

XML

eXtensible Markup Language

1. Пространства имён
(**Namespaces**)
2. Язык навигации внутри
XML-документа (**XPath**)

Пространства имён (Namespaces)

- ◆ Представьте, что в одном XML-документе определены такие элементы:
 - ◆ **name, age, company, position**
- ◆ А в другом такие:
 - ◆ **name, border, size, position**
- ◆ Что случится, если нам понадобится объединить данные из этих документов?
Парсер запутается в элементах **name** и **position**.
К какому типу их отнести?

Пространства имён (Namespaces)

- ◆ **Пространства имён** позволяют разделять наборы элементов, относящихся к разным объектам. Например:
 - **name, age, company, position** можно отнести к пространству имён «**people**»,
 - а **a name, border, size, position** — к пространству имён «**image**»
 - ◆ Пространства имён задаются ссылками **URI** (Uniform Resource Identifier).
 - Например, URI может иметь вид URL (гиперссылки):
<http://www.philol.msu.ru/people>
- NB! В данном случае URL — лишь **форма** задания уникального имени. По такому адресу в сети может ничего не существовать.

Пространства имён (Namespaces)

- ◆ Пространства имён, используемые в документе, должны быть **объявлены**
 - объявление можно сделать в корневом элементе или в том элементе, где данное пространство имён используется
 - каждому пространству имён (URI) даётся краткое обозначение
- ```
<root xmlns:pers="http://www.philol.msu.ru/people"
 xmlns:image="http://www.philol.msu.ru/photo">
```
- ◆ Это краткое обозначение добавляется к имени элемента как префикс (отделяется двоеточием)
  - ◆ У каждого элемента может быть только один префикс
    - ◆ **pers:name, pers:age, pers:company, pers:position**
    - ◆ **image:name, image:border, image:size, image:position**

# XML

## eXtensible Markup Language

1. Пространства имён  
(**Namespaces**)
2. Язык навигации внутри  
XML-документа (**XPath**)

# XPath — язык

## для навигации внутри документа

- ◆ Назначение XPath:  
выбирать из документа узлы или значения, которые нужны для совершения с ними каких-л. операций (напечатать, подсчитать, преобразовать...)
- ◆ Основные понятия:
  - узел (node) документа
    - ◆ узлами являются не только элементы, но и их атрибуты (attribute node), текстовое содержимое (text node) и т. д.
  - атомарное значение (atomic value)
    - ◆ число, строка, дата или булево значение
  - последовательность (sequence)  
**Выражение на XPath (путь) выбирает в документе последовательность узлов или атомарных значений, отвечающих заданным условиям**
    - ◆ Последовательность может состоять из 0 или 1 члена

# XPath — навигация внутри документа

- ◆ Основные понятия:
  - корень документа
  - элементы
  - атрибуты
  - абсолютный и относительный путь
  - родитель-дети
  - предки-потомки
  - сиблинги (узлы одного уровня)

# XPath — навигация внутри документа

## ◆ Простые выражения

- путь от корня (абсолютный) начинается с /
- фрагмент пути между двумя / — шаг по дереву (по умолчанию — от корня к листьям)
- в результат входят **все узлы, подходящие под описанный путь**

`/booklist ; /booklist/book/author`

- путь без / (относительный) считается не от корня, а от текущей позиции

`book/author`

- // любое количество шагов

`//book ; //author ; /booklist//author`

# XPath — навигация внутри документа

## ◆ Простые выражения

- `.` (точка) Текущий узел
- `..` (две точки) Родительский узел  
`../title` (путь от `book/author` к `book/title`)
- `*` Узел с любым именем  
`/booklist/*` все дети  
`/booklist/*/*` все внуки
- `@` Атрибут  
`//book/@lang` атрибут по имени `lang`  
`//book/@*` все атрибуты
- `node()` Любой узел
- `text()` Узел с текстовым содержимым

# XPath — навигация внутри документа

- ◆ Выражения с условием (предикатом)
  - условие на номер узла в последовательности  
`book[2]` ; `book[last()]` ; `book[position()<3]`
  - условие на значение дочерних элементов (путь отсчитывается от текущей позиции)  
`book[city="Москва"]`  
`book[price>250]`
  - условие на значение атрибутов  
`//book[@lang="rus"]/title`
  - Условие `[...]` применяется к тому узлу, после которого стоит  
`book[2]/author` vs. `book/author[2]`

# XPath — навигация внутри документа

- ◆ Оси (axes, ед.ч. axis)
  - Оси задают «направление движения» в конкретном шаге
  - В полном виде шаг выглядит так:
    - ... /**имя\_оси**::**узел**[**условие**]/ ...
    - ... /**sibling**::**book**[**25**]/ ...
  - Ось по умолчанию — **child**
    - /booklist/**child**::**book** = /booklist/book

# XPath — навигация внутри документа

## ◆ Оси (axes)

- child
- parent
- sibling
- self
- ancestor
- ancestor-or-self
- descendant
- descendant-or-self
- preceding
- following
- preceding-sibling
- following-sibling
- attribute
- namespace

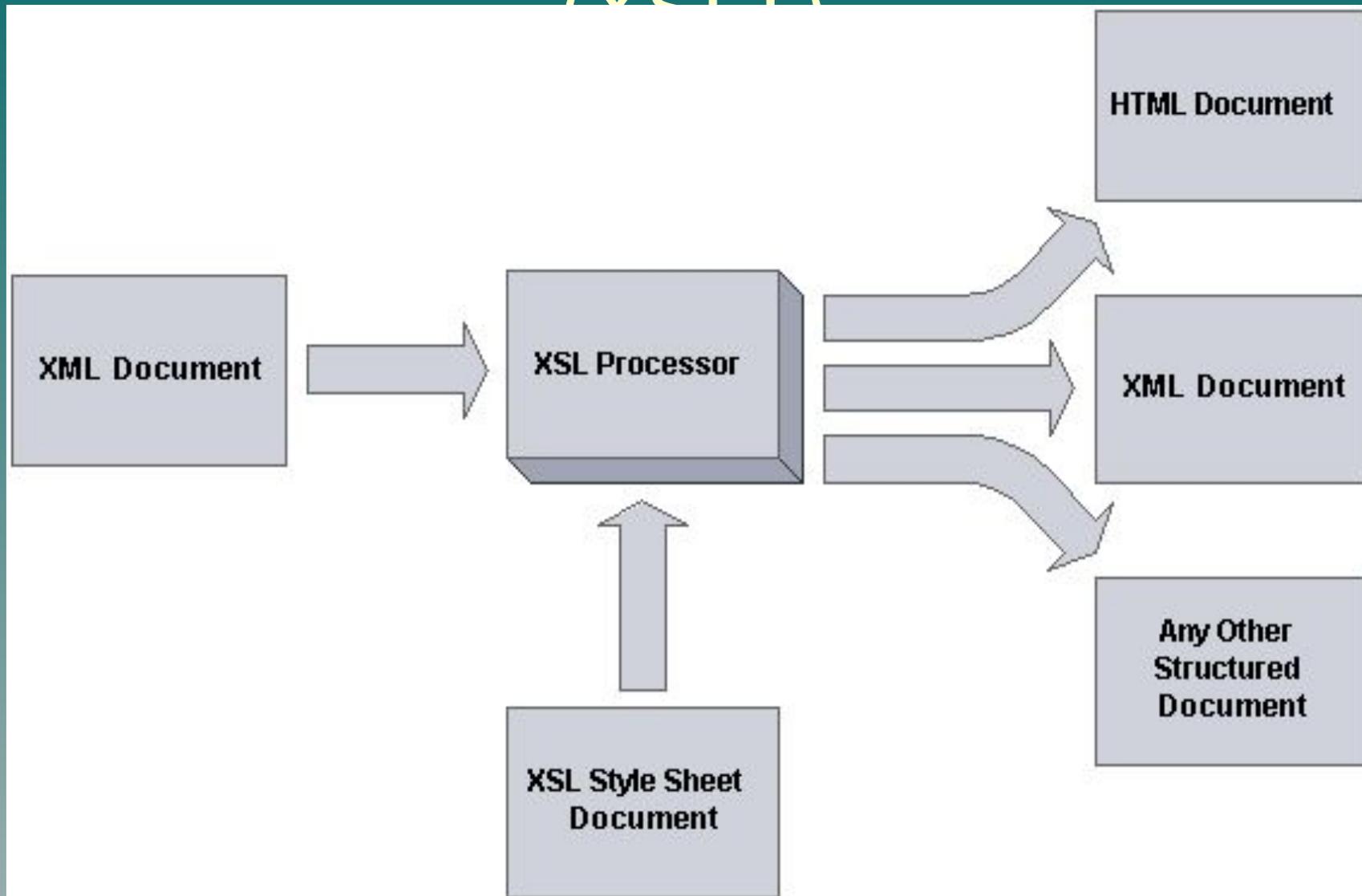
# XPath — навигация внутри документа

- ◆ Запишите полностью выражения
  - /books//author
  - //book[3]/\*
  - ..[name()='author']/@lang
  - ..[@lang='rus']/title
  - ./@\*
  - /books/\*/city

# Преобразования XML-данных (XSLT)

- ◆ Как мы помним, XML ничего сам не делает. Его задача — **описывать** структуру данных
- ◆ Чтобы с этими данными что-то сделать, используются специальные средства
- ◆ XSL — **eXtensible Stylesheet Language**
  - **XSLT: XSL Transformations**  
XSLT выполняет преобразования данных
  - **XSL-FO: XSL Formatting Objects**  
XSL-FO форматирует данные для печати

# Преобразования XML-данных (XSLT)



# Преобразования XML-данных (XSLT)

- ◆ Что умеет делать XSL?
  - Отбирать (фильтровать) определённые данные из целого документа
  - Упорядочивать данные независимо от исходного порядка
  - Менять исходную структуру данных (до неузнаваемости)
  - Преобразовывать XML
    - ◆ в другой XML
    - ◆ в правильный HTML
    - ◆ в другие текстовые форматы

# Преобразования XML-данных (XSLT)

- ◆ В отличие от многих языков программирования (BASIC, Pascal, C,...), **XSL** — не **процедурный** язык, а **декларативный**.

Программа на XSL (*transformation*, она же *stylesheet*) сообщает не что нужно делать (последовательность операций), а **что должно получиться**.

# Преобразования XML-данных (XSLT)

- ◆ Программа на XSL состоит из **шаблонов** (templates), которые сопоставляются с различными частями дерева XML на входе и выдают некоторый выход.
- ◆ (Далее см. примеры + учебник)