Курс «Языки разметки»

Лекция 5. Связывание XML-элементов: языки XLink, XPointer

Содержание

- Язык XLink
- Язык XPointer

Язык XLink

- ***XML Linking Language (XLink)*** - это разработанный в консорциуме W3C язык разметки, позволяющий вставлять в XML документы элементы, чтобы создать и описать ссылки между ресурсами.
- XLink использует синтаксис XML, чтобы создать структуры, которые смогут описать как простые однонаправленные ссылки сегодняшнего HTML, так и сложные ссылки.
- Пример простой ссылки, аналогичной html-элементу A:

```
<my_link xlink:type="simple"
        xlink:href="http://mysite.com"
        my_alt="Мой сайт"/>
```

Возможности XLink

- XLink позволяет:
  - создавать двунаправленные ссылки (туда-обратно),
  - создавать мультинаправленные ссылки,
  - формировать кольцо ссылок на однородные ресурсы
  - как напрямую адресовать конкретный документ, так и указывать относительное положение ресурса (**XPointer**)

Типы элементов XLink

- **simple** – простые ссылки (подобные "a" и "img" в HTML)
- **extended** – расширенные ссылки (двунаправленные и пр.)
- **extended link groups** – группы расширенных ссылок

Структура ссылок в XLink

- Все гиперссылки в XLink являются обычными элементами XML с дополнительными **атрибутами**:
- **xlink:href** – определяет адрес ресурса;
- **xlink:type** – определяет тип ресурса (simple, extended и др.)

Простые ссылки

Пример:

- XML:

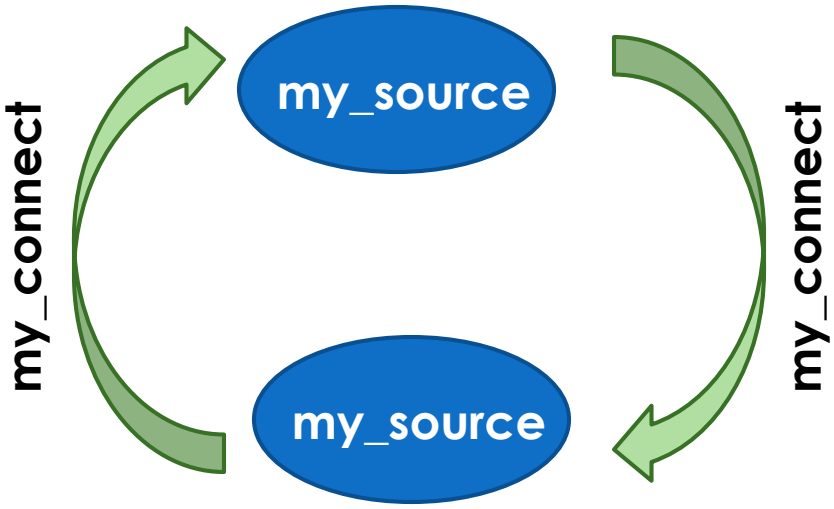
```
<my_element xlink:type="simple"
            xlink:href="http://mysite.ru" />
```

- DTD, описывающий данный XML:

```
<!ELEMENT my_element EMPTY>
<!ATTLIST my_element
   xlink:type CDATA #FIXED "simple"
   xlink:href CDATA #REQUIRED>
```

Расширенные ссылки

- Каждая расширенная ссылка является элементом, который включает в себя ***ресурсы и их связи***.
- Пример: опишем следующую ссылку



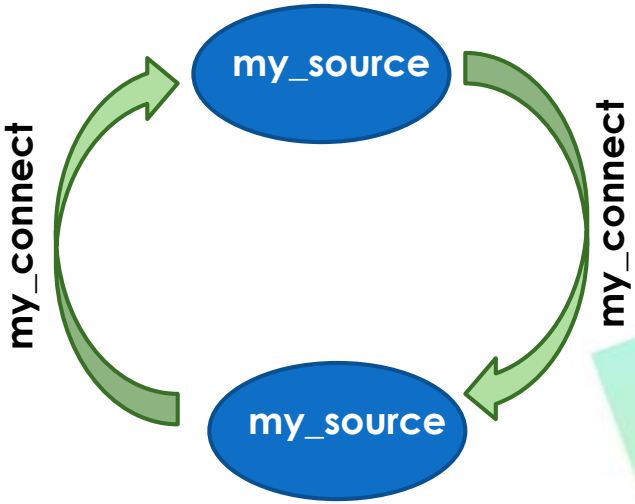
Расширенные ссылки

- Пример:

```
<my_link xlink:type="extended">
    <my_source ...>First site</my_source>
    <my_source ...>Second site</my_source>
    <my_connect .../>
    <my_connect .../>
</my_link>
```

РЕСУРСЫ

СВЯЗИ



Типы ресурсов (xlink:type) в расширенных ссылках

- **Локальные (внутренние) ресурсы** - такой ресурс, который является частью расширенной ссылки.

  xlink:type = resource

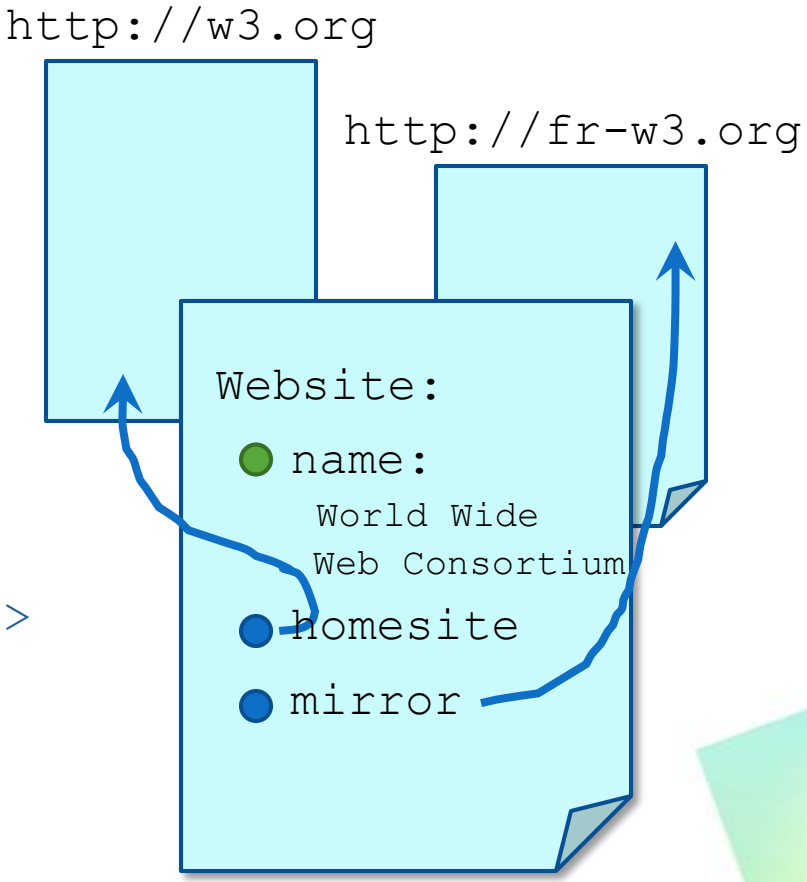
- **Внешние ресурсы** - такой ресурс, который не является частью элемента расширенной связи и обычно находится в другом документе.

  xlink:type = locator

Типы ресурсов (xlink:type) в расширенных ссылках

- Пример:

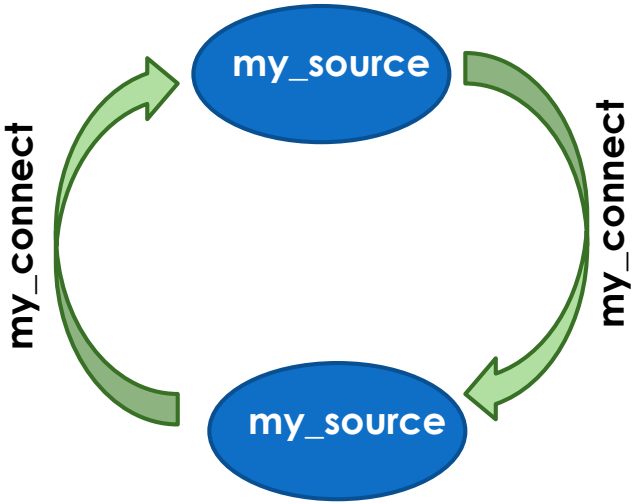
```
<website xlink:type="extended">
  <name link:type="resource">World
      Wide Web Consortium </name>

  <homesite xlink:type="locator"
      xlink:href="http://w3.org"/>

  <mirror xlink:type="locator"
      xlink:href="http://fr-w3.org"/>
</website>
```



Расширенные ссылки

- Пример (ресурсы):

```
<my_link xlink:type="extended">
   <my_source xlink:type="locator"
     xlink:href="http://www.site1.com">
     First site</my_source>
   <my_source xlink:type="locator"
     xlink:href="http://www.site2.com">
     Second site</my_source>
    <my_connect .../>
    <my_connect .../>
</my_link>
```



Связь в расширенных ссылках

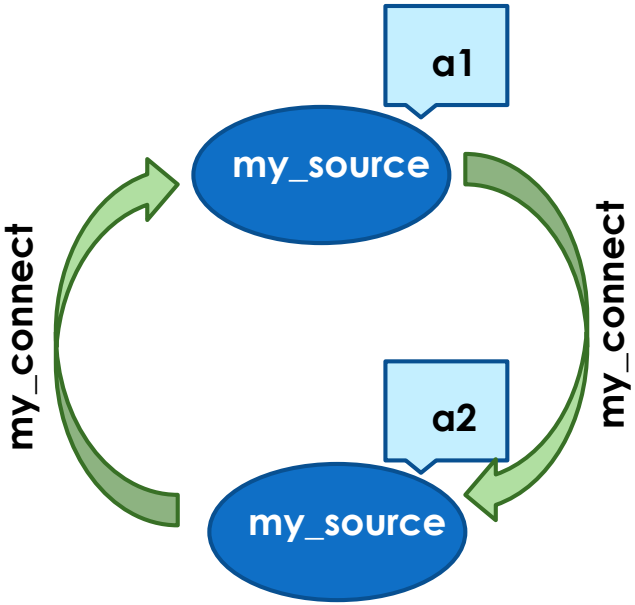
- По сравнению с простыми связями расширенные связи являются более сложными, т.к. они предоставляют множество различных путей перехода между ресурсами.
- Например, в расширенной связи с тремя ресурсами **A, B** и **C** возможны девять различных переходов:
  - A --> A (от A к A), B --> B, C --> C, A --> B, B --> A, A --> C, C --> A, B --> C, C --> B
- Каждый из этих потенциальных путей между ресурсами может иметь различные правила определения того, когда связь должна обработаться и что должно происходить при ее обработке. Эти потенциальные пути называются **ребрами (arc)**, они представляются с помощью элементов, у которых значение атрибута **xlink:type** равно **arc**.
- Для указания направления перехода используются атрибуты **xlink:to** и **xlink:from**.

Расширенные ссылки

- Для возможности создавать связь ресурсам присваивается метка с помощью атрибута **xsl:label**

◆ Пример (метки):

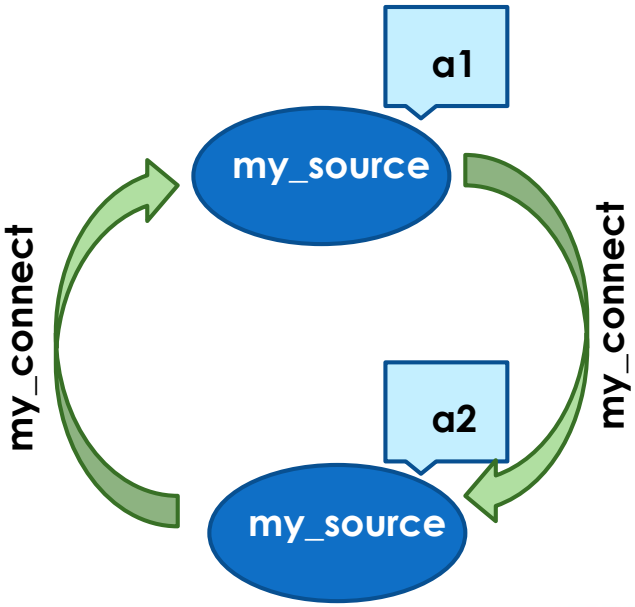
```
<my_link xlink:type="extended">
   <my_source xlink:type="locator"
     xlink:href="http://www.site1.com"
     xsl:label="a1">First site
   </my_source>
   <my_source xlink:type="locator"
     xlink:href="http://www.site2.com"
     xsl:label="a2">Second site
   </my_source>
    <my_connect .../>
    <my_connect .../>
</my_link>
```



Расширенные ссылки

- Пример (связи):

```
<my_link xlink:type="extended">
   <my_source xlink:type="locator"
     xlink:href="http://www.site1.com"
     xsl:label="a1">First site
   </my_source>
   <my_source xlink:type="locator"
     xlink:href="http://www.site2.com"
     xsl:label="a2">Second site
   </my_source>
    <my_connect xlink:type="arc"
     xlink:from="a1" xlink:to="a2"/>
    <my_connect xlink:type="arc"
     xlink:from="a2" xlink:to="a1"/>
</my_link>
```



Правила перехода

- Правила перехода указываются добавлением атрибутов **xlink:show** и **xlink:actuate** к элементам типа arc.

- **Атрибут xlink:show** предоставляет приложениям информацию о том, как следует отображать содержание, когда связь активизирована, например, должно ли открываться новое окно, чтобы показать адресуемое содержание, либо это содержание должно загружаться непосредственно в текущее окно.

- **Атрибут xlink:actuate** позволяет определить, когда активизировать связь, например, сразу после загрузки документа или исключительно после запроса пользователя.

Атрибут xlink:show

- xlink:show может принимать одно из следующих значений:
  - replace, new, embed, other, none.
- **replace** - при активизации связи (как правило, посредством щелчка мышкой по этой связи) адресат связи заменяет текущий документ в том же самом окне (это поведение является действием по умолчанию для связей HTML)
- **new** - активизация связи вызывает открытие нового окна, в котором отображается адресуемый ресурс. Это похоже на поведение связей HTML, когда атрибуту target присвоено значение blank.
- **embed** - при активизации связи адресуемый ресурс вставляется в существующий документ. Что именно это означает - зависит от приложения. Обычно предполагается, что приложение должно каким-то образом изобразить связываемое содержание и показать его как часть заключительного документа.
  - В качестве примера приведем фрагмент кода, в котором этот атрибут используется для того, чтобы указать, что изображение JPEG должно быть встроено в этот документ:

    <PHOTO xlink:type="simple" xlink:href="images/nypride.jpg" xlink:show="embed" ALT="Marchers on 5th Avenue, June 2000"/>

Атрибут xlink:show

- **other** - предполагается, что приложение будет искать другую разметку в документе, которая объяснит, что делать. Как правило, это могло бы использоваться, чтобы отдельное приложение XML использовало другие, отличные от XLink элементы для описания поведения связи. Например, у многих Web-страниц в заголовке находится элемент LINK, который указывает таблицу стилей (style sheet) и может выглядеть следующим образом:
  - <LINK REL="stylesheet" TYPE="text/css" HREF="http://www.w3.org/StyleSheets/TR/W3C-WD"/> Это связь, но то, что находится в ее конце, не заменяет текущий документ, не встраивается в него и не отображается в новом окне. Для XML-документов вы могли бы условиться, что такое поведение предполагается всякий раз, как встречается элемент STYLESHEET. Поскольку это поведение не является ни одним из трех предопределенных поведений связей, необходимо присвоить xlink:show значение other.
  - <STYLESHEET xlink:show="other" xlink:href="http://www.w3.org/StyleSheets/TR/W3C-WD"/> Наконец, атрибуту xlink:show может быть присвоено значение none, чтобы показать, что документ не содержит никакой информации, которая могла бы помочь приложению решить, что, если уж на то пошло, делать со связью. В этом случае все зависит только от приложения.
- Независимо от того, какое поведение атрибут xlink:show предлагает, браузер или иное приложение, читающее документ, при активизации связи может делать все, что угодно, в том числе и ничего. Например, браузер, у которого отключена автоматическая загрузка изображений, может решить проигнорировать xlink:show="embed".

Язык XPointer

- ***XML Pointer Language (XPointer)*** — расширяемая спецификация, определяющая способы адресации структурных элементов и фрагментов документов в формате XML.
- Спецификация XPointer включает несколько частей:
  - **описание базовых правил или каркаса (framework)**, служащего основой для различных схем адресации фрагментов XML-документов;
  - **описания этих схем**.
- Существуют три схемы:
  - element() – предназначена для поиска элементов по их расположению
  - xmlns() – предназначена для поиска элементов на основе пространства имен
  - xpointer() – предназначена для поиска элементов при помощи языка XPath.

Схема element()

- Схема element() ссылается на элемент XML-документа примерно в таком стиле:

  «сослаться на второй абзац третьего параграфа договора №5»

- Например:

  element(/1/3/2)

  В этом примере начальная наклонная черта и единица показывают, что отсчет начинается с корневого элемент. В нем выбирается третий по счету непосредственно вложенный элемент. В этом элементе выбирается второй по счету вложенный в него элемент.

Пространство имен в XML

- **Пространство имен в XML** — именованная совокупность имен элементов и атрибутов, служащая для обеспечения их уникальности в XML-документе.
- Все имена элементов в пределах пространства имен должны быть уникальны.
- XML-документ может содержать одинаковые имена элементов и атрибутов из нескольких пространств имен.
- Пример:

```
<customer>
    <id>105</id>
</customer>
<product>
    <id>p1</id>
</product>
```

У нас будут два одинаковых имени элемента несущих разную смысловую нагрузку и так будет, пока не введем пространства имен для их различения.

Объявление пространства имен

- Пространства имен объявляются с помощью XML атрибута **xmlns**, значение которого должно быть ссылкой URI
  - Например: `xmlns="http://www.w3.org/1999/xhtml"`
  - *Примечание*: URI в действительности не читается как адрес в сети, это просто обрабатываемая XML парсером строка.
- В объявление включается короткий префикс, который однозначно идентифицирует пространство имен каждого элемента, например:

```
<root xmlns:ns1="http://my.org/ns1" xmlns:ns2="http://my.org/ns2">
 <customer>
     <ns1:id>105</ns1:id>
 </customer>
 <product>
     <ns2:id>p1</ns2:id>
 </product>
</root>
```