


---

# Отображение XML в браузере

---



Лекция

# XSL

Для отображения документов XML могут быть использованы язык XSL или технология CSS.

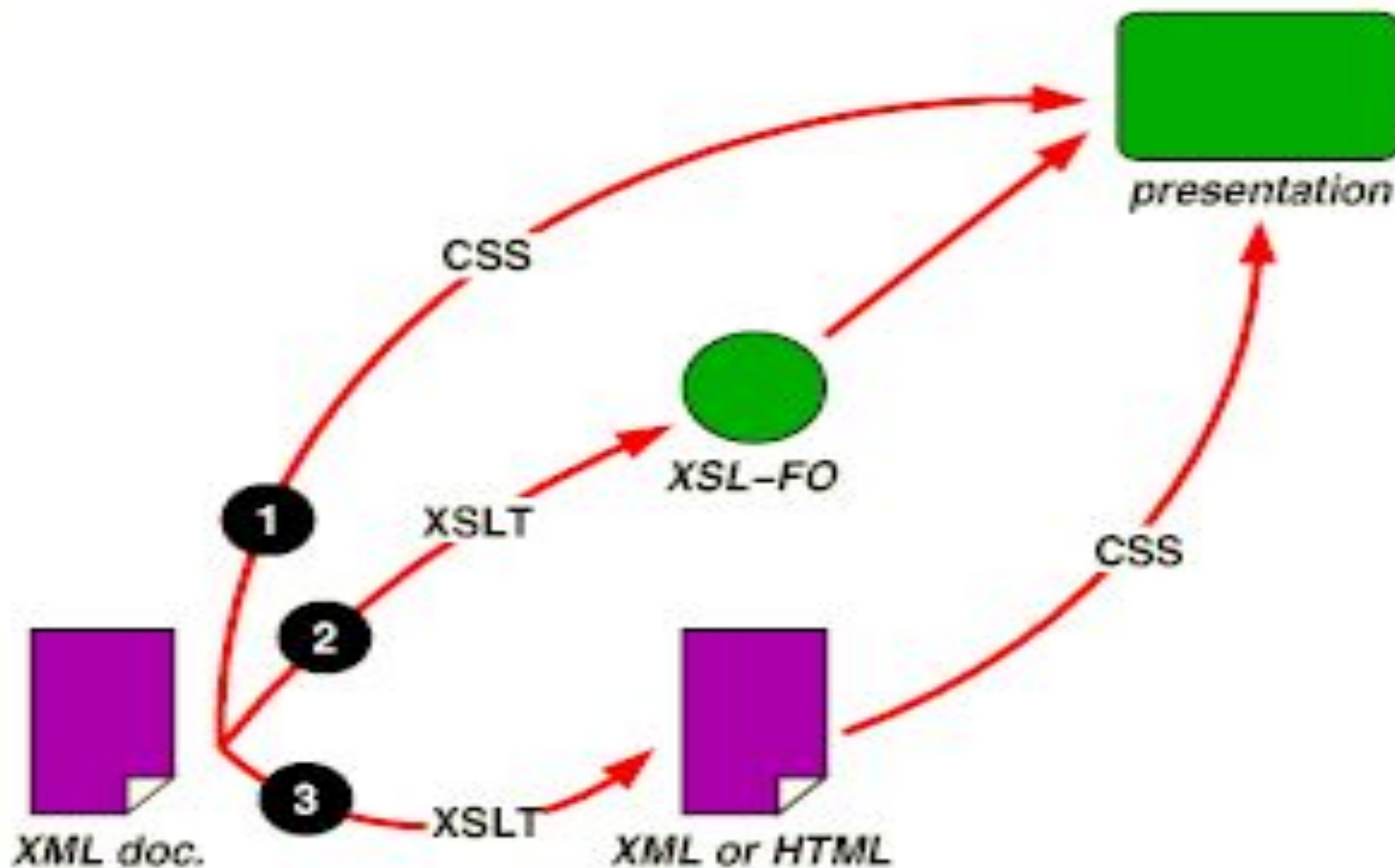
**XSL** (Extensible Stylesheet Language) - это набор определенных стандартом инструкций, которые могут быть применены к документу XML.

XSL состоит из двух независимых частей: язык для преобразования XML - XSL-T (XSL Transformations) и язык для форматирования (вёрстки) XML - XSL-FO (XSL Formatting Objects).

С помощью XSL можно трансформировать XML документ в любой вид, например HTML, WML, RTF, PDF, SQL, SWF (flash запускаемый модуль), а так же в XML и XSL. (Примером преобразования данных XML во Flash является сайт [www.panasonic.ru](http://www.panasonic.ru))

Самой распространенной XSL-T машиной на данный момент является MSXML.

# Способы отображения XML в браузере



# Использование CSS

**Подключение внешнего файла CSS к документу XML:**

```
<?xml-stylesheet href="URL" type="text/css"?>
```

Пример:

**common.css**

```
article, headline, author { display: block }
```

**classic.css**

```
article { font-family: serif; background: white; color: #003 }
```

```
author { font-size: large }
```

```
headline { font-size: x-large }
```

**article.xml**

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<?xml-stylesheet href="common.css" type="text/css"?>
```

```
<?xml-stylesheet href="classic.css" title="Classic" media="screen, print"  
type="text/css"?>
```

```
<article>
```

```
  <headline>Библиотечные системы</headline>
```

```
  <author>Дж. Солтон</author>
```

```
</article>
```

# Использование CSS

**Таблица CSS может быть встроена и внутрь документа:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="#style" type="text/css"?>
<article>
  <S id="style">
    article, headline, author { display: block }
    S { display: none }
  </S>
  <headline>Библиотечные системы</headline>
  ...
</article>
```

# Использование XSL

## Подключение внешнего файла XSL к документу XML:

```
<?xml-stylesheet href="URL" type="text/xsl"?>
```

## Структура файла XSL

```
<?xml version="1.0"?>
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
...
```

```
</xsl:stylesheet>
```

## XSLT

### Формат записи шаблона:

```
<xsl:template match="имя_узла">
```

```
...
```

```
</xsl:template>
```

### Шаблоны по отношению к дочерним элементам узла:

```
<xsl:apply-templates/>
```

### Получение значений узлов:

```
<xsl:value-of select="имя_узла"/>
```

# Использование XSLT

Пример:

Преобразуем XML-документ в HTML с использованием инструкций XSLT.

## **ex1.xml**

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="ex1.xsl"?>
<planets>
  <planet>
    <name> Земля </name>
    <radius units="мили"> 2107 </radius>
  </planet>

  <planet>
    <name> Марс </name>
    <radius units="мили"> 1117 </radius>
  </planet>
</planets>
```

# Использование XSLT

## ex1.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="planets">
    <html><xsl:apply-templates/></html>
  </xsl:template>
  <xsl:template match="planet">
    <p><xsl:value-of select="name"/></p>
  </xsl:template>
</xsl:stylesheet>
```

## Получаем HTML-документ:

```
<html>
  <p>Земля</p>
  <p>Марс</p>
</html>
```



# Использование XSLT

## Пример

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="ex1.xsl"?>
<planets>
  <planet>
    <name> Земля </name>
    <name> Синяя планета </name>
    <radius units="мили"> 2107 </radius>
  </planet>

  <planet>
    <name> Марс </name>
    <name> Красная планета </name>
    <radius units="мили"> 1117 </radius>
  </planet>
</planets>
```

# Использование XSLT

Для выбора всех элементов <name> используется тег **xsl:for-each**:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="planets">
    <html>
      <xsl:apply-templates/>
    </html>
  </xsl:template>
<xsl:template match="planet">
  <xsl:for-each select="name">
    <p><xsl:value-of select="."/></p>
  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

# Использование XSLT

Если необходимо выполнить обработку только тех элементов XML-документа, которые являются потомками других элементов, то в этом случае **имена элементов задаваемые в атрибутах match и select** имеют вид:

"имя1//имя2" – обработать элементы являющиеся потомками (например, "planet//name")

"имя1/имя2" – обработать элементы являющиеся прямыми потомками (например, "planet/name")

**Доступ к атрибутам элементов выполняется через символ @.**

## Пример

Название	Радиус
Земля	2107мили
Марс	1117мили

# Использование XSLT

необходимо к файлу **ex1.xml** применить XSLT-преобразование:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/planets">
    <HTML>
      <head><title>Планеты</title></head>
      <body>
        <table border="1">
          <tr><td>Название</td>
          <td>Радиус</td></tr>
          <xsl:apply-templates/>
        </table>
      </body>
    </html>
  </xsl:template>
```

# Использование XSLT

```
<xsl:template match="planet">
  <tr>
    <!-- Отображается название планеты -->
      <td><xsl:value-of select="name"/></td>
    <!-- Вызывается обработка тэга radius -->
      <td><xsl:apply-templates select="radius"/></td>
  </tr>
</xsl:template>

  <xsl:template match="radius">
    <!-- Отображается значение радиуса -->
      <xsl:value-of select="."/>
    <!-- Отображается символ пробела -->
      <xsl:text> </xsl:text>
    <!-- Отображается единицы измерения радиуса -->
      <xsl:value-of select="@units"/>
  </xsl:template>

</xsl:stylesheet>
```

# Использование XSLT

Можно ограничить количество элементов, отвечающих шаблону, **введя фильтр** - выражение, заключенное в квадратные скобки ([]) и следующее непосредственно за оператором пути.

```
match="BOOK[PRICE='100']"
```

Если в фильтр включено только имя элемента, то соответствующий элемент должен иметь дочерний элемент с указанным именем.

```
match="ITEM[CD]"
```

## Сортировка данных XML

```
<xsl:for-each select="BOOKS/BOOK"  
order-by="+AUTHOR/LASTNAME;+AUTHOR/FIRSTNAME">
```

```
<xsl:apply-templates select="BOOKS/BOOK" order-by="-TITLE">
```

# Использование XSL-FO


Интерпретатор, например, является Apache XML.

## ex2.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format" version="1.0">
  <xsl:template match="planets">
    <fo:root>

      <fo:layout-master-set>
        <fo:simple-page-master master-name="page" page-width="300mm">
          <fo:region-body margin-top="0mm" margin-bottom="10mm"/>
        </fo:simple-page-master>
      </fo:layout-master-set>


      <fo:page-sequence master-reference="page">
        <fo:flow flow-name="xsl-region-body">
          <xsl:apply-templates/>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
  </xsl:template>
```



---

# Отображение XML в браузере

---



Раздел:  
Технологии создания сайтов