

Web-технологии 3

Введение в Extensible Markup Language
(XML)

Литература

- W3C: XML Technology. <http://www.w3.org/standards/xml/>
- Эллиот Гарольд и У. Скотт Минс "XML. Справочник"
- Валиков А.Н. "Технология XSLT"
- Хабибуллин И. "Самоучитель XML"
- Эрик Рэй "Изучаем XML"
- Курс "Основы XML" <http://www.intuit.ru/department/internet/xml/>
- Anders Møller and Michael Schwartzbach "XML Tutorial"
- А. С. Деревянко "Язык запросов XQuery"
- О.Н. Кищенко "Языки информационного обмена"

История

- Работа с текстом:
 1. Специальный редактор (Adobe, MS Office)
 2. Текст + разметка (html, xml, TeX)
- Обобщенный структурированный язык разметки (Structured Generalized Language — SGML, 1983-86гг.)
- SGML – описание набора элементов и структуры документа
- Набор наиболее употребительных элементов – SGML приложение

- Начало 90х - Разработка HTML как компактного варианта SGML
- В 1996 г. группа XML Working Group разработала ветвь языка SGML - Extensible Markup Language (XML)
- Использование XML:
 - виртуальное представление документов любого типа;
 - сортировка, фильтрация, упорядочение, поиск и манипулирование информацией иными способами;
 - представление информации в структурированном виде.

Официальные концептуальные цели XML

1. XML должен стать языком прямого использования в Internet.
2. XML будет поддерживать большое количество приложений.
3. XML будет совместим с SGML.
4. Будет легче писать программы, обрабатывающие XML-документы.
5. Количество дополнительных функций в XML должно быть минимальным, а в идеале — нулевым.
6. XML-документы должны быть понятными и ясными для пользователя.
7. Разработка XML должна быть завершена достаточно быстро.
8. Язык XML должен быть формальным и кратким.
9. XML-документы будет проще создавать.
10. Сжатая форма при XML-разметке не важна.

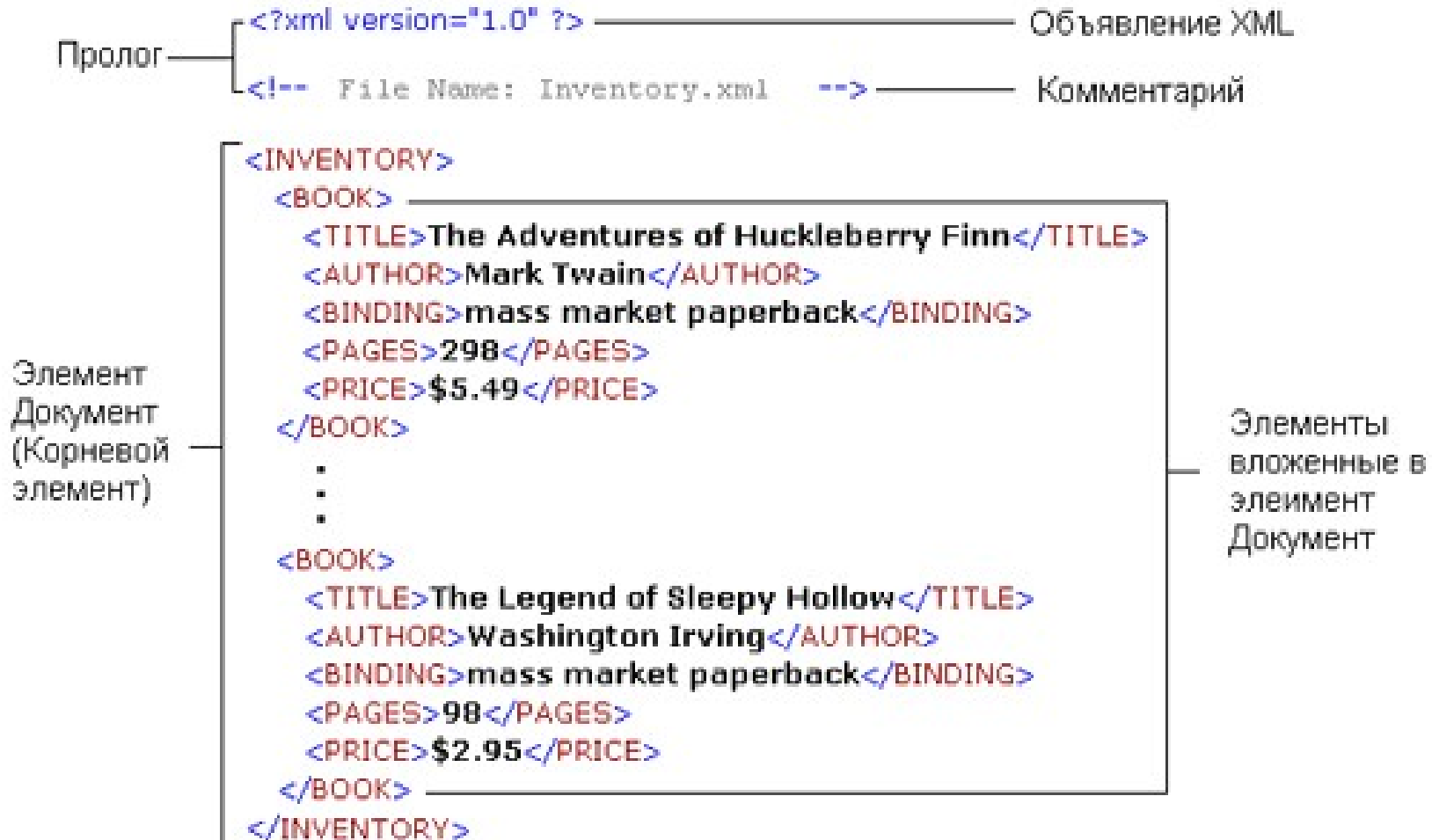
Использование XML

- Работа с БД
 - Организация данных
 - Поиск, сортировка, фильтрация
- Структурирование документов
 - Иерархическая структура
- Векторная графика
 - VML - Vector Markup Language
- Мультимедиа презентации
 - HTML + TIME - HTML Timed Interactive Multimedia Extensions
 - SMIL - Synchronized Multimedia Integration Language

- Описание каналов
 - CDF - Channel Definition Format
- Взаимодействие приложений через Web
 - SOAP - Simple Object Access Protocol
- Математические формулы в Web
 - MathML - Mathematical Markup Language
- Музыкальные партитуры
 - MusicML - Music Markup Language

Структура документа XML

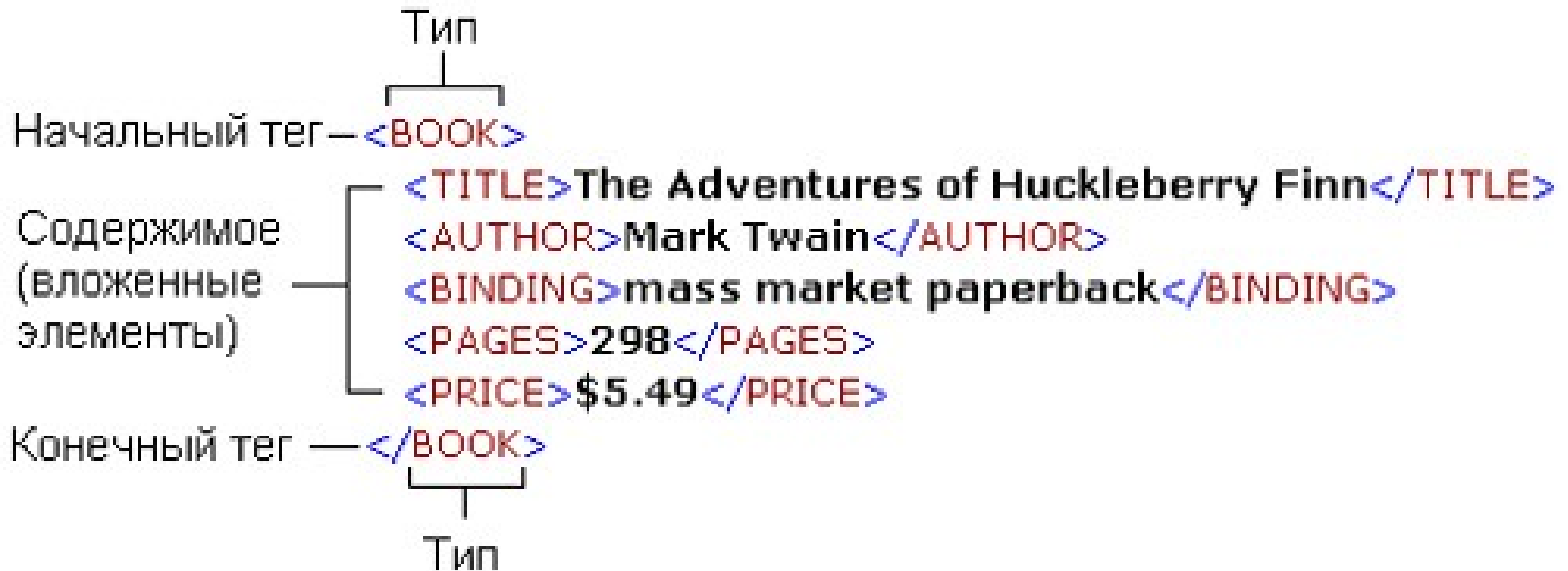
- Описание правил
- Не содержит набора тегов



Пролог

- Определение что это XML документ
- Не является обязательным, но спецификация требует включения
 - `version` – версия формата XML
 - `encoding` – кодировка
 - `standalone` – автономность
- Комментарии
- Объявление типа документа (DTD)
- Инструкции по обработке:
 - `<?ScriptA Category="books"Style="formal"?>`

Тело документа



Атрибут

- определение в начальном теге
- имя = "значение"
- Правила оформления имени:
 - начинается с буквы или с "_"
 - может содержать буквы, цифры, ".", "-" и "_"
 - префикс "xml" зарезервирован
 - имя может присутствовать только один раз

- Правила оформления значения
 - строка заключена в ' ' или " "
 - внутри строки не может быть той же кавычки
 - может содержать ссылку на символ или примитив
 - не может содержать символ <
 - не может содержать символ & если это не ссылка

Данные

- любые символы за исключением <, & и "[>"
- содержимое элемента CDATA:
 - <![CDATA[...]]>
- КОДЫ СИМВОЛОВ:
 - < = <; > = >
- Пустой тег:
 - <hr></hr> или <hr/>

Базовые правила

- только 1 элемент верхнего уровня
- вложение упорядоченным образом (начало и конец в одном элементе)
- наличие начального и конечного тега
- имя типа в начальном теге должно в точности соответствовать имени типа в конечном теге
- чувствительность имени к регистру
- значения атрибутов в кавычках
- информация м/у тегами – данные элемента, в т.ч. пробелы

Имя тега

- Начинается с буквы или "_"
- состоит из символов Unicode, цифр, "_", "-", ".", " "
- Префикс "xml" зарезервирован
- ":" - разделитель пространства имен

Пространство имен

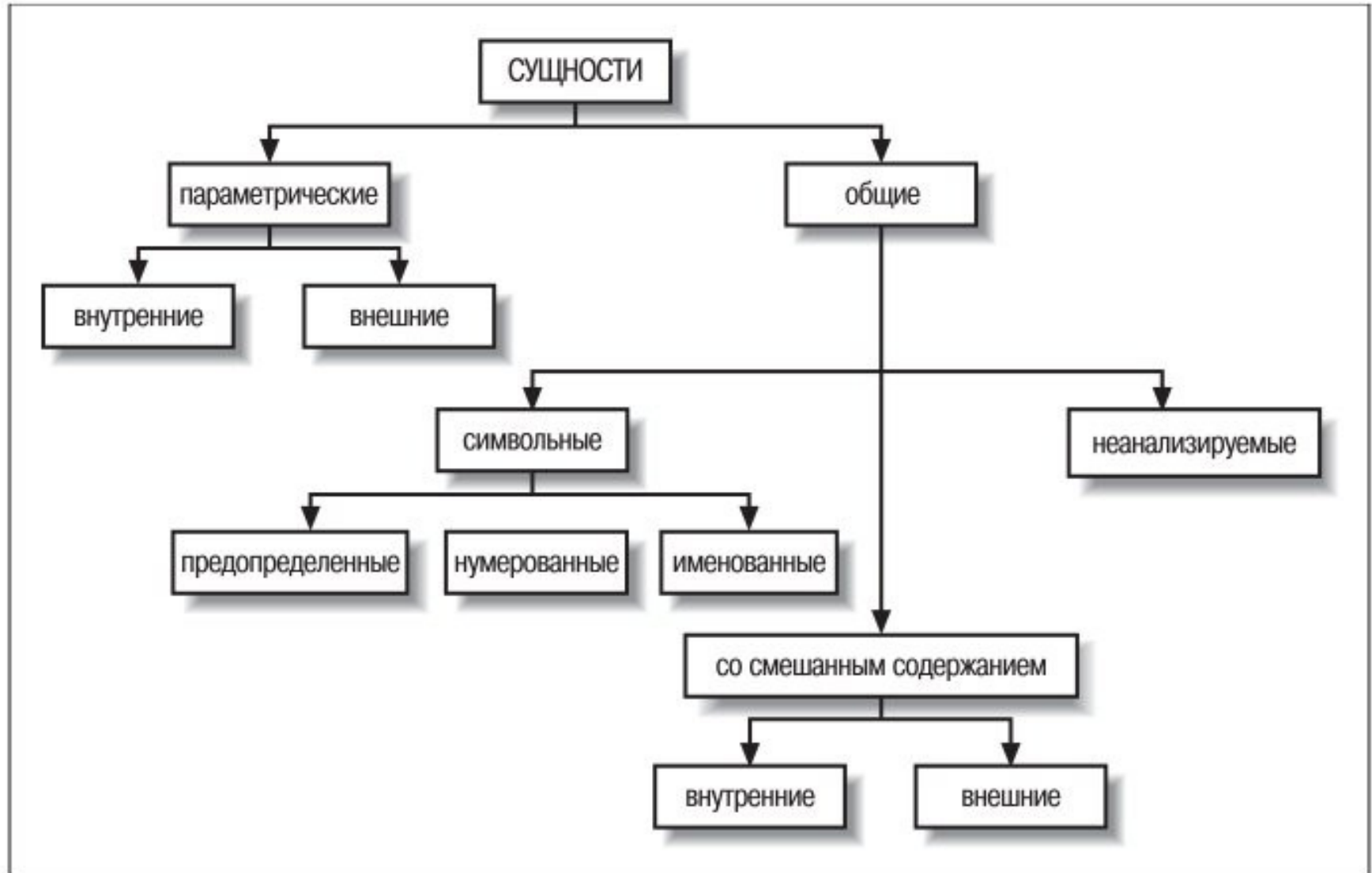
- Логическое разделение объектов на области в XML
- Предотвращение конфликтов имен
- объявление способа обработки частей XML
- объявление пространства:
 - `<node xmlns:prfx=uri> ... </node>`
 - все вложенные элементы входят в пространство
- Использование:
 - `<prfx:date> ... </prfx:date>`
- атрибуты и элементы по умолчанию без префикса
- пространство по умолчанию:
 - `<journal xmlns=http://www.psycholabs.org/mynamespace/>`

Зарезервированные имена атрибутов

- `xml:lang` – определение языка элемента
- `xml:space` – принадлежность пробельных символов к элементу (по умолчанию `preserve` / `default` – без сохранения)
- `xml:link` – элемент-ссылка
- `xml:attribute` – переназначение атрибута для XLink

СУЩНОСТИ

- один раз объявить, много раз использовать



- Состоит из имени и значения
- определение сущностей в объявлениях
- подстановка значений вместо ссылок
- продолжение разбора с точки перед ссылкой
- синтаксис ссылок:
 - &имя; - общая сущность
 - %имя; - параметрическая сущность

Пример

```
<?xml version="1.0"?>
<!DOCTYPE message SYSTEM "/xmlstuff/dtds/message.dtd"
[
<!ENTITY client "Mr. Ivanov">
<!ENTITY agent "Ms. Sally Tashuns">
<!ENTITY phone "<number>617 555 1299</number>">
]>
<message>
<opening>Dear &client;</opening>
<body> To reserve a place for your holiday, call &agent; at &phone;.
Hurry, &client;,. Time is running out!</body>
</message>
```

- Сущность ñ – нумерованная символная сущность (определены неявно)
- Предопределенные символные сущности:
 - & &
 - ' '
 - > >
 - < <
 - " ''

Сущности со смешанным содержанием

- Пример внешних сущностей

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

```
<!DOCTYPE longdoc SYSTEM "http://www.dtdsrus.com/generic.dtd"
```

```
[
```

```
<!ENTITY part1 SYSTEM "p1.xml">
```

```
<!ENTITY part2 SYSTEM "p2.xml">
```

```
<!ENTITY part3 SYSTEM "p3.xml">
```

```
]>
```

```
<longdoc>
```

```
&part1;
```

```
&part2;
```

```
&part3;
```

```
</longdoc>
```

Сравнение XML и JSON

- JavaScript Object Notation (JSON) – формат обмена данными
- Основан на подмножестве языка JavaScript
- Структуры данных: объект (коллекция пар) и массив (упорядоченный список)
- Поддержка вложенности структур данных

Пример

- JSON

```
{  
  "firstName": "Иван",  
  "lastName": "Иванов",  
  "address": {  
    "streetAddress": "Московское ш., 101,  
кв.101",  
    "city": "Ленинград",  
    "postalCode": 101101  
  },  
  "phoneNumbers": [  
    "812 123-1234",  
    "916 123-4567"  
  ]  
}
```

- XML

```
<person firstName="Иван"  
lastName="Иванов">  
  <address streetAddress="Московское  
ш., 101, кв.101" city="Ленинград"  
postalCode="101101" />  
  <phoneNumbers>  
    <phoneNumber>812 123-  
1234</phoneNumber>  
    <phoneNumber>916 123  
4567</phoneNumber>  
  </phoneNumbers>  
</person>
```