

Введение в XML

eXtensible Markup Language, 1996

Для XML-совместимого документа соблюдаются следующие правила:

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<a>...` или `<a ... />`
3. `<a>...........`
4. `<a> ≠ <A>`
5. ``
6. корневой тег

Подключение XSLT осуществляется с помощью:

```
<?xml-stylesheet type="text/xsl" href="style.xslt"?>
```

Зачем нужен XML

- XML отделяет данные от HTML
 - С XML данные можно хранить в отдельных файлах XML. При этом HTML/CSS используется для отображения и шаблонизации, и поступающие XML-данные не требуют каких-либо изменений в коде HTML документа.
- XML упрощает распределение и передачу данных
 - XML данные хранятся в простом текстовом формате. Это обеспечивает программную и аппаратную независимость. Также легко решается проблема чтения данных на несовместимых системах
- XML делает данные более доступными
 - XML-данные становятся доступными для машинной обработки, в частности, для установления и анализа связей между различными сущностями, описанными с помощью XML.
- XML используется для создания новых интернет-языков
 - XHTML
 - WSDL для описания доступных веб-сервисов
 - **RSS языки для новостных каналов, в том числе для Турбо-страниц Яндекса**
 - RDF и OWL для описания ресурсов и онтологии

Применение XML: RSS и Atom

- Из множества конкурирующих стандартов до наших дней благополучно дожили лишь два: **RSS и Atom**.
- RSS является действительно "народным" форматом, практически синонимичным для многих самому понятию синдикация. И RSS, и Atom - основанные на XML форматы, предназначенные для импорта\экспорта информации, то есть *агрегирования*.
- Эти стандарты, как, впрочем, и все основанные на XML, реализуют давнюю **идею разделения оформления и содержания** и лишены одного из главных недостатков HTML - запутанности и перегруженности служебными и "оформительскими" элементами, которые сильно затрудняют автоматическое унифицированное получение и обработку информации: в данном случае транслируется только полезное содержание без каких-либо элементов дизайна, навигации итд.

Разные определения RSS

- Rich Site Summary (RSS 0.9x) — обогащённая сводка сайта;
- RDF Site Summary (RSS 0.9 и 1.0) — сводка сайта с применением инфраструктуры описания ресурсов;
- Really Simple Syndication (RSS 2.x) — **очень простое приобретение информации.**

RSS — семейство XML-форматов, предназначенных для **описания лент новостей, анонсов статей, изменений в блогах** и т. п. Информация из различных источников, представленная в формате RSS, может быть собрана, обработана и представлена пользователю в удобном для него виде специальными программами-агрегаторами.

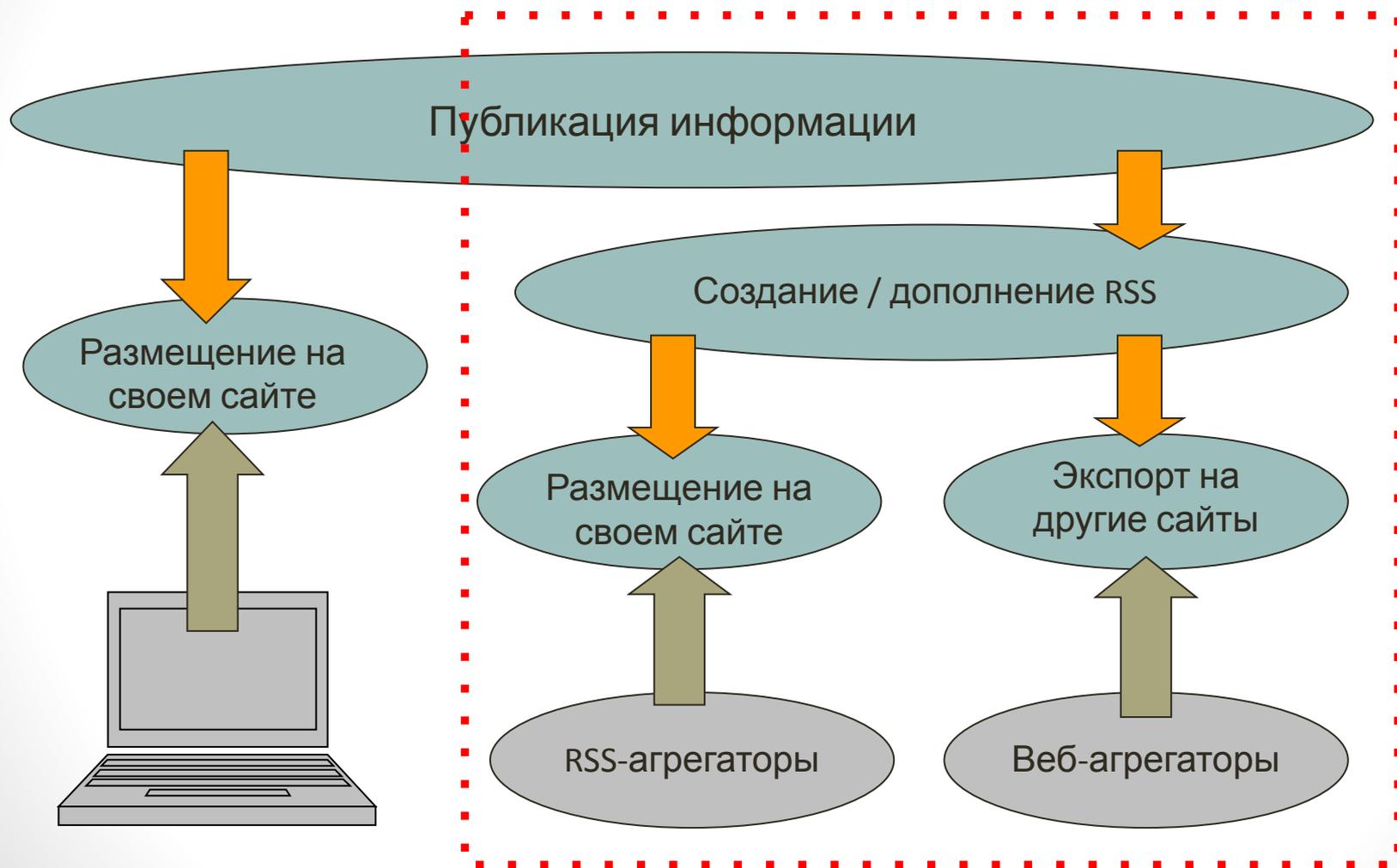
Назначение

С помощью RSS даётся краткое описание новой информации, появившейся на сайте, и ссылка на её полную версию. **Интернет-ресурс в формате RSS называется RSS-каналом, RSS-лентой или RSS-фидом.**

С помощью RSS можно экспортировать новости своего портала на другие информационные ресурсы.



Логика работы RSS-канала



Пример RSS v2.0

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/">
<channel>
  <title>XML.com</title>
  <link>http://www.xml.com/</link>
  <description>XML.com features a rich mix of information and services for the
XML community.</description>
  <language>en-us</language>

  <item>
    <title>Normalizing XML, Part 2</title>
    <link>http://www.xml.com/pub/a/2002/12/04/normalizing.html</link>
    <description>In this second and final look at applying relational
normalization techniques to W3C XML Schema data modeling, Will Provost
discusses when not to normalize, the scope of uniqueness and the fourth
and fifth normal forms.</description>
    <dc:creator>Will Provost</dc:creator>
    <dc:date>2002-12-04</dc:date>
  </item>
</channel>
</rss>
```

Введение в XSLT

eXtensible Stylesheet Language Transformations

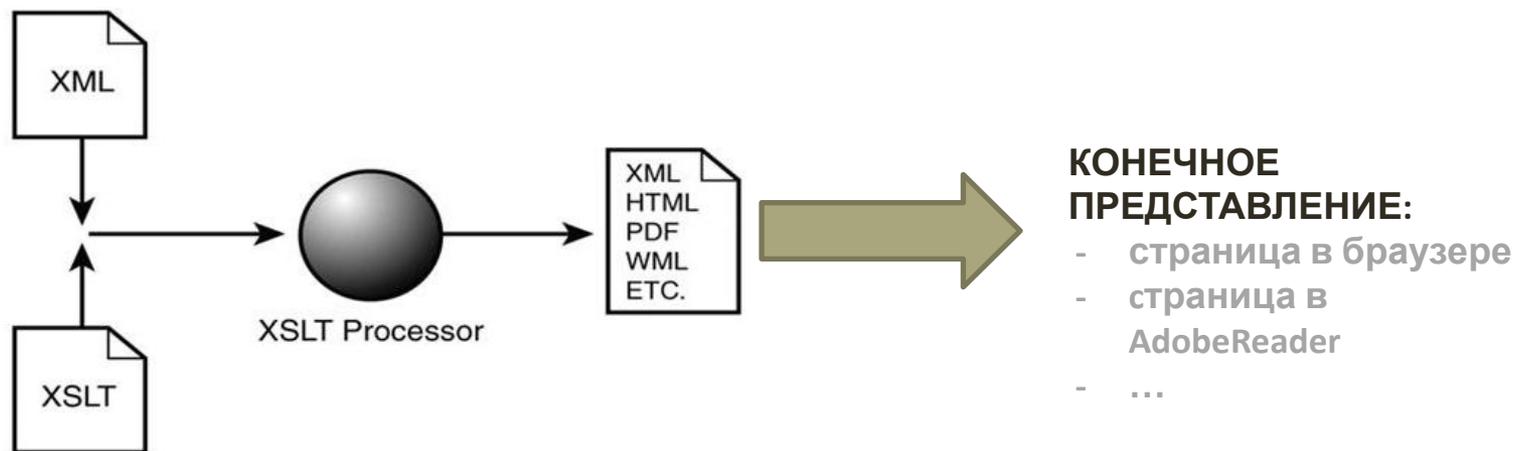
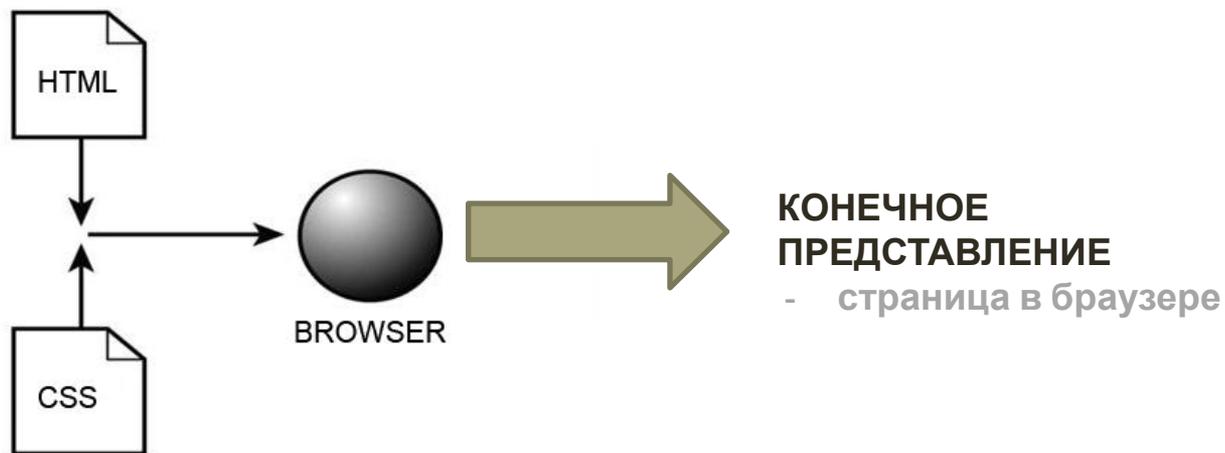
Простейший шаблон:

```
<?xml version="1.0" encoding="utf-8"?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  <xsl:output method="html"/>  
  <xsl:template match="/">  
    ...разбор XML...  
  </xsl:template>  
</xsl:stylesheet>
```

Некоторые XSL-инструкции:

- `<xsl:value-of select="имятэга"/>`
- `<xsl:for-each select="имятега"> ...тело цикла... </xsl:for-each>`
- `<xsl:if test="условие">...если условие выполняется...</xsl:if>`
- `<xsl:variable name="имя">значение</xsl:variable>`

СВЯЗЬ XML и XSLT



Пример

index.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl"
href="style.xml"?>
<garage>
  <auto>
    <brand>Lada</brand>
    <model>Kalina</model>
  </auto>
  <auto>
    <brand>Volvo</brand>
    <model>S40</model>
  </auto>
</garage>
```

style.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/
Transform">
<xsl:output method="html"/>
<xsl:template match="/">
<html>
<head><title>XSL sample</title></head>
<body>
<xsl:for-each select="/garage/auto">
  <h1><xsl:value-of select="brand"/></h1>
  <xsl:if test="brand='Volvo'">
    <p><xsl:value-of select="model"/></p>
  </xsl:if>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Введение в JSON

JavaScript Object Notation, 1999

- Происхождение от JavaScript
- Формат может использоваться практически с любым языком программирования
- За счёт своей лаконичности по сравнению с XML может быть более подходящим для сложных структур

Применяется в веб-приложениях как для обмена данными между браузером и сервером (AJAX), так и между серверами (программные HTTP-сопряжения).

Структура JSON

1. **Набор пар *ключ: значение***
2. **Ключ** - только регистрозависимая строка
3. **Значения:**
 - **Запись** — неупорядоченное множество пар **ключ: значение**, заключённое в фигурные скобки «{ }»
 - **Массив** — упорядоченное множество **значений**. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми
 - **Число**
 - **Литералы** *true*, *false* и *null*
 - **Строка** — упорядоченное множество символов юникода, заключённое в двойные кавычки.

XML vs JSON

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<garage>
  <auto>
    <brand>Lada</brand>
    <model>Kalina</model>
  </auto>
  <auto>
    <brand>Volvo</brand>
    <model>S40</model>
  </auto>
</garage>
```

JSON:

```
{
  "garage":
    {
      "auto":
        [
          {
            "brand": "Lada",
            "model": "Kalina"
          },
          {
            "brand": "Volvo",
            "model": "S40"
          }
        ]
    }
}
```

AJAX

Asynchronous JavaScript And XML

«Фоновый» обмен данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

- Пользователь заходит на веб-страницу и нажимает на какой-нибудь её элемент (либо другое событие).
- JavaScript определяет, какая информация необходима для обновления страницы, формирует и отправляет соответствующий запрос на сервер.
- Сервер возвращает данные на которую пришёл запрос, и JavaScript вносит изменения на страницу с учётом полученной информации без полной перезагрузки страницы.