XSL: Formatting Objects (FO)

XSL-FO is about formatting XML data for output

Introduction

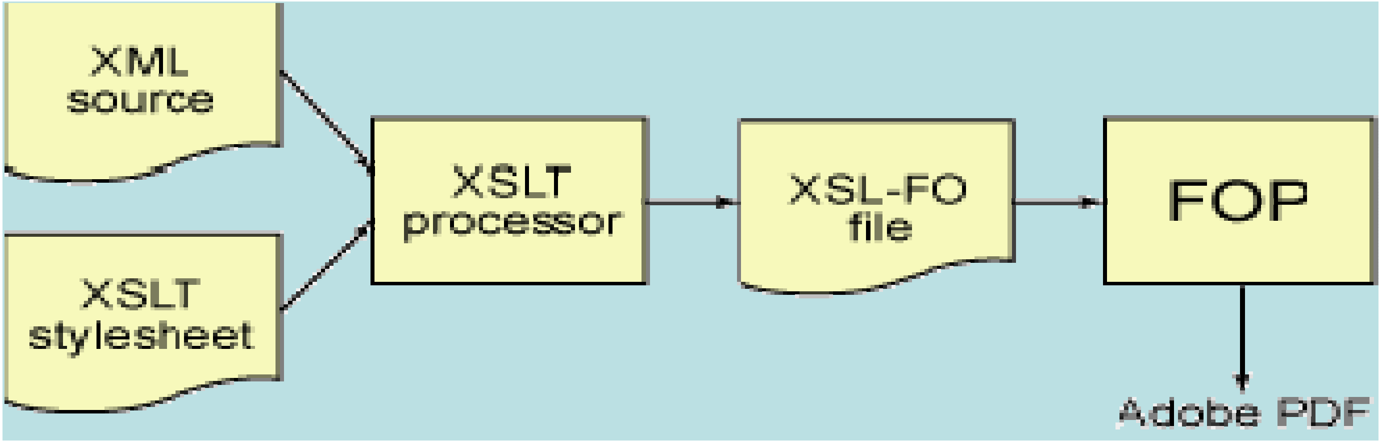
- **What is XSL-FO (formatting objects)?**
  - **XSL-FO stands for Extensible Stylesheet Language Formatting Objects**
    - **Styling** is both about **transforming and formatting** information
  - **XSL-FO is about formatting**
    - XSL-FO is an **XML based markup language describing the formatting** of XML data for output to **screen**, **paper** or **other media**
  - **XSL-FO is a W3C Recommendation**
    - XSL-FO became a **W3C Recommendation** 15. October 2001. Formally named XSL
  - **XSLFO** is an XML vocabulary that is used to **specify a pagination and other styling** for page layout output

Introduction (cont.)

- XSLFO can be used in conjunction with **XSLT** to convert from **any XML format** into a **paginated layout** ready **for printing or displaying**
- XSLFO defines **a set of elements** in XML that describes the way **pages** are set up.
- The **contents of the pages** are filled from **flows**.
- There can be **static flows** that appear on every page (for headers and footers) and the **main flow** which **fills the body** of the page

Introduction (cont.)

- The Object Formatting Process
  - Use an **XSLT stylesheet to transform** the XML document into a file of **XSL-FO elements**
  - To perform the transformation, you simply invoke the **XSLT processor** with the **XML document** and the **stylesheet**
    - An **XSLT stylesheet** that **converts** XHTML elements into **formatting objects**
  - Use **a rendering engine** (for example, **FOP**) to convert the **XSL-FO elements** into a **PDF file**

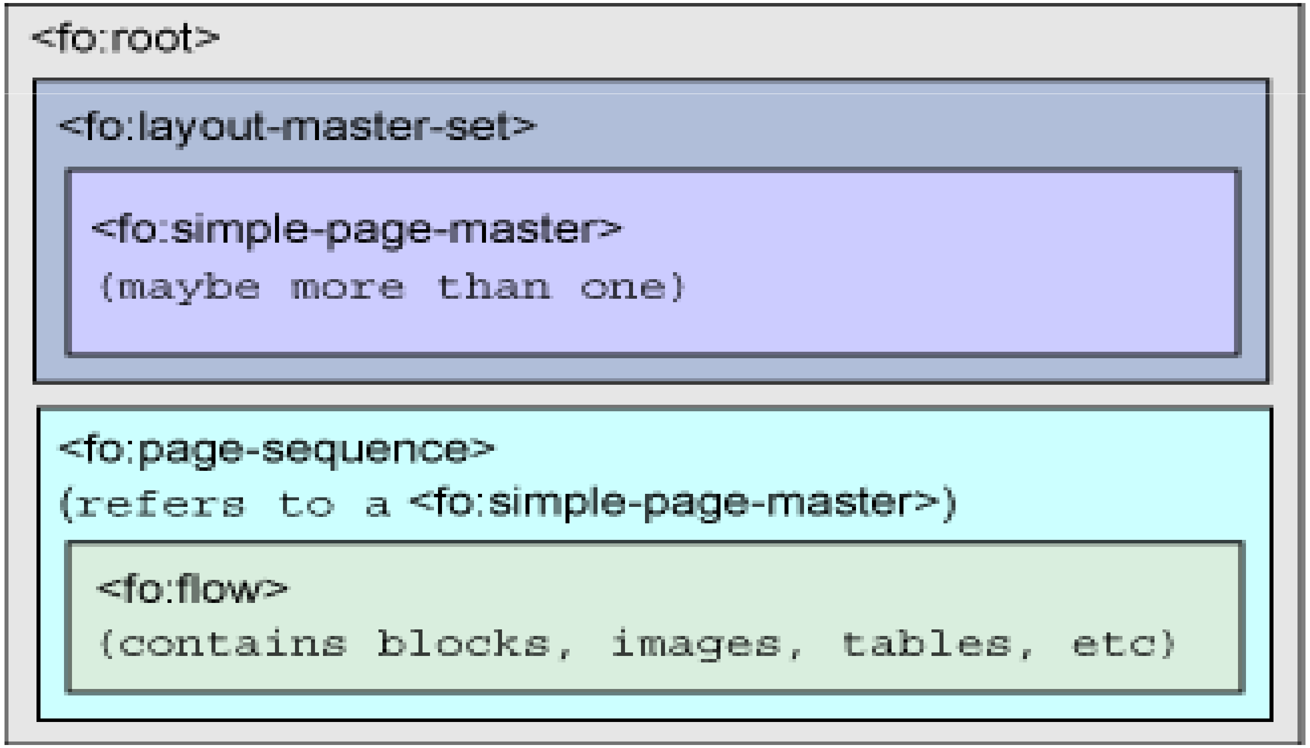


Software Tool Set Up

- XSL-FO Processors
- FOP (Formatting Objects Processor) from apache
  - http://www.apache.org/dyn/closer.cgi/xml/fop
  - Version: the **0.20.5** version of FOP
- Requirement
  - XML parser
  - XSLT Processor
  - Apache Batik: an SVG library

XSL-FO Document

- **XSL-FO documents are XML files with output information**
  - They contain information about the **output layout and output contents**
  - **XSL-FO documents** are stored in files with a ***.fo** or a ***.fob** extension
- **XSL-FO document structure at a glance**



Most of the things in the figure never change

XSL-FO Document (cont.)

XSL-FO Document Structure

XSL-FO documen

The <fo:root> contains a <fo:layout-master-set>
and a <fo:page-sequence>.
The <fo:layout-master-set> contains page layouts information,
The <fo:page-sequence> contains actual content

```
<?xml version="1.0
<fo:root xmlns:fo="
<fo:layout-master-set>
  <fo:simple-page-master master-name="A4">
    <!-- Page template goes here -->
  </fo:simple-page-master>
</fo:layout-master-set>
<fo:page-sequence master-reference="A4">
  <!-- Page content goes here -->
</fo:page-sequence>
</fo:root>
```

XSL-FO Document (cont.)

- The **<fo:root>** element contains the XSL-FO document.
- It also **declares the namespace** for XSL-FO:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      ................
      <!-- The full XSL-FO document goes here -->
      ................
</fo:root>
```

Typically, the **root element** contains

- A **<fo:layout-master-set>**
- Followed by **one or more** <fo:page-sequence>**s**

XSL-FO Document (cont.)

- Each **<fo:simple-page-master>** element contains **a single page template**
- The element **defines the layout** for a particular page
- Each template must have a **unique name (master-name)**:

```
<fo:simple-page-master master-name="A4">
        <!-- One page template (the layout) goes here -->
</fo:simple-page-master>
```

XSL-FO Document (cont.)

- **Here's an example,**
- **master-name**
  - Defines **a name** for this page master.
  - You can create several different **<fo:simple-page-master>** elements
  - Then refer to **each of them** as you need to use different page layouts throughout your document.
- **margin-top and margin-bottom**
  - Define the **margins** at the **top and bottom** of the page.
  - Acceptable units are **points, inches, and centimeters**

```
<fo:simple-page-master
  master-name="main"
  margin-top="36pt"
  margin-bottom="36pt"
  page-width="8.5in"
  page-height="11in"
  margin-left="72pt"
  margin-right="72pt">

<fo:region-body
  margin="50pt" />

</fo:simple-page-master>
```

XSL-FO Document (cont.)

- **page-width and page-height**
  - Define the **size** of the physical page.
  - This example defines **a letter-sized page**;
  - To use **A4-sized paper**, the attributes page-width**="21cm"** and page-height**="29.7cm"** would do the trick.
- **margin-left and margin-right**
  - Define the **margins** at the **left and right** side of the page.

```
<fo:simple-page-master
  master-name="main"
  margin-top="36pt"
  margin-bottom="36pt"
  page-width="8.5in"
  page-height="11in"
  margin-left="72pt"
  margin-right="72pt">

<fo:region-body
  margin="50pt" />

</fo:simple-page-master>
```

Units in XSL-FO documents

XSL-FO supports these **actual units** for length properties, for measuring items such as **margin-left**, **page-width**, and **page-height**:

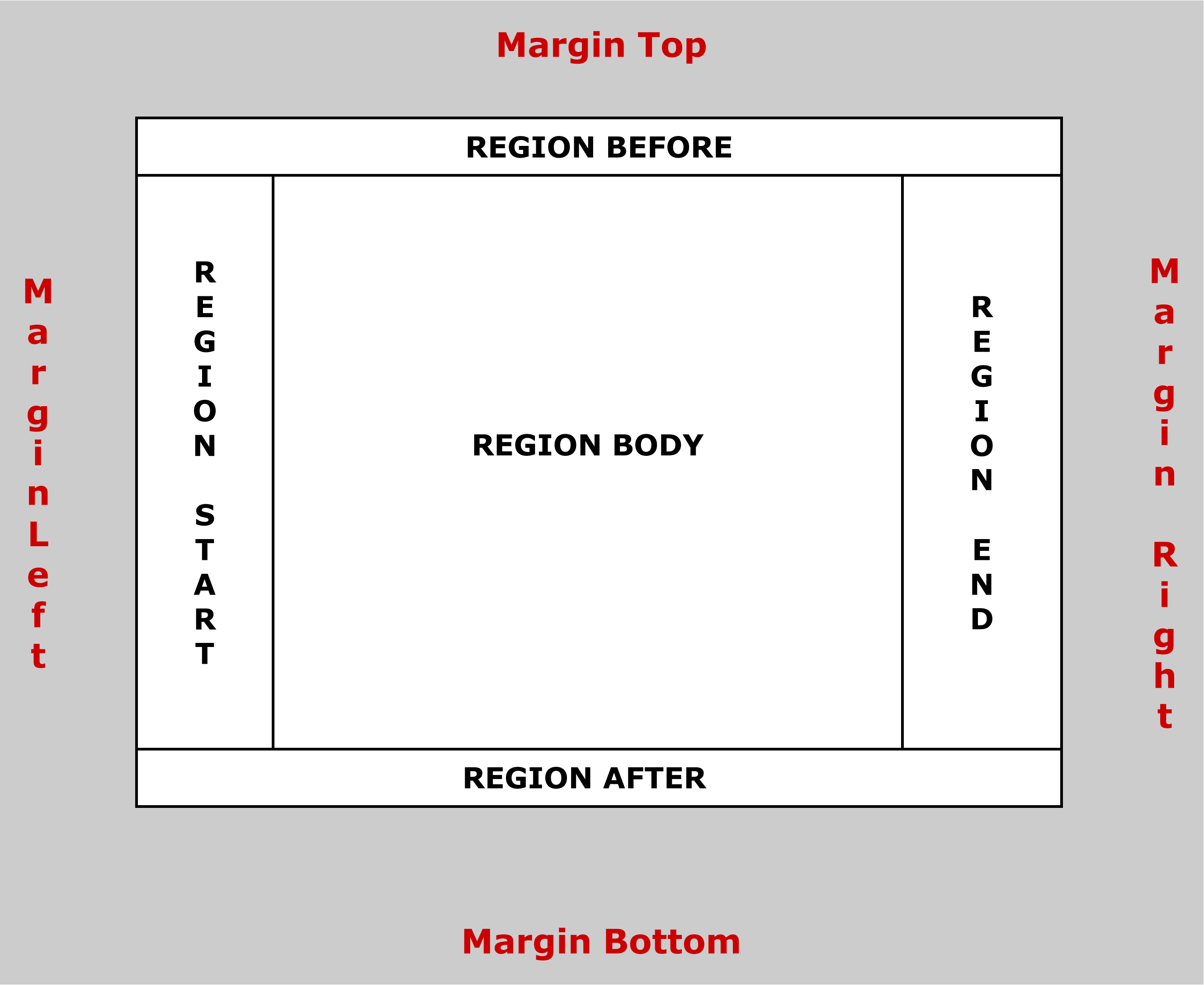
Unit	Meaning
cm	centimeters
mm	millimeters
in	inches
pt	points (72 points = 1 inch)
pc	picas (12 points = 1 pica, 6 picas = 1 inch)
px	pixels (sometimes different from one formatter or device to the next, so **be careful**)
em	the width of a capital *M*

The <fo:region-body> element

- The XSL-FO spec defines **five regions** on a page
  - **region-body** defines the **dimensions of the main area** in the center of the page. Here's a sample:

    <fo:region-body margin="50pt" />

  - **margins** for the region-body area.
  - **region-before**, the area at the top of the page (normally used for **running heads**)
  - **region-after**, the area at the bottom of the page (normally used for **running feet**)
  - **region-start**, the area to the left of the page
  - **region-end**, the area to the right of the page



Note: These definitions assume that **the text** in your document goes **from left to right** and **top to bottom**

XSL-FO Document (cont.) -- Page

- One or more **<fo:page-sequence>** elements describe **page contents (i.e., output pages)**
- It defines **the sequence of page layouts** to be used within the document
- The **master-reference** attribute refers to the **“master-name”** of the **<fo:simple-page-master>** with the same name
- Each **output page** refers to **a page master** which defines **the layout**
- Each **output page** has a **<fo:flow>** element defining **the output**
- Each **output page** is printed (or displayed) **in sequence**

```
<fo:page-sequence master-reference="A4">
     <!-- Page content goes here -->
</fo:page-sequence>
```

Note: The master-reference **"A4"** does not actually describe a predefined page format. **It is just a name**. You can use any name like "**MyPage**", "**MyTemplate**", etc.

XSL-FO Output

- How to insert **some content** to be rendered
- XSL-FO defines output inside **<fo:flow>** elements
- **XSL-FO Page, Flow, and Block**
  - **"Blocks"** of content **"Flows"** into **"Pages"** and then to the output media
  - XSL-FO output is normally **nested** inside **<fo:block>** elements, **nested** inside **<fo:flow>** elements, **nested** inside **<fo:page-sequence>** elements

```
<fo:page-sequence>
  <fo:flow flow-name="xsl-region-body">
    <fo:block>
      <!-- Output goes here -->
    </fo:block>
  </fo:flow>
</fo:page-sequence>
```

What's this?

An Output Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
         <fo:simple-page-master master-name="A4">
                <fo:region-body margin="50pt" />
         </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="A4">
         <fo:flow flow-name="xsl-region-body">
              <fo:block>Hello W3Schools</fo:block>
                <fo:block>Hello Class of XML and SOA</fo:block>
         </fo:flow>
    </fo:page-sequence>
</fo:root>
```

XSL-FO Output -- <fo:flow>

- **<fo:page-sequence>**
- XSL-FO pages are filled with data starting from **<fo:flow>** elements
- **<fo:flow>** defines some content that **will flow** within the current **layout**
- When **the page is full**, the same page master will be used over (and over) again until all the text is printed.

```
<fo:flow  flow-name="xsl-region-body">
```

The legal values are:
xsl-region-body (into the region-body)
xsl-region-before (into the region-before)
xsl-region-after (into the region-after)
xsl-region-start (into the region-start)
xsl-region-end (into the region-end)

XSL-FO Output -- <fo:block>

- Two main XSL-FO elements for formatting content are **<fo:block>** and **<fo:inline>**
- Blocks are **sequences of output** in **rectangular boxes**
- Think of it as similar to the **HTML <p>** element.
- A **<fo:block>** element always causes **a line break**

```
<fo:block border-width="1mm">
   This block of output will have a one millimeter
   border around it.
</fo:block>
```

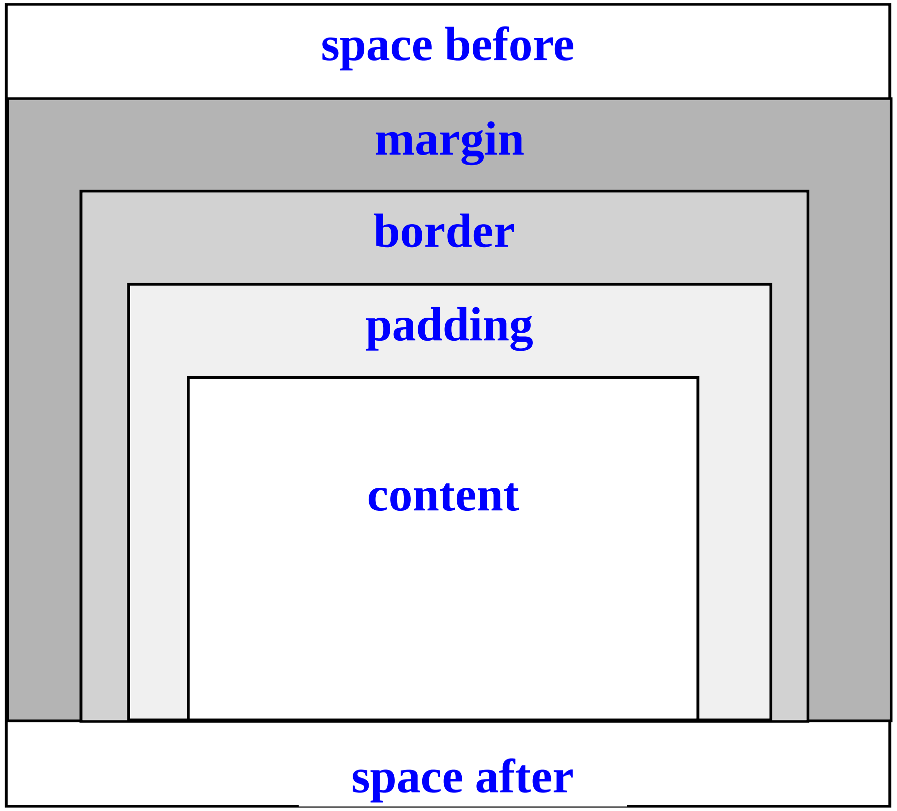
XSL-FO Output -- <fo:block>(cont.)

Since **block areas** are rectangular boxes, they share many common area properties:

- **space before and space after**
- **margin**
- **border**
- **padding**

Note: for the attributes of each,

Refer to W3c spec.



- The **content** area contains the actual content like **text, pictures, graphics, or whatever**.

An XSL-FO Blocks Example

```
<fo:block
       font-size="14pt"
       font-family="verdana"
       font-color="red“
       space-before="5mm"
       space-after="5mm">
       W3Schools
</fo:block>
<fo:block
       text-indent="5mm"
       font-family="verdana"
       font-size="12pt“
       space-before="5mm"
       space-after="5mm">
       At W3Schools you will find all the Web-building tutorials you need, from basic
       HTML and XHTML to advanced XML, XSL, Multimedia and WAP.
</fo:block>
```

Normally XSL-FO document do not combine formatting information and content like we have done here

XSL-FO Output -- <fo:inline>

- The **<fo:inline>** formatting object is commonly used **for formatting** a portion of text
- It defines some **text properties** within an existing **<fo:block>**
  - If you want **to italicize** several words within a paragraph, as in the example, you use **<fo:inline>** to do the job.

```
<fo:block font-size="20pt" line-height="30pt">
    This is a paragraph of text. font size = 20 pt, line height = 30pt,
    Notice that we applied fo:inline here:
    <fo:inline font-style="italic" font-weight="bold" color="red">
        this meaningless prose
    </fo:inline>
    , the inline stops here. drones on and on, the FOP
    software automatically calculates line breaks for us.
    Isn't that fascinating?
</fo:block>
```

XSL-FO Output -- <fo:inline>

- Here's how you can use the attributes of **XSL-FO <fo:inline>** element to **format text**:
  - **Bold text:** Use the <fo:inline> element with an attribute of font-weight="bold".
  - *Italicized text:* Use the <fo:inline> element with an attribute of font-style="italic".
  - Monospaced text: Use the **<fo:inline>** element with an attribute of **font-family="monospace".**
  - **Changing fonts**: Use the **<fo:inline>** element with an attribute of **font-family="serif"** for **a serif font** (usually similar to Times-Roman).
  - Use the attribute **font-family="sans-serif"** for **a sans serif font** (usually similar to Arial).
  - See the **FOP documentation** for information on how to convert the fonts installed on your machine into fonts FOP can use
- Others: http://www.w3schools.com/xslfo/obj_inline.asp

XSL-FO Properties

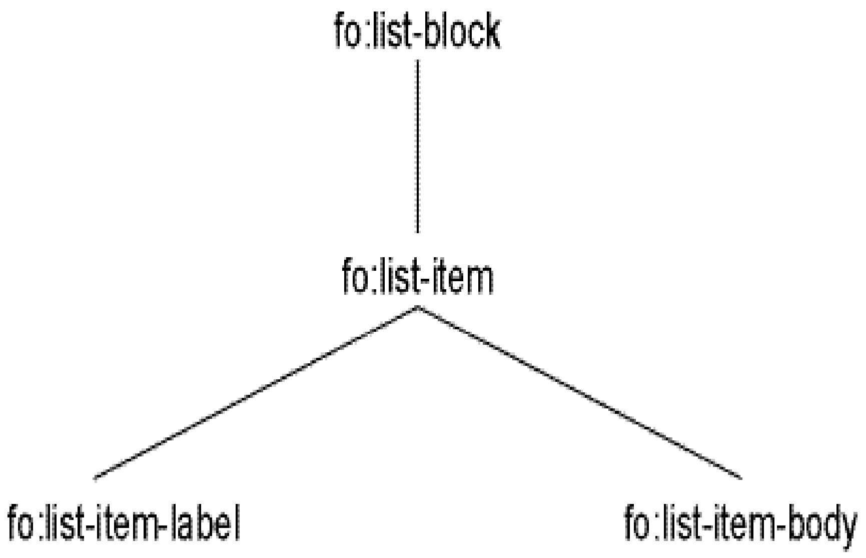
- Many **XSL-FO properties** are identical to **the CSS properties** you might recognize.
- **XSL-FO** and **CSS** also share the same way that **elements** usually **inherit properties** from their ancestors
- An XSL-FO property is just **an XML attribute**

XSL-FO – list

- When building these lists with XSL-FO, you must **handle a number of things** yourself
  - How much **horizontal space** separates the label of a list item (the bullet or number) and the item itself
  - How much the list items are **indented** from the normal margins
  - What the **bullet character** is
  - How much **vertical spacing** falls between list items
- Almost all of these properties **are handled automatically** in an **HTML browser**, **but** when you **use formatting objects**, it's **up to you** to define them.

XSL-FO – list (cont.)

- XSL-FO uses **list-blocks** to define **lists**
- There are **four XSL-FO objects** used to create lists:
  - **fo:list-block** (contains the whole list)
  - **fo:list-item** (contains each item in the list)
  - **fo:list-item-label** (contains the label for the list-item - typically an <fo:block> containing a number, character, etc.)
  - **fo:list-item-body** (contains the content/body of the list-item - typically **one or more <fo:block>** objects)



XSL-FO list - An Example

```
<fo:list-block>
       <fo:list-item>
               <fo:list-item-label>
                 <fo:block>*</fo:block>
               </fo:list-item-label>
               <fo:list-item-body>
                 <fo:block>Volvo</fo:block>
               </fo:list-item-body>
       </fo:list-item>
       <fo:list-item>
               <fo:list-item-label>
                 <fo:block>*</fo:block>
               </fo:list-item-label>
               <fo:list-item-body>
                 <fo:block>Saab</fo:block>
               </fo:list-item-body>
       </fo:list-item>
</fo:list-block>
```

XSL-FO – table <fo:table>

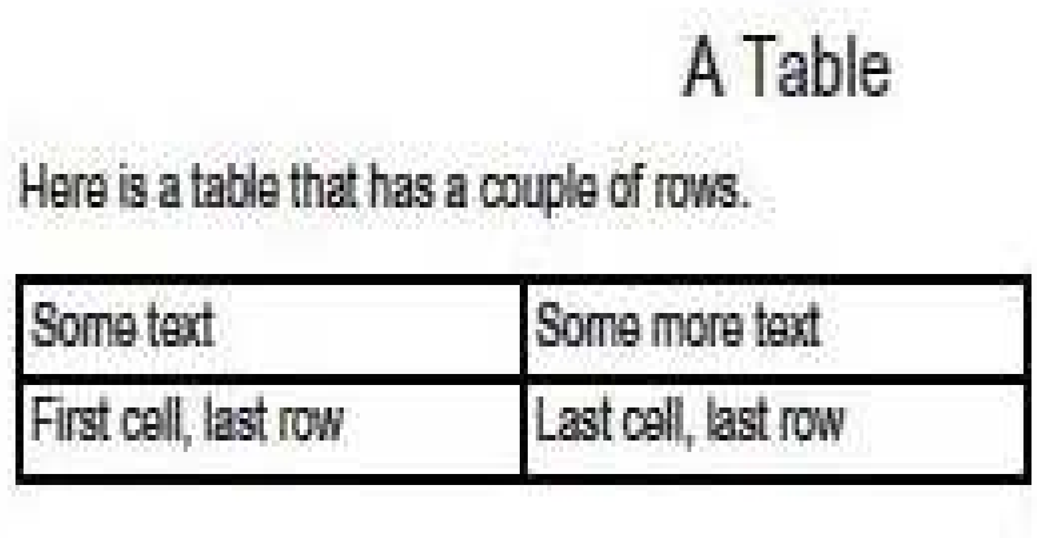
- The **<fo:table>** is used **to format** the **tabular material** of a table.
- The **<fo:table>** contains
  - <fo:table-column>
  - <fo:table-header>
  - <fo:table-body>
  - <fo:table-footer>
- Each of these elements has one or more <fo:table-row> objects, with one or more <fo:table-cell> objects:

XSL-FO element	Description
<fo:table>	**Formats the tabular material of a table**
<fo:table-column>	**Formats the columns of a table**
<fo:table-header>	**Defines a table header**
<fo:table-body>	**Container for table rows and table cells**
<fo:table-footer>	**Defines a table footer**
<fo:table-row>	**Defines a table row**
<fo:table-cell>	**Defines a table cell**
<fo:table-caption>	**Formats a table and its caption**

XSL-FO – table <fo:table>

- The basic **table structure**
  - Information of the table: **caption, header, and footer**

- A typical example:



A Table

Here is a table that has a couple of rows.

Some text	Some more text
First cell, last row	Last cell, last row

XSL-FO and XSLT

- With a little help from **XSLT**, it can then **be transformed** to **XSL-FO** file

- The file is ready to **be formatting** to other type of document: i.e. **PDF file**

```
<?xml version="1.0"?>
        <xsl:stylesheet version="1.0"
          xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
          xmlns:fo="http://www.w3.org/1999/XSL/Format"
          xmlns:fox="http://xml.apache.org/fop/extensions">
        <xsl:template match="header">
                <fo:block
                    font-size="14pt" font-family="verdana" font-color="red"
                    space-before="5mm" space-after="5mm">
                    <xsl:apply-templates/>
                </fo:block>
        </xsl:template>
        <xsl:template match="paragraph">
                <fo:block
                    text-indent="5mm"
                    font-family="verdana" font-size="12pt"
                    space-before="5mm" space-after="5mm">
                    <xsl:apply-templates/>
                </fo:block>
        </xsl:template>
</xsl:stylesheet>
```