

# **Вывод документов XML с использованием CSS и XSL**

# Цели занятия

- Применение таблиц стилей типа CSS в документах XML
- Преобразование документов XML с помощью таблиц стилей типа XSL
- Применение шаблонов
- Переключение стилей в документе

# Таблицы стилей – 1



Документ XML может быть выведен в различных форматах на различные устройства вывода, такие, как компьютер, принтер и т.д.

# Таблицы стилей – 2

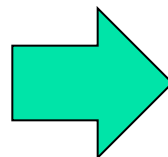
- Таблица стилей – это набор инструкций для вывода документов



Данные

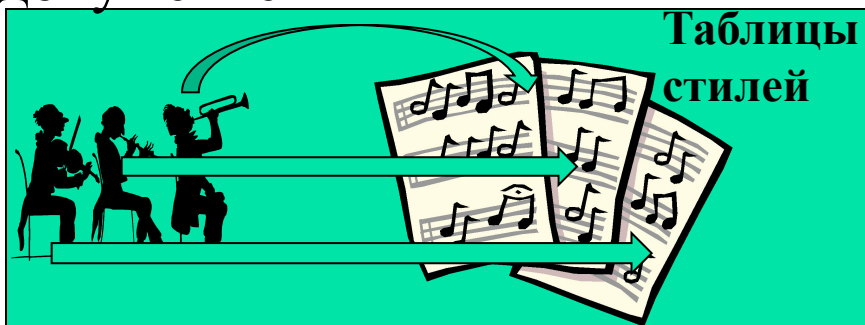


Уровень



представления

Отделяет уровень представления от данных, содержащихся в документе

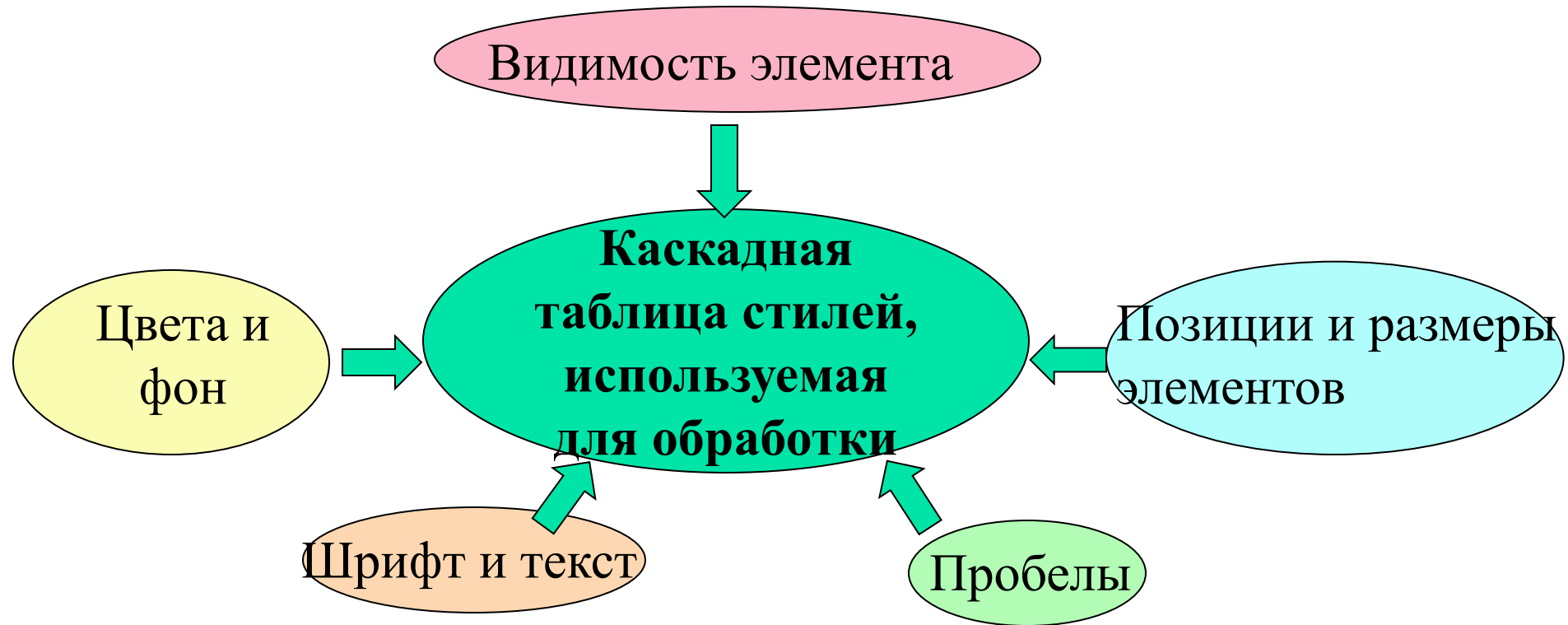


Один документ XML  
может иметь несколько  
таблиц стилей

# Таблицы стилей – 3

- Некоторые таблицы стилей, предлагаемые на рынке:
  - ◆ CSS – Каскадные таблицы стилей
  - ◆ XSL – Расширяемые таблицы стилей
  - ◆ DSSL – Document Style Semantics and Specification Language (Язык семантических стилей и спецификаций документа)

# Использование CSS и XSL

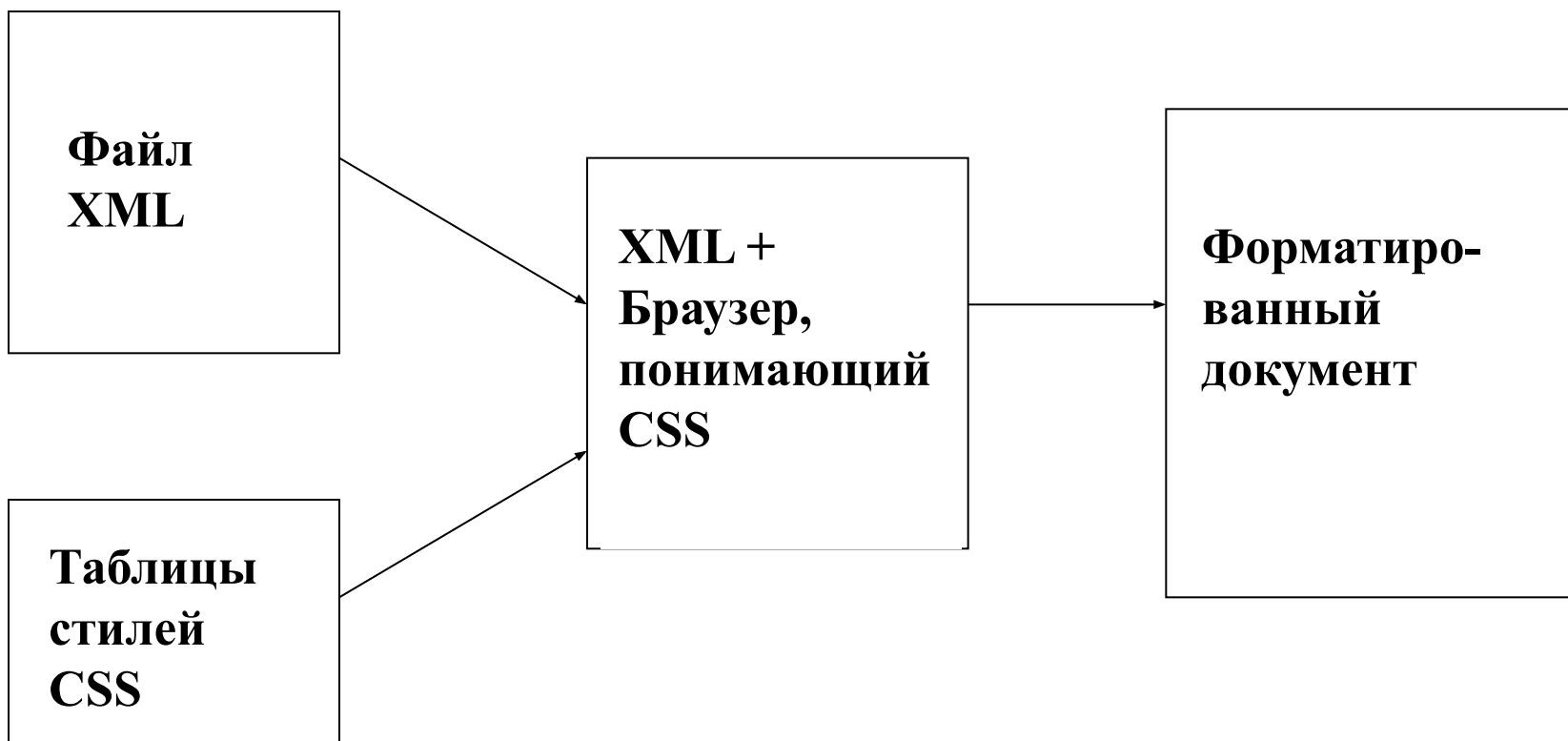


XSL используется для преобразования документов, написанных на одном XML DTD, в другое DTD и для форматирования

# XML и таблицы стилей

- Документы XML – это обычные текстовые файлы
- Таблицы стилей используются для форматирования и просмотра документа XML
- Для XML широко используются два типа таблиц стилей :
  - ◆ Cascading Style Sheets (CSS) – Каскадные таблицы стилей – расширение языка HTML
  - ◆ Extensible Style sheet Language (XSL) – Расширяемый язык таблиц стилей – специальный язык стилей для XML

# Работа с языками форматирования в XML





# Вывод документа XML с помощью CSS – 1

- Документ XML

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

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

```
<xsampdoc>
```

```
  <greeting>Hello!</greeting>
```

```
  <question>How were you!</question>
```

```
  <answer>
```

```
    <ans>Fine</ans>
```

```
    <ans>Thank you.</ans>
```

```
  </answer>
```

```
</xsampdoc>
```

# Вывод документа XML с помощью CSS – 2

## Документ CSS для документа XML

```
xsampdoc
{ margin-top:.7in;
margin-bottom: .7in;
margin-left:1.5in;
margin-right:1in;
color: navy;
background-color:white;
display: block }
greeting
{ display:block;
font-family: Arial, Helvetica, sans-serif;font-size: 32pt;
width: 30em;
color: red }
```

```
question
{ display:block;
font-size: x-large;
color: black; }
answer.ans
{ display: block;
font-size: 20pt;
color: blue
}
```

# Правила стиля CSS

- Синтаксис для правил стилей в каскадных таблицах стилей:  
`Selector { declaration }`
- Селектор (selector) идентифицирует тэг, к которому применяется заданный стиль
- Объявление (declaration) представляет правила стилей, применяемые к данному селектору
- Такой селектор обозначается, как "Простой селектор" ('Simple Selector')

## Пример:

```
greeting
{display:block;font-family: Arial, Helvetica,sans-serif;
  font-size: 32pt;    width: 30em;    color:  red }
```

# Множественный селектор

- Используется для применения одного правила стиля к различным элементам
- Синтаксис:

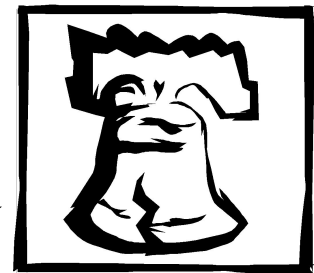
```
selector, selector..... {declaration}
```

Правило стиля

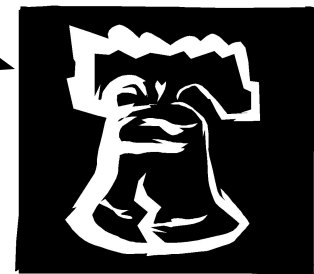


Одно Правило стиля  
использует и Селектор1,  
и Селектор2

Селектор 1



Селектор 2



**Пример:**

```
greeting, question  
{font-family: sans-serif;  
font-size: 32pt; }
```

# Контекстные селекторы – 1

- Помогает отличать друг от друга различные вхождения одного тэга

tag (тэг)



Контекстный селектор

Вхождение 1

tag([declaration])

Вхождение 2

tag([declaration])



Вхождение 1



Вхождение 2

# Контекстные селекторы – 2

- **Пример:**

```
<xsampdoc>
  <greeting>Hello!
  <extension>there</extension></greeting>
  <answer>
    <ans>Fine<extension>here</extension></ans>
    <ans>Thank you.</ans>
  </answer>
</xsampdoc>
```

Контекстные селекторы позволяют отличать разные вхождения элемента `<extension>`

```
greeting.extension{ [declarations] }
```

```
answer.ans.extension{ [declaration] }
```

# СИМВОЛЫ, ИСПОЛЬЗУЕМЫЕ В CSS

Символ	Имя	Функция
:	Двоеточие	Разделяет свойство и его значение
;	Точка с запятой	Разделяет несколько комбинаций свойство/значение
.	Точка	Идентифицирует контекст селектора
,	Запятая	Разделяет несколько селекторов в правиле стиля

# Связывание CSS с XML

- В документе XML должны быть организованы ссылки на каскадные таблицы стилей
- Для этого мы используем инструкции обработки
- Синтаксис:

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

Пример:

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



# Свойства и значения

- Значением свойства стиля CSS может быть строка, число с единицей измерения, целое число или цветное значение
- Значения могут быть абсолютными или относительными, наследуемыми или не наследуемыми

## Пример абсолютного значения

```
P{margin-left:3cm;}
```

Создаётся поле в 3 сантиметра независимо от размера страницы

## Пример относительного значения

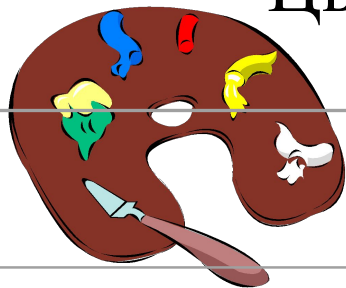
```
P{margin-left:10%}
```

Создаётся поле шириной 10% от общей ширины страницы

# Цветовые значения



## Цветовые значения, поддерживаемые в XML



	<b>Gray (Серый)</b>	<b>White (Белый)</b>	<b>Red (Красный)</b>
<b>Краткая форма</b>	#888	#FFF	#F00
<b>Десятичное целое</b>	rgb(136,136,136)	rgb(255,255,255)	rgb(255,0,0)
<b>Проценты</b>	rgb(55%,55%,55%)	rgb(100%,100%,100%)	rgb(100%,0,0)

# Форматирование текста

- Свойства CSS позволяют определить, какой шрифт следует использовать для вывода элемента, размер и цвет текста
- Ключевые слова `<absolute-size>` и `<relative-size>` используются в свойствах шрифта

## **`<absolute-size>`**

xx-small

x-small

small

medium

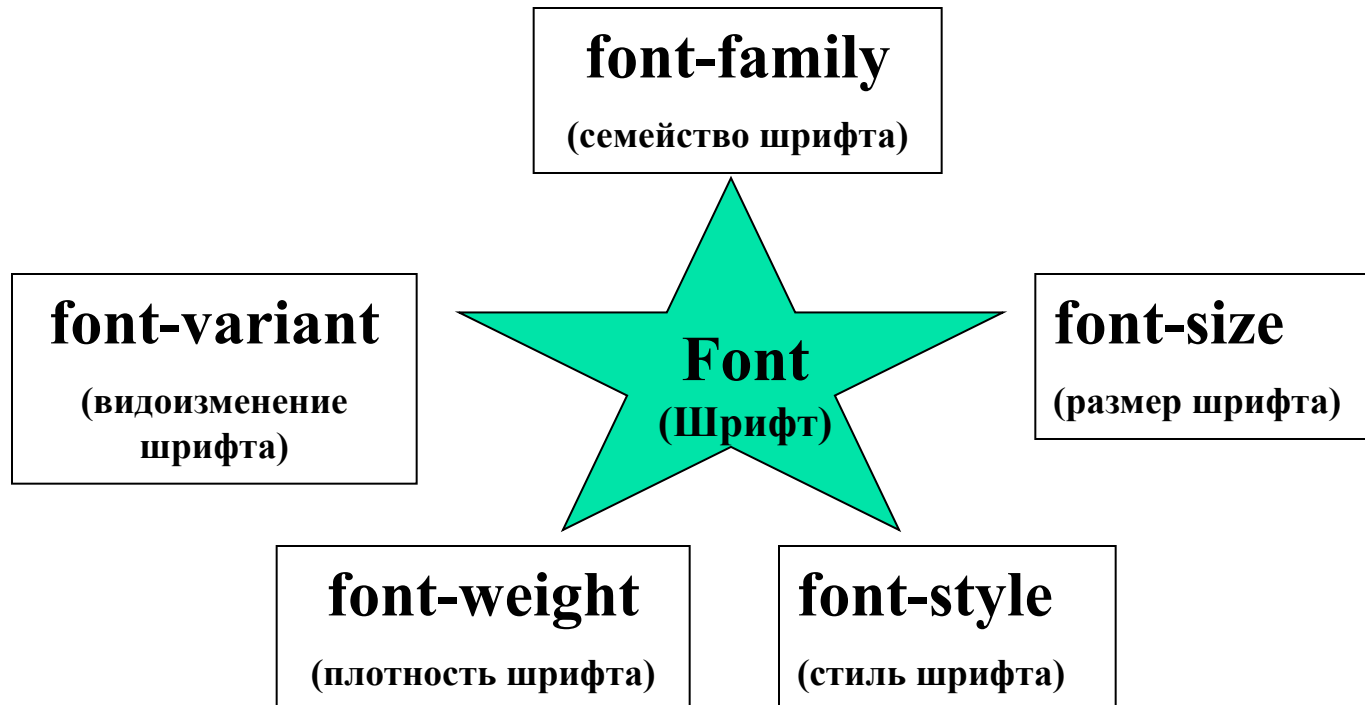
## **`<relative-size>`**

larger

smaller

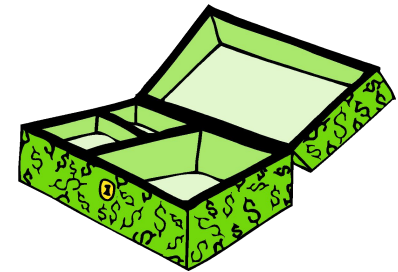
# Форматирование текста

- Различные свойства шрифта, которые могут быть установлены:

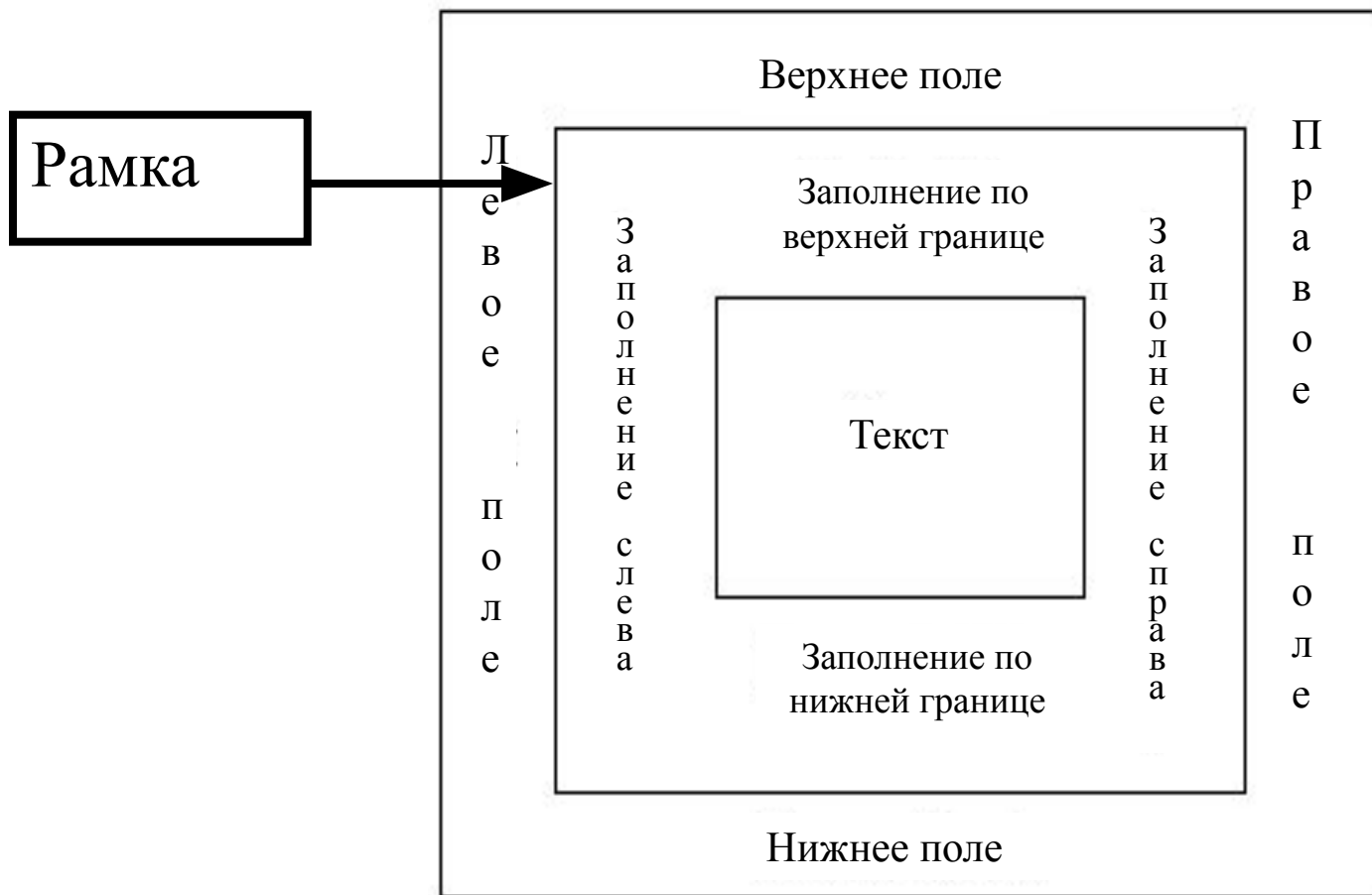


# Панели – 1

- Блоки текста могут быть помещены в панель
- Для панелей существуют три свойства:
  - ◆ Margins (Поля)
  - ◆ Border (Рамка)
  - ◆ Padding (Заполнение)
- Расстояние между содержимым панели и её границей называется заполнением (padding)
- Расстояние между рамкой и внешним краем смежной панели или между рамкой и содержащей её панелью называется её полями (margin)

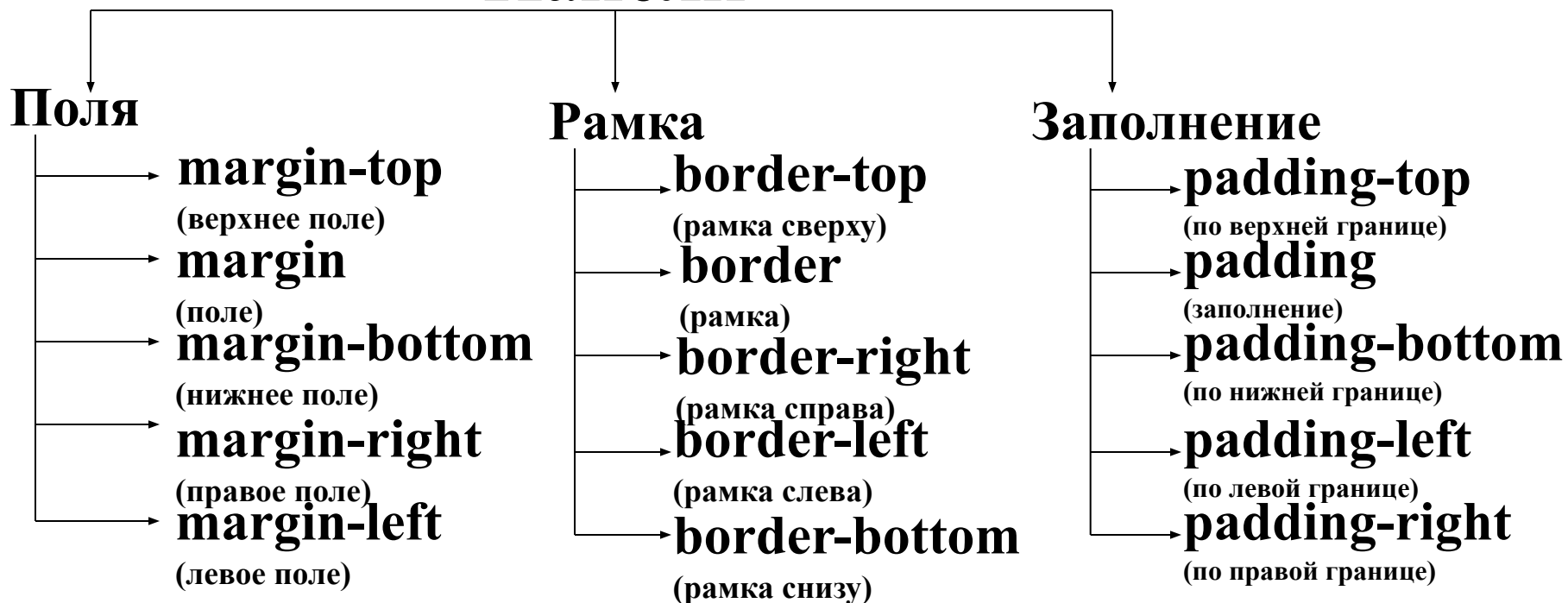


# Панели – 2



# Поля, рамка и заполнение

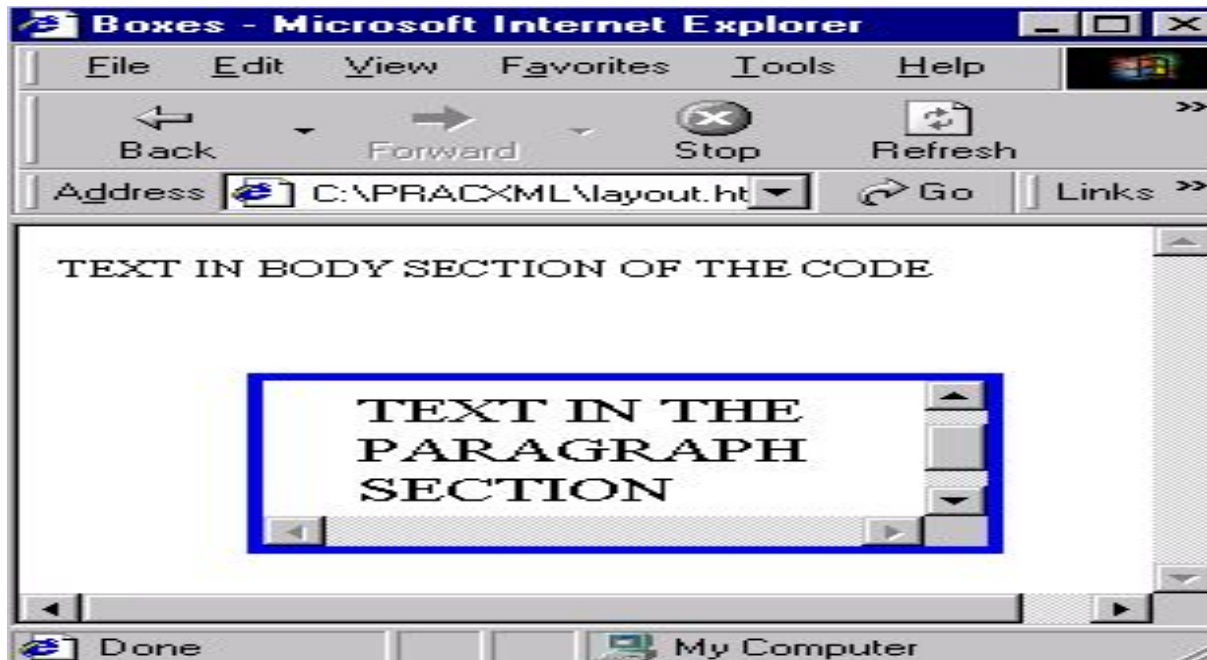
## Панели



- Свойство **padding** (заполнение) используется для определения расстояния или пространства между рамкой и содержимым панели.

# Управление макетом

- Свойства макета CSS могут управлять панелью на экране
- Панели могут перекрываться при использовании свойства z-index



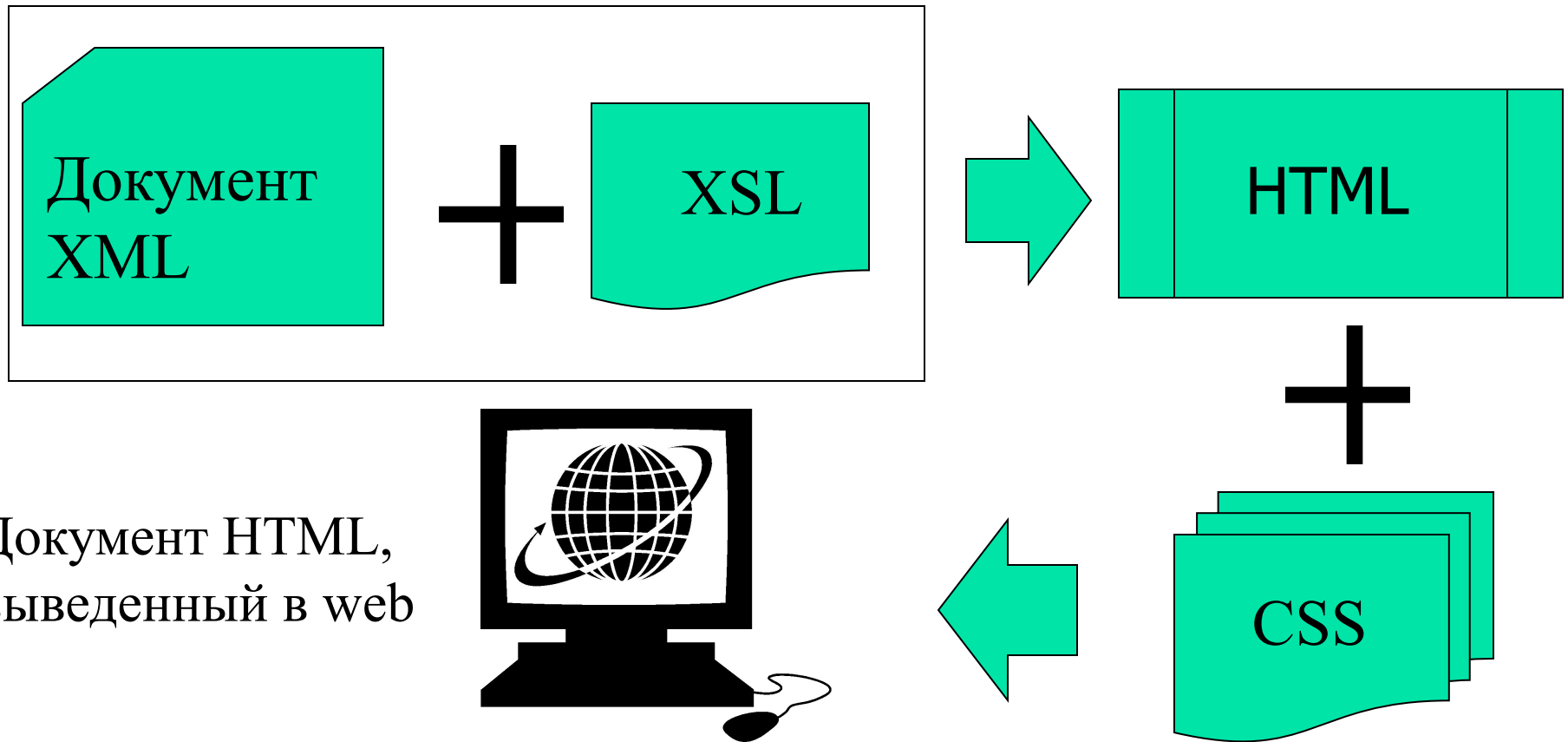
Выводится абзац, который имеет размеры два дюйма на один дюйм, а также полосу прокрутки.



# XSL

- XSL – это реализация таблиц стилей, созданная специально для языка XML.
- Функциональные возможности XSL:
  - ◆ Предоставляет язык преобразования (XSLT)
  - ◆ XSL может быть использован, как язык форматирования
  - ◆ XSL может быть использован для сортировки и фильтрации
  - ◆ XSL может быть использован для поиска совпадений образцов, что может помочь нам при поиске записей

# XSL-2



Документ HTML,  
выведенный в web

# Пример использования XSL

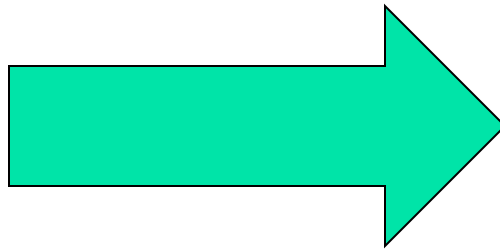
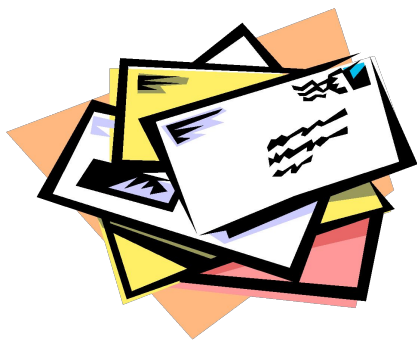
## Код таблицы стилей XSL

```
<html>
<head>
<title>Icons In Cricket</title>
</head>
<body>
<h2>Icons In Cricket</h2>
      <table border="2"
cellpadding="3">
<tr>
      <td>Player</td>
      <td>No. Of Catches</td>
      <td>No. Of 100's</td>
      <td>No. Of 50's</td>
</tr>
<xsl:for-each
  select="cricket/player" order-by
  ="-catches">
```

```
    <tr>
    <td><xsl:value-of
      select="name"/></td>
    <td><xsl:value-of
      select="no_50"/></td>
    <td><xsl:value-of
      select="no_100"/></td>
    <td><xsl:value-of
      select="catches"/></td>
    </tr>
  </xsl:for-each>
</table>
</body>
</html>
```

# Образцы (Patterns)

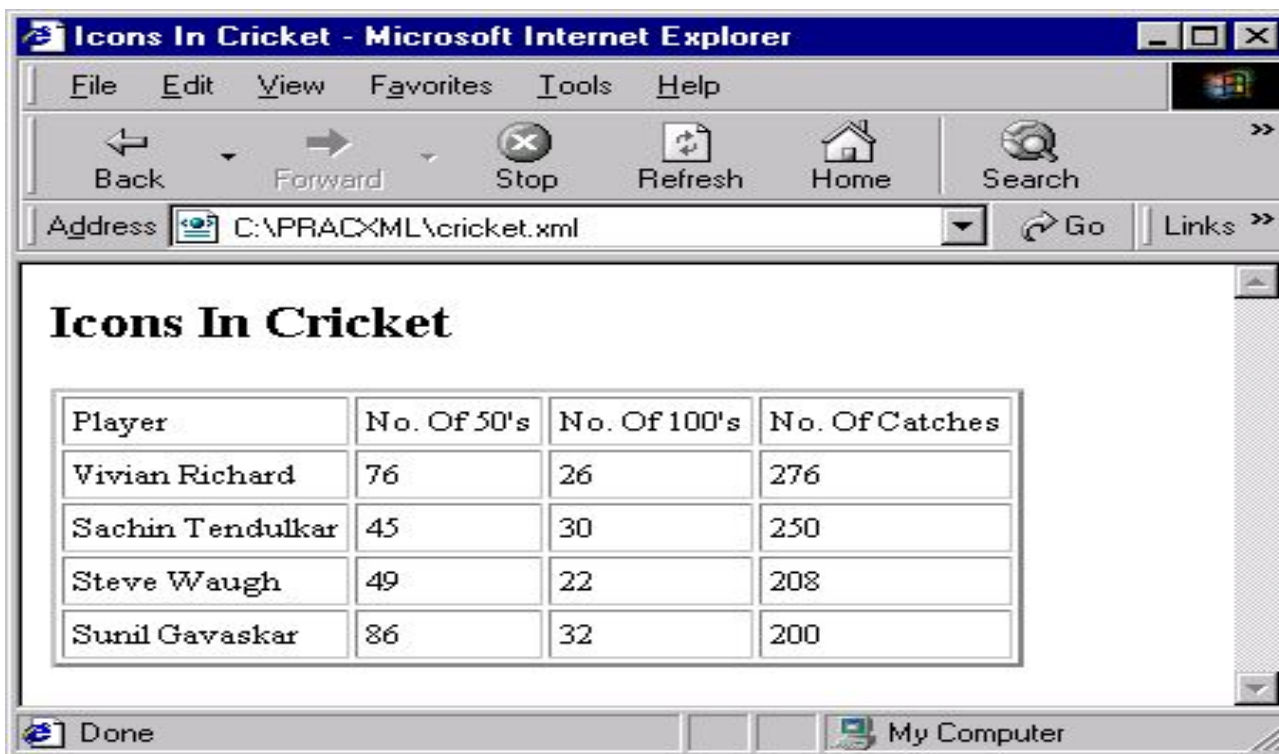
- Образцы, поддерживаемые в языке XSL:
  - ◆ Sorting (Сортировка)
  - ◆ Operators (Операторы)
  - ◆ Filtering (Фильтрация)



Письма  
после  
сортировки

# Сортировка

- По умолчанию сортировка выполняется в порядке возрастания
- Знак '-' используется для сортировки в порядке убывания



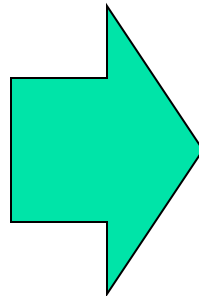
The screenshot shows a Microsoft Internet Explorer window titled "Icons In Cricket - Microsoft Internet Explorer". The address bar shows "C:\PRACXML\cricket.xml". The main content area displays a table with the following data:

Player	No. Of 50's	No. Of 100's	No. Of Catches
Vivian Richard	76	26	276
Sachin Tendulkar	45	30	250
Steve Waugh	49	22	208
Sunil Gavaskar	86	32	200

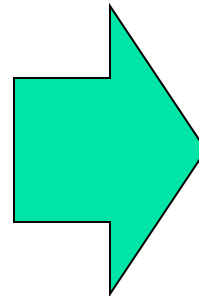
Таблица  
организована в  
порядке  
убывания  
количества  
перехватов

# Операторы

Различные типы операторов, которые поддерживаются в XSL:



/  
./  
//  
./.  
\*  
@  
=



**/\* /name**  
**\* /\***  
**@\***  
**//name**

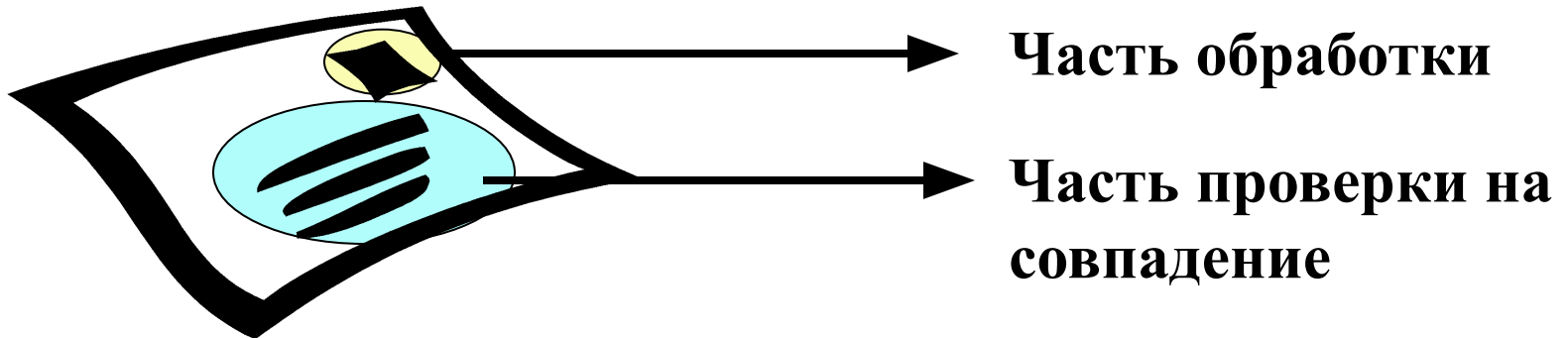
# Фильтрация и логические операторы

- Операции фильтрации могут содержать такие выражения, как логические выражения с применением операторов AND, OR и NOT
- Различные логические операторы, которые могут быть использованы:

Оператор	Описание
AND	Логическое И
OR	Логическое ИЛИ
NOT	Логическое отрицание
=	Равенство
!=	Не равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно

# Шаблоны

- Инструкции в таблице стилей XSL, которые управляют преобразованием элемента и его содержимого, называются шаблонами (templates)
- Представлены тэгами `<xsl:template>...</xsl:template>`
- Используются при помощи элемента `<xsl:apply-templates/>`
- Шаблон содержит две части
  - ◆ Часть проверки на совпадение (matching part)
  - ◆ Часть обработки (processing part)





# Типы совпадений

Совпадение  
по имени

Совпадение по  
предку

Совпадение по  
нескольким  
именам

Совпадение по  
корневому  
элементу



Совпадение

Совпадение по  
идентификатору ID

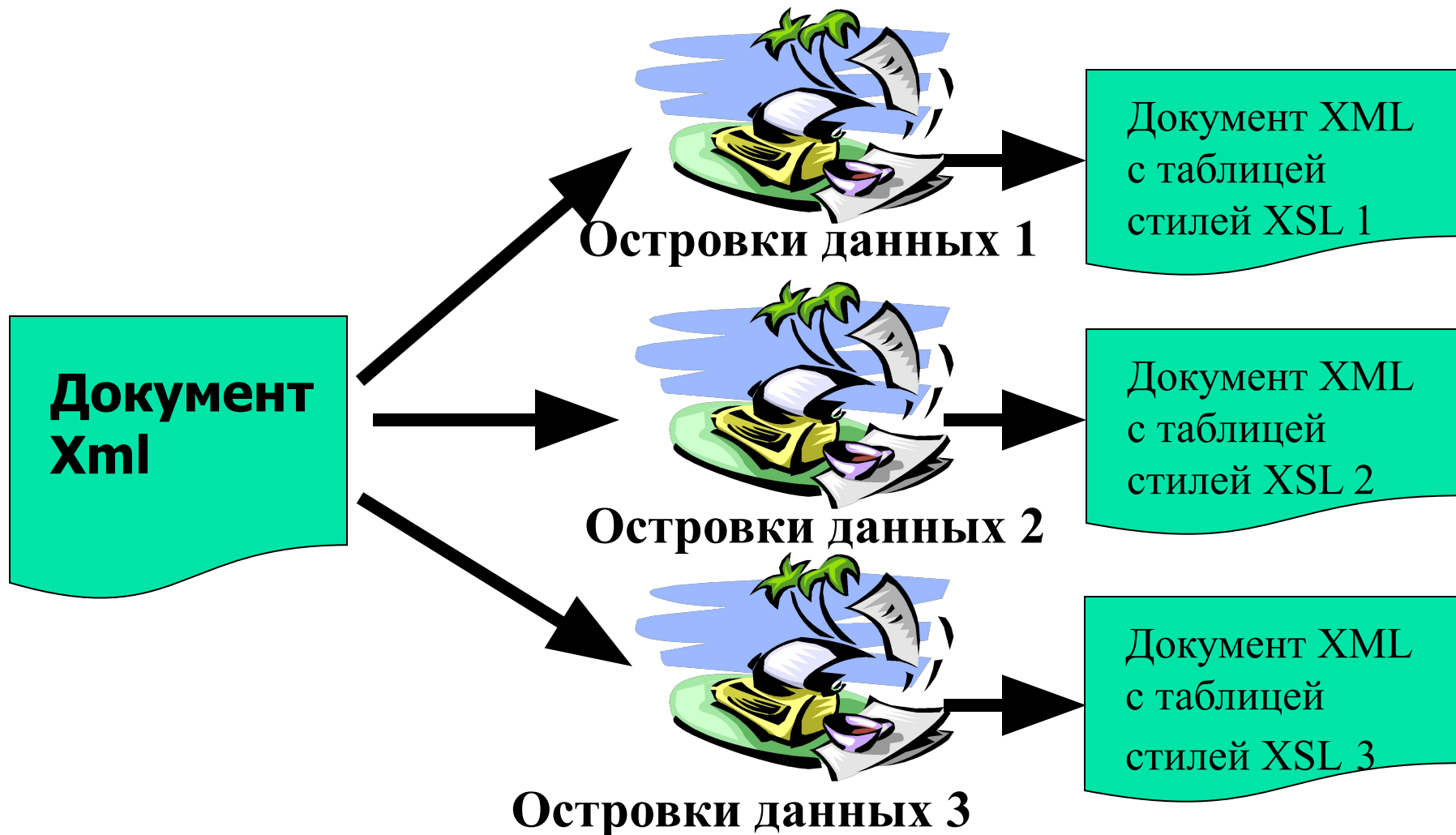
Совпадение по  
шаблонным  
символам

Совпадение по  
атрибуту

# Обработка выражений

- XSL поддерживает пять типов выражений. Вот они:
  - ◆ Node Sets – Наборы узлов
  - ◆ Booleans – Логические выражения
  - ◆ Strings – Строки
  - ◆ Numbers – Числовые выражения
  - ◆ Result Tree Fragments – Фрагменты дерева результата

# Переключение стилей



# xsl:import и xsl:include

- Таблицы стилей, созданные другими разработчиками, могут быть импортированы с использованием `xsl:import`
- Синтаксис импортирования таблиц стилей:
- `<xsl:import href= 'another stylesheet.xml' />`
- `<xsl:import href= 'another stylesheet1.xml' />`
- Все импортированные таблицы стилей организуются в дерево импортирования.
- При импортировании таблицы стилей в дереве импортирования создаётся новый узел

# Различия между XSL и CSS

<p>CSS работает посредством присваивания набора свойств выводу элементу HTML</p>	<p>XSL обеспечивает средства преобразования документов XML</p>
<p>CSS определяет визуальное представление страницы</p>	<p>XSLT позволяет установить соответствие определённых образцов с содержимым исходного документа</p>
<p>It does not change the structure of the document</p>	<p>Преобразует код XML в структуры, такие, как списки или таблицы</p>

# ИТОГОВЫЙ обзор – I

- XML является переносимым языком. Мы можем переслать документ XML с одной системы на другую и даже от приложения к приложению, без изменения содержания.
- Таблица стилей (style sheet) – это набор инструкций, используемый для вывода документов.
- Таблицы стилей могут быть написаны на нескольких языках. Вот два примера таких языков:
  - ◆ **Cascading Style Sheets (CSS)**, расширение HTML
  - ◆ **Extensible Stylesheet Language (XSL)**, разработанный специально для XML язык стилей
- Селектор идентифицирует тэг, к которому применяется стиль, а объявление содержит правила стиля, применяемого к селектору.
- Существуют три вида селекторов – Simple (Простые), Multiple (Составные) и Contextual (Контекстные) селекторы.
- Блок текста может быть помещён в панель, а эта панель может быть затем размещена в браузере настольного компьютера.

# ИТОГОВЫЙ обзор – II

- XSL – это язык описания таблиц стилей, созданный специально для XML. Он используется для преобразования XML-данных в HTML-документ.
- XSL обеспечивает поддержку следующих образцов (patterns):
  - ◆ Sorting (Сортировка)
  - ◆ Operators (Операторы)
  - ◆ Filtering (Фильтрация)
- Типы совпадений образцов:
  - ◆ Совпадение по атрибуту
  - ◆ Совпадение по идентификатору ID
  - ◆ Совпадение по образцу с шаблонными символами (wildcard)
  - ◆ Совпадение по корневому элементу (root)
  - ◆ Совпадение по имени
  - ◆ Совпадение по предку
  - ◆ Совпадение по нескольким именам
- XSLT и CSS являются стандартами, совместимыми друг с другом