NURE

Гипертекст и гипермедиа

ведущий лектор 1-го потока

Ерохин Андрей Леонидович

лектор 2-го потока

Груздо Ирина Владимировна

NURE

Лекция №11
Основы стандарта DOM

Уровни DOM

- **DOM** – прикладной интерфейс программирования для HTML и XML документов
- Стандарт определяет логическую структуру документа и способы доступа и манипулирования документами

- DOM разработан с целью использования с любыми языками программирования

  Популярные реализации стандарта:
  OMG (Object M Group) в виде IDL (Interface definition language)

NURE

Level 0:

1) Включает в себя все специфические модели DOM, которые существовали до появления Уровня 1, например, **document.images**, **document.forms**, **document.layers** и **document.all**. Необходимо обратить внимание, что эти модели формально не являются спецификациями DOM, опубликованными W3C, а скорее являются информацией о том, что существовало до начала процесса стандартизации.

2) DOM Level 0 не описан в каком-либо документе, но частично вошел в стандарт HTML 4.

NURE

была стандартизована в октябре 1998 года

Level 1:

1) 1) базовые функциональные возможности по обработке XML-документов:

- обход по структуре XML-файла, получение всех потомков отдельного узла, проверка типа узла - является ли он узлом элемента или узлом атрибута или другим типом узла
- задание и чтение атрибутов отдельного узла элемента
- переход к следующему или предыдущему сестринскому узлу отдельного узла
- добавление или удаление атрибутов отдельного узла элемента
- добавление нового потомка к узлу
- получение списка всех элементов с определённым именем
- работа с инструкциями обработки XML

- 2) специальные интерфейсы, которые могут обрабатывать отдельные HTML-элементы (работа с таблицами HTML, формами)

- 3) нет средств (методов и свойств) для динамического управления элементами страницы, не содержит управления событиями, не содержит управления доступом к странице, не обеспечивает рендеринг через таблицы стилей

https://www.w3.org/TR/REC-DOM-Level-1/

была стандартизована в ноябре 2000 года

Level 2: Level 1 +

- регламентирует разработку событийной системы, которая должна позволять регистрацию событий, описывать всплывание событий по дереву, обеспечивать методы перехвата событий и выдавать контекстуальную информацию для каждого объекта

- поддержка пространства имен

(В Level 1 отсутствует)

- для работы с несколькими пространствами имен в одном документе DOM уровня 1 требовалось написание собственной программной логики для управления URI пространств имен, префиксами и именами элементов

https://www.w3.org/TR/DOM-Level-2-Core/

была стандартизована в ноябре 2000 года

Level 2: Level 1 +

- Состоит из следующих основных частей:
- DOM2-Core описывает XML, включая основные понятия: Node, Document и пр. Этот стандарт бывает полезен изредка. Он поддерживается очень хорошо всеми браузерами.
- DOM2-HTML расширяет Core для HTML/XHTML.
  При этом Document дополняется методами и свойствами до HTMLDocument, Node - до HTMLElement.
  Пожалуй, самый полезный стандарт, отлично освоенный браузерами.
- Стандарт DOM2-Events описывает события W3C.
  Как известно, здесь много кросс-браузерных несовместимостей.
- Стандарт DOM2-Style описывает работу с CSS: с файлами стилей StylesheetsСтандарт DOM2-Style описывает работу с CSS: с файлами стилей Stylesheets и CSS-правилами.
  Поддерживается неплохо, однако некоторые несовместимости в браузерах все же есть.
- Стандарт DOM2-Traversal-Range описывает DOM 2 TraversalСтандарт DOM2-Traversal-Range описывает DOM 2 Traversal - итераторы и фильтры для обхода документа, и DOM 2 Range - средства для выделения текста и DOM-узлов.
  Traversal браузеры только начинают осваивать, ну а Range хорошо

NURE

Level 3: Level 2 +

- платформо- и языково- нейтральный интерфейс
  позволяет программам и скриптам получать динамический доступ и динамически обновлять контент, структуру и стили документа

- возможность работать с многочисленными расширениями DOM, предназначенными для схем данных (например, для собственных XML-словарей можно определять программные интерфейсы для схемы) - высокоуровневый программный интерфейс для манипулирования XML-документами в соответствии с собственной схемой

https://www.w3.org/TR/DOM-Level-3-Core/

NURE

В XML выделены "фундаментальные" стандарты, составляющие концептуальную и синтаксическую основы платформы.

Структурообразующие стандарты:

- Xpointer - язык, позволяющий поддерживать адресацию во внутренней структуре XML
- XLink - язык декларации различного рода связей между XML-ресурсами
- XInclude - включение документов
- XFragment - обмен фрагментами XML-документов
- XPath - язык путей XML (задание модульности XML-документов)

Стандарты форматирования и трансформации XML-документов:

- XSL/XSLT - расширяемый язык таблиц стилей + форматные преобразования
- CSS – каскадные таблицы стилей
- SVG – язык масштабируемой векторной графики

NURE

Стандарт управления запросами

XQuery - язык запросов для XML

В октябре 1999 года W3C образовал рабочую группу XML Query Working Group [6] с целью разработки такого языка запросов, получившего название Xquery

В XML-документах имеется внутренний порядок, а реляционные данные неупорядочены, XML-данные часто бывают <разреженными>

https://www.w3.org/XML/Query/

- **Стандарты инструментариев для разработчиков**
- **DOM – объектная модель документа**
- **SAX - простой API для XML**
- **XOM - объектная модель XML**
- **WSDL – язык описания программных интерфейсов для взаимодействия различных веб-сервисов**
- **SOAP – универсальный стандарт удаленного вызова процедур на языке XML**
- **XML RPC – технология вызова удаленных процедур с использованием протокола HTTP и XML**
- **XUL – язык описания программных интерфейсов**

NURE

Стандарты обеспечения преемственности HTML

- XHTML - расширяемый гипертекстовый язык разметки
- XML Base - стандарт поддержки средствами стандарта XLink некоторых видов гиперссылок, используемых в языке HTML

Стандарты HTTP-транспорта данных

- XForms - расширенный язык описания веб-форм

Стандарты безопасности

- XML Signature (XMLEncryption) - стандарт цифровой подписи
- XKMS - инфраструктуры поддержки открытых ключей
- XACML - язык разметки управления доступом XML
- SAML - язык разметки утверждений безопасности

NURE

Рекомендации

- XML EDI (UN/EDIFACT и ANSI X-12) – рекомендации по обмену электронными документами
- UDDI - универсальный метод описания, обнаружения и интеграции web-сервисов для B2B систем электронной коммерции
- WCAG – рекомендации по созданию общедоступных сайтов
- ATAG - руководящие принципы авторской разработки удобных инструментов

Специализированные стандарты

- MathML, CML, GML – языки математических, химических и географических определений
- VoiceXML – описание протокола передачи голосовых данных
- WML – язык интерфейса мобильных устройств
- SMIL – язык интерактивных мультимедийных приложений и презентаций
- UAAG – язык описания принципов создания пользовательских программ, снижающие барьеры для доступности в Сети.

NURE

Фундаментальные мета-данные

- XML Information Set (InfoSet) набор информационных элементов XML
- XML Namespace – пространство имен XML

Стандарты, определяющие структуру документов

- DTD – описание документов определенного типа
- XML Schema (Relax NG) - язык определения схемы XML
- RDF - Стандарт средств описания семантики информационных ресурсов для среды XML (Dublin Core 1.1)

Структура DOM-модели XML

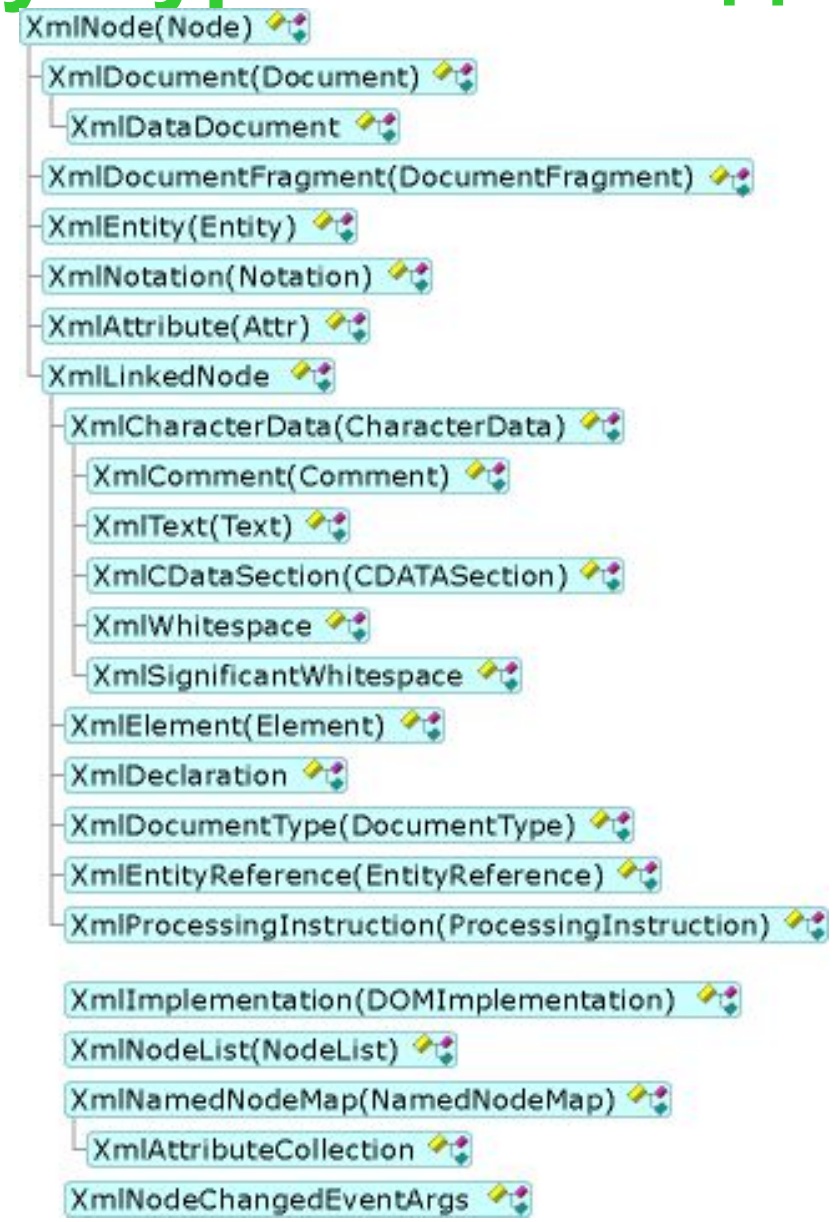
DOM представляет документ как иерархию объектов Node (узлы), которые реализуют другие, более специализированные интерфейсы

Некоторые типы узлов могут иметь унаследованные узлы различных типов, а некоторые *Node* являются листьями и от них не может ничего наследоваться

Объекты **Node** имеют набор методов и свойств, а также базовые и четко определенные характеристики. Некоторые из этих характеристик:

- Узлы имеют один родительский узел, родительский узел является узлом, расположенным непосредственно над ними. Единственным узлом, у которого нет родителя, является корень документа, поскольку он является узлом верхнего уровня и содержит сам документ и фрагменты документа.
- Большинство узлов могут иметь несколько дочерних узлов, которые находятся непосредственно под ними.

Структура DOM-моделі XML



NURE

Структура DOM-модели XML

- DOM представляет XML-документ в виде дерева, состоящего из узлов. В DOM определяются различные типы узлов, соответствующие разным конструкциям XML

Например, XML-элемент — это узел элемента, пара атрибут XML и его значение — это узел атрибута, содержание элемента — это текстовый узел

- XML-документ может быть загружен в дерево узлов DOM

- Узлы модели DOM связаны между собой отношениями родитель-потомок или сестринский узел.
- Все узлы в DOM являются прямыми или косвенными потомками узла document

- Интерфейсы DOM используют понятие наследования, определённое в ООП
- Различные типы узлов (узел элемента, узел атрибута, текстовый узел) наследуется из родового интерфейса Node

NURE

Структура DOM-модели XML

- Каждый узел — это объект, который содержит и данные, и методы для манипулирования этим объектом

- Например, можно взять узел, вызвать его методы, чтобы получить список узлов элементов-потомков этого узла.
- Затем можно взять отдельный элемент-потомок и вызвать его методы, чтобы пойти дальше по дереву DOM.

В спецификации DOM для различных типов узлов определены программные интерфейсы для вызова их функциональности

Типы узлов

Узел	*Унаследованные*
Document	Element (max 1), Comment, ProcessingInstruction, DocumentType
DocumentFragment	Comment, ProcessingInstruction, DocumentType, Text, CDataSection, EntityReference
DocumentType	
Element	Element, Comment, ProcessingInstruction, Text, CDataSection, EntityReference
EntityReference	Element, Comment, ProcessingInstruction, Text, CDataSection, EntityReference
Attr	Text, EntityReference
ProcessingInstruction	
Comment	
Text	
CDATASection	
Notation	
Entity	Element, Comment, ProcessingInstruction, Text, CDataSection, EntityReference

https://docs.microsoft.com/en-us/dotnet/standard/data/xml/types-of-xml-nodes

https://docs.microsoft.com/en-us/dotnet/standard/data/xml/mapping-the-object-hierarchy-to-xml-data

NURE

Например:

В DOM описан интерфейс NodeList– список узлов

На практике его вызывает метод :
Element.getElementByTagName ( ).NameNodeMap –

// доступ к неупорядоченным узлам, таким как NodeList

NURE

Управление памятью

Большинство API определены не столько как классы, сколько как интерфейсы.

Обычные конструкции ООП языков не могут быть использованы для создания DOM-объектов

- Решение: определение фабричных методов (factory), которое создает сущности объектов

В интерфейсе ***document*** создается некоторый документ X
с помощью метода createX( ) интерфейса ***document,***
где X – это имя объекта.

- Ядро DOM API разработано с целью обеспечения совместимости широкого диапазона языков

NURE

Управление памятью

- Платформа Microsoft .NET Framework включает три модели обработки XML-данных: классы XmlDocument, класс XPathDocument, а также LINQ to XML (C#) и LINQ to XML (Visual Basic).
- Класс XmlDocument реализует базовую модель DOM W3C 1-го уровня и базовые рекомендации объекта DOM 2-го уровня. DOM - древовидное представление XML-документа в памяти (кэш). С помощью XmlDocument и связанных классов можно конструировать XML-документы, загружать данные и обращаться к ним, изменять данные и сохранять изменения.
- Класс XPathDocument - доступное только для чтения хранилище данных в памяти, на базе модели данных XPath. В классе XPathNavigator предусмотрено несколько вариантов редактирования и способов навигации с помощью модели курсора для XML-документов в доступном только для чтения классе XPathDocument, а также в классе XmlDocument.
- LINQ to XML — это модель для обработки XML-данных, представленная на платформе .NET Framework версии 3.5. Это размещаемая в памяти модель, которая использует синтаксис LINQ. LINQ расширяет синтаксис C# и Visual Basic, обеспечивая новые возможности запросов.

NURE

Соглашения об именах

В DOM предполагается поддержка структуры имен,
как OMG IDL, так и ECMAScript (язык программирования, определенный стандартом ECMA-262).
Представитель - JavaScript

- OMG → CORBA (Common Object Request Broker Architecture)

CORBA – коллекции объектов и библиотек, которые позволяют создать приложения, содержащие объекты, которые создают и получают запросы и отклики (ответы) в распределенных открытых системах.

- Объектная модель OSI из которой “выросли” стандарты (TCP / IP)

- «Выберите один стиль и будьте последовательны»!

Характеристики интерфейсов

- **Node -** базовый интерфейс для остальных элементов объектной модели
- **Document -** используется для получения информации о документе и изменении его структуры. В XML этот интерфейс представляет корневой элемент документа. Содержит методы доступа, позволяющие создавать дочерние конструкции.

Например:

createElement () // создает новый элементный узел. Если вновь созданный элементный узел существует в объекте элемента, он заменяется новым. Несмотря на то, что этот метод создает новый объект в контексте документа, он не добавляет автоматически новый объект в дереве документа.

createComment ()

createText () // создает новый узел комментария. Узел комментариев включен в программу для легкого понимания функциональности кода.

Методы создают различные дочерние объекты, перемещают, добавляют и т.п.

Например:

removeChild () // удаляет дочерний узел

replaceChild () // заменяет один дочерний узел на другой

- **В XML интерфейс Document называется XMLDOMDocument.**
- области видимости в порядке убывания: сервер, приложение, сеанс, запрос и страница

Интерфейс Document Fragment

Интерфейс *DocumentFragment –*

минимальный объект класса Document. который не имеет родителя. Он используется как легкая версия **Document**, чтобы хранить хорошо сформированные или потенциально не хорошо формированные фрагменты XML.

Вставляется в объект Document

Стандартные методы:

appendChild( ) // Добавляет элемент в конец списка дочерних элементов родителя. Если элемент уже существует он удаляется из текущего родителя и вставляется заново.

Если child ссылается на существующий элемент в документе , тогда appendChildперемещает элемент с его текущей позиции на новую (т.е необязательно удалять элемент из родителя перед тем как перемещать его в другой элемент).

insertBefore( ) // вставляет новый дочерний узел перед существующим дочерним узлом. Этот метод возвращает новый дочерний узел.

Синтаксис: interface DocumentFragment: Node{ };

Интерфейс Document

Interface Document представляет целый *html* или *xml* документ. Концептуально является корнем дерева Document. Позволяет осуществить доступ к данным документам.

Содержит фабричные методы для создания допустимых узловых объектов

Пример:

```
Interface Document: Node{
    readonly attribute DocType;
    readonly attribute Element;
     createAttribute (DOMStringName);
     …
                        }
```

Атрибут DocType устанавливает декларацию типа документа (DTD), связанную с этим документом.

Если атрибут отсутствует в конкретном экземпляре документа, то он по умолчанию равен NULL.

NURE

В DOM регламентированы интерфейсы, которые обрабатывают исключительные ситуации

Они объявляются следующим образом:

- exception DOMException { }

Внутри этого объекта содержатся константы, отвечающие за код ошибок:

- Описание ошибки.
- Исключение, которое является причиной текущего исключения.
- Номер строки, показывающий, где произошла ошибка.
- Размещение строки, показывающее, где произошла ошибка.

Глобальные стандартные свойства

Согласно стандарту DOM, любому элементу в DHTML могут быть добавлены любые атрибуты, доступ к которым может осуществляться с помощью элементов типа <CDATA>

- *className* - свойство, используется для того, чтобы определить, какое значение из параметров таблицы стилей использует данный элемент

<img src = 'url' class = 'plavno'>

Атрибут **document** реализуется через свойство документа. Содержит ссылку к объекту документа, в котором содержится элемент.

Поддержка событий для объектов типа <CDATA>

- Свойство **event.type** возвращает имя события без приставки ***on***

  event.type = 'mouseover'

  Пример использования:

  switch (event.type)

  {case 'click': код; break}

Например: для работы с мышью существуют свойства which / button :

- event.which == 1 – левая кнопка
- event.which == 2 – середняя кнопка
- event.which == 3 – правая кнопка

NURE

Введение в XML

- XML – eXtensible Markup Language (Расширяемый Язык Разметки)
- Первая официальная версия XML (1.0) - 1998 р.
- XML является подмножеством SGML (Standard Generalized Markup Language)
- XML-документы используют простой и самоописываемый синтаксис

NURE

Введение в XML

- XML - метаязык, предназначенный для создания языков разметки
- В XML расширяемым является не сам XML, а язык документа, составленный на XML
- Если в HTML используется один и тот же набор правил языка, то в XML можно использовать собственные наборы правил
- ссылка либо на набор правил в отдельном файле, либо на набор правил, встроенных в документе

XML - это метаязык, на котором пишутся специализированные языки, описывающие данные определенной структуры
Такие языки называются *XML-словарями*

XML не содержит никаких указаний на то, как описанные в XML-документе данные должны отображаться

Способ отображения данных для различных устройств задается языком описания стилей XSL, который играет для XML примерно ту же роль, что CSS для HTML

XML может содержать любые теги, которые сочтут нужным использовать создатели XML-словаря

- MathML - язык математических формул
- SMIL — язык интеграции и синхронизации мультимедийных средств
- SVG — язык двумерной векторной графики
- RDF — язык метаописаний ресурсов
- XHTML — переформулировка HTML в терминах XML

NURE

При определении правил в XML-интерпретаторах возможны теги, например:

<студент>Петренко</студент>

При *передаче:*

Петренко – экземпляр объекта студент

Анализатор XML должен обеспечивать семантический анализ содержимого.

Структура XML-документа

```
<?xml version="1.0" encoding="CP-1251"?>
<!- письмо -->
<note date="15.05.2019">
  <to>Петренко</to>
  <from>Сидоров</from>
  <heading>Напоминание</heading>
  <body>Не забудь о встрече!</body>
</note>
```

- Пролог
  <?xml version="1.0" encoding="CP-1251"?>
- Корневой элемент
  <note> … </note>
- Элементы
  <to>Сидоров</to> …
- Атрибуты
  date="15.05.2019"
- Текст
  Не забудь о встрече

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

NURE

Разметка

- Инструкции синтаксического анализатора

```
<?xml version="1.0" encoding="CP-1251"?>
```

- Текст разметки заключен в угловые скобки ('<' и '>') – это теги

```
<note date="15.04.2010">
  …
</note>
```

- Символьные данные – это текст, расположенный между открывающим и закрывающим тегами.

```
<body>
  Не забудь о встрече!
</body>
```

- Комментарии

```
<!– е письмо -->
```

NURE

XML- элементы

- Тег – это идентификатор, кодируемый парой угловых скобок
  - открывающий тег <node>
  - закрывающий тег </node>
  - самозакрывающий тег <image src="img.png"/>

- Элемент разметки начинается открывающим тегом и заканчивается закрывающим тегом

- Содержимое XML-элемента:
  - текст <from>Петренко</from>
  - дочерние XML-элементы

```
<game>
 <player name="Вася"/>
 <player name="Петя"/>
</ game>
```

NURE

Атрибуты XML-элемента

- Атрибуты описывают характеристики XML-элемента
- Указываются в открывающем теге XML-элемента
- Синтаксис атрибута:

  ім’я = "значение"

```
<note date="15.04.2010">
   …
</note>
```

Символы

- в XML можно использовать любые символы ASCII
  - в прологе указывают кодировку:
    - <?xml version="1.0" encoding="CP-1251"?>
- зарезервированные символы: '<', '>', '&', '' ''.
- Ссылки на сущность:
  - начинается с амперсанта ('&'),
  - заканчивается точкой с запятой (';').

& – амперсанд
< – левая угловая скобка
> – правая угловая скобка
' – апостроф
" – кавычки

NURE

Разделы CDATA

- в XML значимыми являются все символы
- нормализация – пропуск пробельных символов
- теги <![CDATA[и]]> позволяют использовать текст, включающий любые инструкции и специальные символы

```
<gameData> <![CDATA[
!@#$$%^&**&^%$#@!
]]></gameData>
```

NURE

Правильные XML-документы

- текст должен подчиняться определенным правилам – "*well formed* “ -- XML Schema - определения правил, которым должен подчиняться документ
- теги могут быть открывающими и закрывающими или самозакрывающими
- вложенность тегов
- если текст не подчиняется правилам “well formed”, то он не является XML-документом

NURE

Правила синтаксиса (Валидность)

Структура XML документа должна соответствовать определенным правилам. XML документ отвечающий этим правилам называется *валидным* (*англ.* Valid — правильный) или *синтаксически верным*. Соответственно, если документ не отвечает правилам, он является *невалидным* .

Основные правила синтаксиса XML:

- Теги XML регистрозависимы — теги XML являются регистрозависимыми. Так, тег <Letter>не то же самое, что тег <letter>.
- Открывающий и закрывающий теги должны определяться в одном регистре:

<Message>Это неправильно</message>

<message>Это правильно</message>

- XML элементы должны соблюдать корректную вложенность:

Некорректная вложенность

Корректная вложенность

- У XML документа должен быть корневой элемент — XML документ должен содержать один элемент, который будет родительским для всех других элементов. Он называется корневым элементом.

NURE

Конфликт имен

информация в таблице

```
<table>
  <tr>
    <td>Яблоки</td>
    <td>Бананы</td>
  </tr>
</table>
```

информацию о столе

```
<table>
  <name>Журнальный столик</name>
  <width>80</width>
  <length>120</length>
</table>
```

Использование префикса

информация в таблице

```
<fruit:table>
   <fruit:tr>
      <fruit:td>Яблоки</fruit:td>
      <fruit:td>Бананы</fruit:td>
   </fruit:tr>
</fruit:table>
```

информация о столе

```
<furniture:table>
   <furniture:name>Журнальный столик</furniture:name>
   <furniture:width>80</furniture:width>
   <furniture:length>120</furniture:length>
</furniture:table>
```

NURE

Пространства имен XML

- При использовании в XML префиксов необходимо определить, так называемое, пространство имен префикса.
- Унифицированный идентификатор ресурса

```
<table xmlns:fruit="http://www.w3.org/TR/html4/">
```

- Дочерние элементы ассоциируются с тем же пространством имен начального тега

- URI – уникальное имя в пространстве имен
- В качестве URI используют URL

Пространства имен по умолчанию

информация в таблице

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Яблоки</td>
    < td>Бананы</td>
  </tr>
</table>
```

информация о столе

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>Журнальный столик</name>
  <width>80</width>
  <length>120</length>
</table>
```

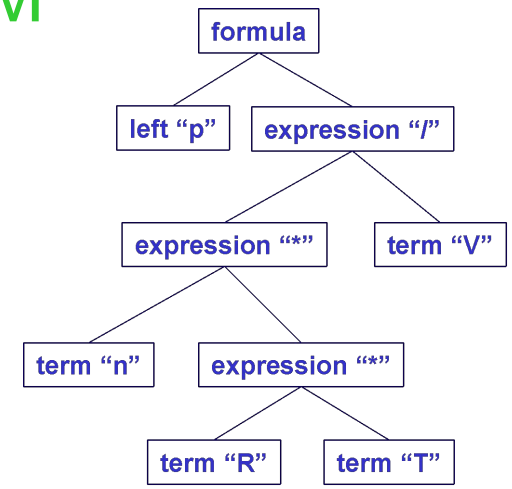
Пространства имен могут декларироваться либо непосредственно в самом элементе, либо в корневом элементе XML документа

```
<root xmlns:fruit="http://www.w3.org/TR/html4/"
xmlns:furniture="http://www.w3schools.com/furniture">
```

Способы программного представления XML-документа. SAX и DOM

- DOM – Document Object Model
  - строит дерево документа в памяти
  - предоставляет произвольный доступ к данным

- SAX – Simple API for XML – простой программный интерфейс для XML
  - данные XML-документа обрабатывается по мере их поступления
  - использует событийную модель
  - при появлении элементов разметки вызывают методы, определенные программистом




```
<?xml version="1.0"?>
<parts>
  <part>TurboWidget</part>
</parts>
```

```
StartDocument( );
StartElement( "parts" );
StartElement( "part" );
Characters( "TurboWidget" );
EndElement( "part" );
EndElement( "parts" );
EndDocument( );
```

Объектная модель документа (DOM)

- DOM-интерфейсы являются способом представления внутренней структуры документа
- Приложение может не знать о способе реализации интерфейсов, ему доступна готовая библиотека методов
- DOM предоставляет программисту гибкую технологию доступа к XML-документу

NURE

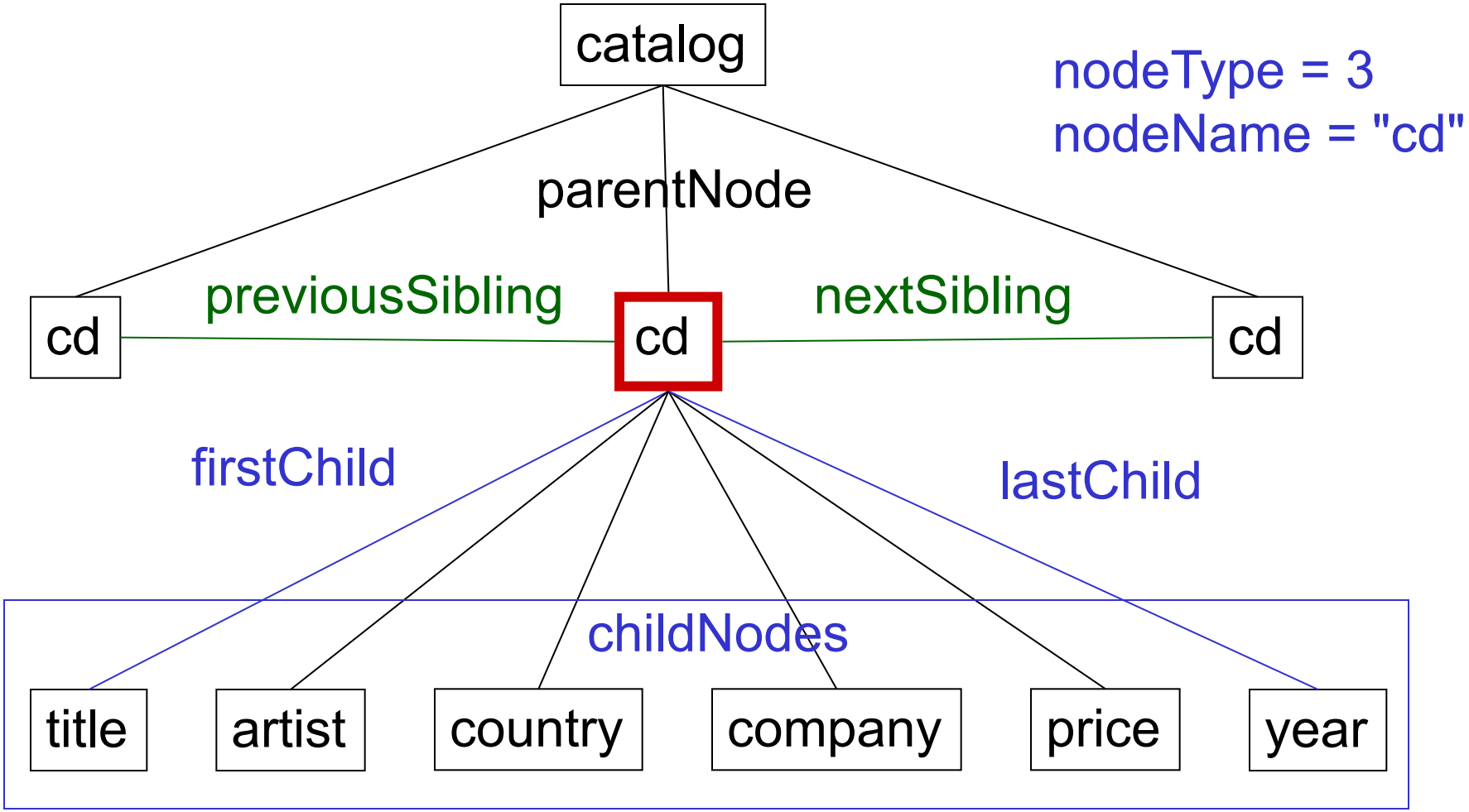
Объекты XMLElement

Объект **XMLElement** предназначен для манипулирования с отдельным узлом дерева документа. Являются одним из наиболее распространенных узлов в модели объектов W3C документов (DOM)

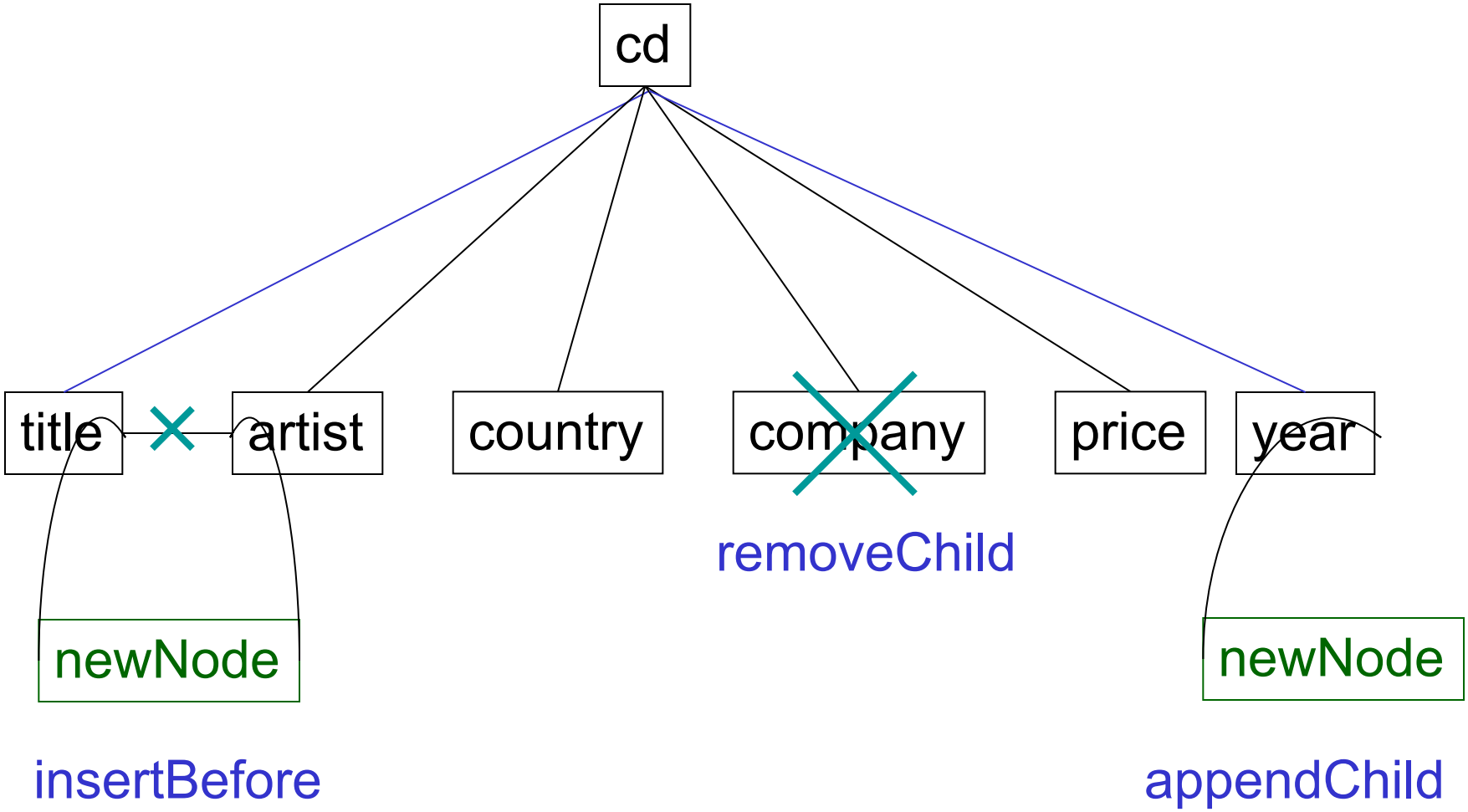
Каждый элемент имеет следующие свойства:

- ссылка на элемент уровня выше
- ссылка на список дочерних элементов
- ссылка на соседние элементы
- список атрибутов

Свойства объекта XMLElement



Методы объекта XMLElement



NURE

Объект XMLDocument

- объект XMLDocument представляет верхний уровень объектной иерархии
- унаследован от XMLElement
- предназначен для создания элементов, атрибутов, комментариев

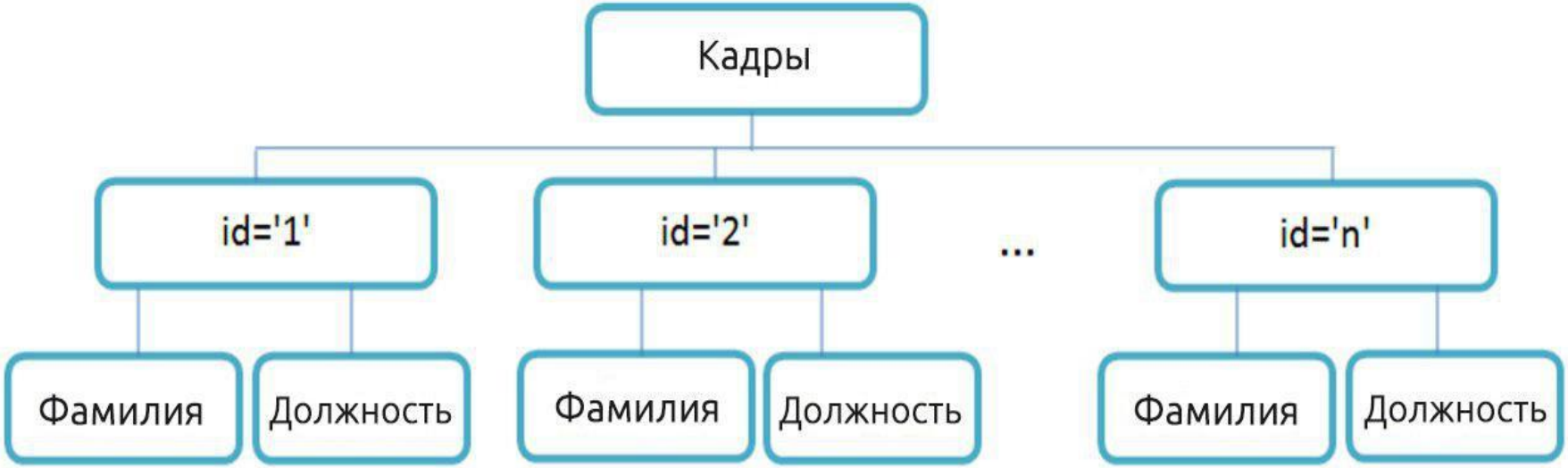
Структура XML – документа

<?XML version='1.0'?>
<!doctype ' .dtd'>

.dtd – это файл правил, которые описывают интерпретацию

Пример:

<кадры>
< рабочий id='1'>
<прізвище>Петренко</прізвище>
<посада>слесарь</посада>
…
</рабочий>
</кадры>


Кадры
id='1'
id='2'
...
id='n'
Фамилия
Должность
Фамилия
Должность
Фамилия
Должность

NURE

1) В XML теги делятся на открывающиеся, закрывающиеся и пустые

2) Тело документа XML состоит из элементов разметки (markup) и содержимого документа (content)

3) XML-теги предназначены для определения элементов документа, атрибутов документов

4) Первый тег XML-документа задает номер версии языка, номер кодовой страницы

5) В XML учитывается регистр символов

6) Все значения атрибутов заключаются в кавычки

7) Вся информация, которая располагается в контейнере, рассматривается как данные, то есть учитываются все символы форматирования