Темы модуля

- Введение в XML
- Обзор возможностей по работе PHP 5 с технологией XML
  - SAX
  - DOM
  - SimpleXML
  - XMLReader и XMLWriter
- обзор XSL/T
- Преобразование данных на сервере

Введение в XML

- XML (Extensible Markup Language)
  - Расширяемый язык разметки
- Предназначен для:
  - хранения структурированных данных
  - обмена информацией между программами
  - создания на его основе других, более специализированных, языков разметки (OFX, OTP, WSDL, SOAP, VML, XSL, ebXML, CML, XLANG)
- Цель создания:
  - обеспечение совместимости при передаче структурированных данных между разными системами обработки информации

Различия XML и HTML

- HTML описывает ИЗ ЧЕГО СОСТОИТ и КАК отображать документ
  -
    Tumba Yumba
    12345678

- XML определяет ЗНАЧЕНИЕ и ОТНОШЕНИЕ данных
  - <person>
    <name>Tumba Yumba</name>
    <phone>12345678</phone>
    </person>

XML разметка

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- Пример XML разметки -->
<catalog>
  <book id="1">
     <title>XML и IE5</title>
     <author>Алекс Гомер</author>
     <price currency="RUR">200</price>
     <exists />
  </book>
</catalog>
```

- XML декларация
- Комментарий
- Элемент документа (корневой элемент)
- Элемент
- Атрибут
- Текстовые данные

Правила XML

- Если документ содержит символы, выходящие за рамки ASCII, необходимо указать кодировку
- XML чувствителен к регистру символов
- XML-документ состоит из вложенных элементов
- Элемент состоит из открывающего и закрывающего тегов, а также содержимого
- Теги должны быть правильно вложены друг в друга
- Все парные теги должны быть закрыты
- Возможно формирование пустых элементов – без содержимого
- Должен существовать только один корневой элемент, который содержит все остальные элементы. Пустой документ (без корневого элемента) недопустим!
- Элементы могут иметь атрибуты
- Значения атрибутов заключаются в одинарные или двойные кавычки
- У каждого конкретного элемента не должно быть повторяющихся атрибутов

Корректность и валидность XML-документов

- Корректные XML-документы (well-formed)
  - Документы, полностью соответствующие правилам оформления XML.
  - Корректность проверяется XML-парсер.
- Валидные XML-документы (valid)
  - Корректные XML-документы, которые соответствуют заранее определенному набору правил.
  - Валидность проверяется валидатором.
- Описание структуры документа
  - DTD – Document Type Definition (определение типа документа)
  - XML Схемы

Пример DTD

```
<!ELEMENT catalog (book+)>
<!ELEMENT book (author+, title, price,
  exists?)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT exists (#PCDATA)>
<!ATTLIST price currency CDATA #IMPLIED>
```

- ```
  <?xml version="1.0"?>
  <!DOCTYPE catalog SYSTEM "catalog.dtd">
  ```
  ```

Пример простой XML-схемы

```
<xs:schema xmlns:xs=
   "http://www.w3.org/2001/XMLSchema">
 <xs:element name="страна" type="Страна"/>
 <xs:complexType name="Страна">
  <xs:sequence>
   <xs:element name="название" type="xs:string"/>
   <xs:element name="население" type="xs:decimal"/>
  </xs:sequence>
 </xs:complexType>
</xs:schema>

<страна xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="country.xsd">
  <название>Франция</название>
  <население>59.7</население>
</страна>
```

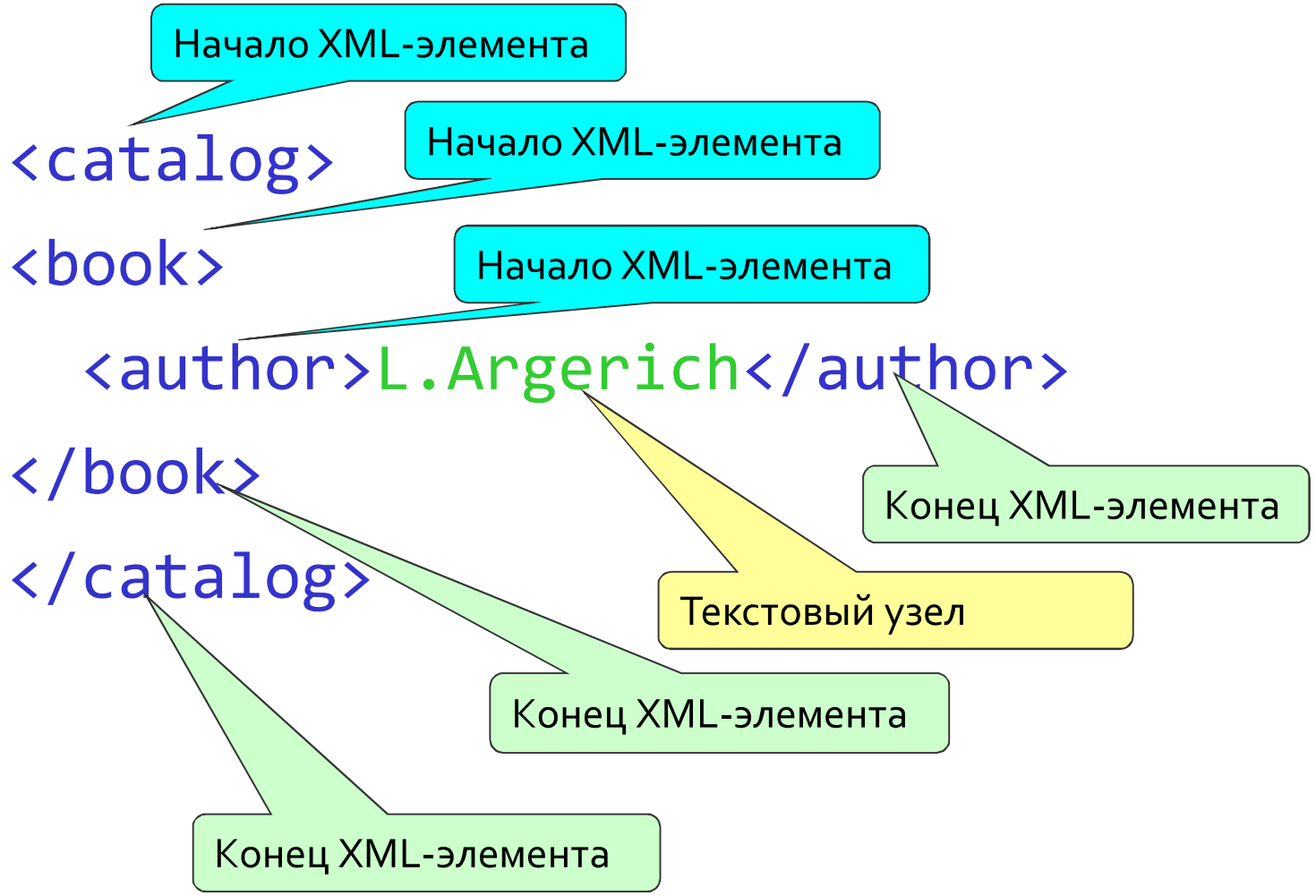
Средства PHP для работы с XML

- SAX (Simple API for XML)
  - получение информации из XML-документа
- DOM (Document Object Model)
  - чтение, модификация и создание новых XML-документов
- SimpleXML
  - чтение и модификация XML-документов
- XMLReader и XMLWriter
  - чтение и модификация XML-документов
- XSL/T (Extensible Stylesheet Language Transformations)
  - преобразование XML-документов в другие форматы

SAX (Simple API for XML)

- Официальный сайт: http://www.saxproject.org/
- Описывает метод парсинга XML-документов для получения данных из них
- Создавать и изменять XML-документы с помощью SAX невозможно
- Основан на событиях
  - XML-парсеру предоставляется набор собственных функций для обработки различных типов XML-данных
  - Парсер автоматически вызывает эти функции в процессе последовательной обработки XML-документа

Функционирование SAX



Использование SAX

- Создание парсера
  - `$sax = xml_parser_create("utf-8");`
- Декларация функций обработки
  - `function onStart ($parser, $tag, $attrs) {}`
  - `function onEnd ($parser, $tag) {}`
  - `function onText ($parser, $text) {}`
- Регистрация функций
  - `xml_set_element_handler($sax, "onStart", "onEnd");`
  - `xml_set_character_data_handler($sax, "onText");`
- Запуск парсера
  - `xml_parse($sax, "XML СТРОКА!");`
- Обработка ошибок
  - `xml_error_string(xml_get_error_code($sax));`

DOM

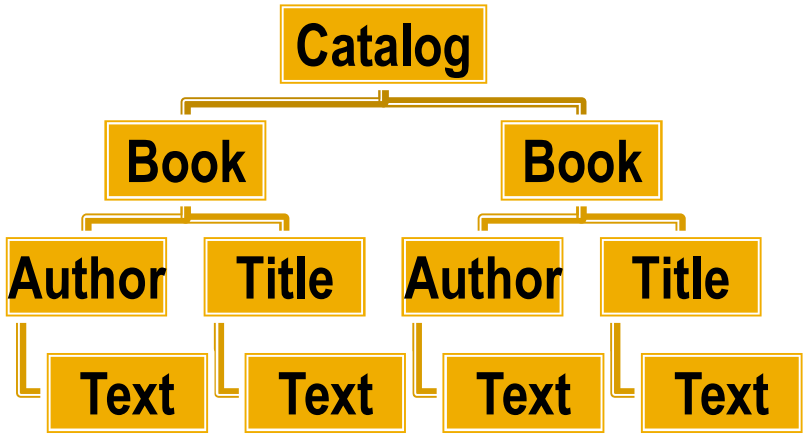
- DOM
  - Document Object Model
- Интерфейс, позволяющий программам управлять содержимым документов XML, а также изменять их структуру
- Представляет XML-документ в виде дерева узлов
- Существует спецификация DOM (W3C)

```
$dom = new DOMDocument();
$dom = new DOMDocument('1.0', 'utf-8');
```

Дерево документа

```
<catalog>
   <book>
      <author>Хьюз</author>
      <title>PHP</title>
   </book>

   <book>
      <author>Григин</author>
      <title>Справочник</title>
   </book>
</catalog>
```

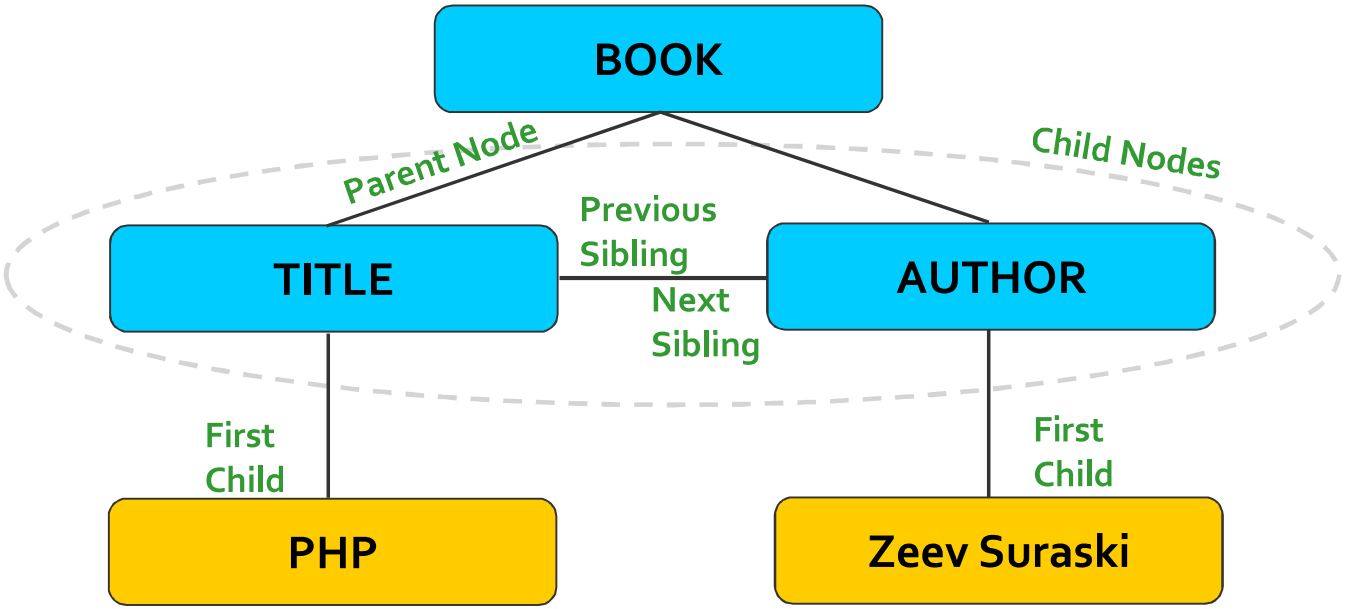


Типы узлов документа

Код типа	Тип узла	Описание	Пример
1	ELEMENT	Элемент	<book>…</book>
2	ATTRIBUTE	Атрибут элемента	lang="ru"
3	TEXT	Текстовый узел	Это текст
8	COMMENT	Комментарий	<!– Комментарий -->
10	DocumentType	Декларация типа документа	<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

Связи между узлами

```
<book>
  <title>PHP</title>
  <author>Zeev Suraski</author>
</book>
```



Работа с DOM на чтение

- Загрузка документа
  - `$dom->load('catalog.xml');`
- Получение корневого элемента
  - `$root = $dom -> documentElement;`
- Получение типа узла
  - `echo $root -> nodeType;`
- Получение дочерних узлов
  - `$children = $root -> childNodes;`
- Получение текстового содержимого узла
  - `echo $root -> textContent;`
- Получение узлов с определенным именем
  - `$titles = $dom->getElementsByTagName('title');`

Создание нового XML-элемента

- Создание нового элемента
  - `$book = $dom -> createElement('book');`
  - `$title = $dom -> createElement('title');`
- Создание текстового узла
  - `$text = $dom -> createTextNode('PHP 5');`
- Присоединение новых элементов
  - `$title -> appendChild($text);`
  - `$book -> appendChild($title);`
  - `$root -> appendChild($book);`
- Сохранение дерева в документ
  - `$dom->save('catalog.xml');`
- `$title = $dom->createElement('title', 'PHP 5');`