    <xsd:enumeration value="Mercedes"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

или

```
<xsd:simpleType name="carType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Audi"/>
    <xsd:enumeration value="Mercedes"/>
  </xsd:restriction>
</xsd:simpleType>
```

Объединение типов элементов (union)

```
<xsd:attribute name="size">
  <xsd:simpleType>
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base="xsd:positiveInteger">
          <xsd:minInclusive value="8"/>
        </xsd:restriction>
      </xsd:simpleType>

      <xsd:simpleType base="xsd:token">
         <xsd:enumeration value="small"/>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
</xsd:attribute>
```

Определение сложных типов

```
<xsd:complexType name="имя">
…
</xsd:complexType>
```

Пустой элемент

```
<xsd:complexType name="imageType">
  <xsd:attribute name="href" type="xsd:anyURI">
</xsd:complexType>

<xsd:element name="image" type="imageType"/>
```

в XML

```
<image href="http://some.com/images/image.png"/>
```

Элемент с простым телом

```
<xsd:complexType name='priceType'>
  <xsd:simpleContent>
    <xsd:extension base='xsd:decimal'>
    <xsd:attribute name='currency' type='xsd:string'/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:element name='price' type='priceType' />
```

- extension Определяет расширения элемента
- simpleContentОпределяет контентную модель типа, который может содержать только символьные данные

Модель группы (порядок появления элементов)

1. последовательность (sequience)

```
<xsd:element name="note">
  <xsd:complexType>
    <xsd:sequence>
     <xsd:element name="to" type="xsd:string"/>
     <xsd:element name="from" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Аналог "," из DTD

Правильно: <note><to/><from/></note>

Неправильно: <note><from/><to/></note>

2. Выбор (choice)

```
<xsd:element name="note">
  <xsd:complexType>
    <xsd:choice>
     <xsd:element name="to" type="xsd:string"/>
     <xsd:element name="from" type="xsd:string"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Аналог '|' из DTD

Правильно: <note><to/></note>

Неправильно: <note><from/><to/></note>

3. Произвольный порядок (all)

```
<xsd:element name="note">
  <xsd:complexType>
     <xsd:all>
      <xsd:element name="to" type="xsd:string"/>
      <xsd:element name="from" type="xsd:string"/>
     </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Правильно: <note><to/><from/></note>
<note><from/><to/></note>

Неправильно: <note><from/><from/><to/></note>

Сложное тело сложного типа

```
<xsd:element name="employee" type="fullpersoninfo"/>

<xsd:complexType name="fullpersoninfo">
  <xsd:complexContent>
    <xsd:extension base="personinfo">
      <xsd:sequence>
        <xsd:element name="address" type="xsd:string"/>
        <xsd:element name="city" type="xsd:string"/>
        <xsd:element name="country" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="personinfo">
  <xsd:sequence>
    <xsd:element name="firstname" type="xsd:string"/>
    <xsd:element name="lastname" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

в XML

```
<employee>
  <firstname>Иван/firstname>
  <lastname>Петров</lastname>
  <address>Мерецкова, 76</address>
  <city>Петрозаводск</city>
  <country>Россия</country>
</employee>
```

Группировка элементов (group)

```
<xsd:group name="custGroup">
  <xsd:sequence>
    <xsd:element name="customer" type="xsd:string"/>
    <xsd:element name="orderdetails" type="xsd:string"/>
    <xsd:element name="billto" type="xsd:string"/>
    <xsd:element name="shipto" type="xsd:string"/>
  </xsd:sequence>
</xsd:group>

<xsd:element name="order" type="ordertype"/>

<xsd:complexType name="ordertype">
  <xsd:group ref="custGroup"/>
  <xsd:attribute name="status" type="xsd:string"/>
</xsd:complexType>
```

Произвольное содержимое (any)

```
<xsd:element name="person">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="firstname" type="xsd:string"/>
      <xsd:any minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Пример:

```
<employee>
  <firstname>Иван/firstname>
  <lastname>Петров</lastname>
  <address>Мерецкова, 76</address>
</employee>
```

Общие атрибуты определений

- maxOccurs Задает максимальное количество вхождений элемента
- minOccurs Задает минимальное количество вхождений элемента
- default Задает значение элемента или атрибута по умолчанию
- fixed Задает фиксированное значение элемента или атрибута
- ref Задает ссылку на глобально определённый элемент
- type Задает тип элемента