Introduction to XSL-FO Concepts

(Printing Directly from XML)

Deborah Aleyne Lapeyre and B. Tommie Usdin
Mulberry Technologies Inc.
17 West Jefferson St.
Suite 207
Rockville MD 20850
Phone: 301/315-9631
Fax: 301/315-8285
info@mulberrytech.com
`http://www.mulberrytech.com`

Version 1.0 (January 2006)

Introduction to XSL-FO Concepts

Appendix

Introduction to XSL-FO Concepts

slide 1

Administrivia

- Start, end, break
- How this will work
- Questions are always in order
- Who we am
- Anything else?

slide 2

Where We Are *Not* Going in This Talk

(this is concepts not syntax)

- Details of XSL-FO syntax
- How to write an XSL-FO "program"
- How to design pages for XSL-FO
- How to solve your business problems
- Particular XSL-FO tools
  - specific features
  - compliance
  - recommendations

slide 3

Where We Are Going Today?

Conceptual XSL-FO

- What XSL-FO is and what it is good for
- How XSL-FO works (system overview)
- What XSL-FO specifies (stylesheet overview)
- Who's using XSL-FO (and what it's good for)
- Brief overview of some XSL-FO Tools
- Why everyone isn't using XSL-FO (when not to use)
- What it takes to use XSL-FO
- Looking at some real pages

slide 4

Warning: We are Going to Show Code

- But not much, this is concepts (how and why)
- You will understand the class even if you do not understand the code
- Don't panic if code is too heavy
- Sorry if you wanted to learn to write XSL-FO from this course

A Quick Poll (Who You Are)

- Where in the pipeline
  - content creators / publishers
  - prepress / composition
  - printers
  - print / web / graphic design
  - fulfillment / distribution
  - System analysts / application programmers
  - Training
- What kind of publishing
  - Books (monographs, reference series, etc.)
  - Journals
  - Magazines and newspapers
  - Product documentation
  - Technical documentation
  - Course materials (CBT, course-packages, tests, textbooks plus, etc.)

slide 6

Do You Work with or Know

- HTML
- XML
- SGML
- PDF
- InDesign, QuarkXPress, other desktop publishing packages
- Microsoft Word
- High-powered composition systems (Miles 33, Penta, 3B2, DataLogics)
- Preflighting software
- Workflow software

slide 7

What is XSL-FO

Extensible Stylesheet Language-Formatting Objects

- A way to make print directly from XML
- A page description language (that particular software can read)
- XSL-FO rendering software
  - takes XML as input
  - gives PDF, Postscript (RTF, MIF, etc.) as output
- Described in a W3C recommendation called XSL (Extensible Stylesheet Language)

(Not new, recommendation since 2001)

slide 8

Design Goals for XSL Formatting

- Allow designers to express *how* structured (XML) content should be presented
  - **In print (pamphlet, magazine, bound volume, poster...)**
  - On screen
  - In other media such as audio or braille

(without being tied to a particular application/vendor)

slide 9

The XSL Specification

- XSL specification is in two parts
  - XSLT (the transformation part)
  - XSL-FO (styling and pages)
- XSL specifies an XML tagset (what we call XSL-FO)
  - says how to put content on the page
  - describes page geometry into which XML documents may be transformed (using XSLT)

slide 10

The Dream Behind XSL-FO

- XML separates specification of format from content
  - XML will specify the content
    (list, product name, paragraph, surname)
  - XSL-FO will specify the layout, pagination, styles
- Goal: high-quality, high-volume, content-driven publishing
  - Batch processing on one file or many
  - Formatting is content-driven (from the XML tags)
- Definition of layout and styles
  - Not bound to any platform
  - Not proprietary to any software

slide 11

The Hype Behind XSL-FO

(mostly myth, we'll explain)

- Specify your pages once and make many outputs
  (PDF, Postscript, web, audio)
- Publish lights-out, no human necessary
- All pagination systems will accept XSL-FO input
- These pages are *just as good as* fine typesetting
- These pages are good enough
- Cheaper, faster, better

slide 12

Where Can XSL-FO Fit in Publishing Processing

Currently, many print pages get made *once*

- using desk top publishing systems (InDesign, QuarkXPress)
- using high-end composition systems
- using graphics editors (Photoshop)
- in a word processor (then make PDF)
- in a web design tool (Dreamweaver)

slide 13

When You Have XML Content

- Content is reusable/repurposeable/customizable/personalizable/ internationalizable/localizable
- XML for single source publishing:
  - XML → transform → HTML
  - same XML → high-end composition → book
  - same XML → desktop publishing → journal and magazine
  - same XML → transform → RSS and ATOM syndication
  - same XML → archive/depository/database

slide 14

Making Pages from XML Content

Many different workflows are possible

- transform XML to an application-specific syntax (InDesign, Quark, RTF, …)
- use XML-aware page-layout application
- use dedicated XML page-layout application
- use XSL-FO
  - through "lights-out" batch XSL-FO application
  - through batch runs and tweaking
  - pouring the XSL-FO into an XSL-FO-aware page-layout application

slide 15

XSL-FO Adds Another Option for Making Print from XML

Directly to print without intervening page-layout *by a person* (designs done once, in advance)

- XML → XSL-FO → books, manuals, reports
- XML → XSL-FO → bills, amendments, laws
- XML → XSL-FO → journals and magazines
- XML → XSL-FO → catalogs and manuals
- XML → XSL-FO → pamphlet or newsletter
- XML → XSL-FO → novels

(Don't Worry: This is *not* the automated typesetting of 20 years ago)

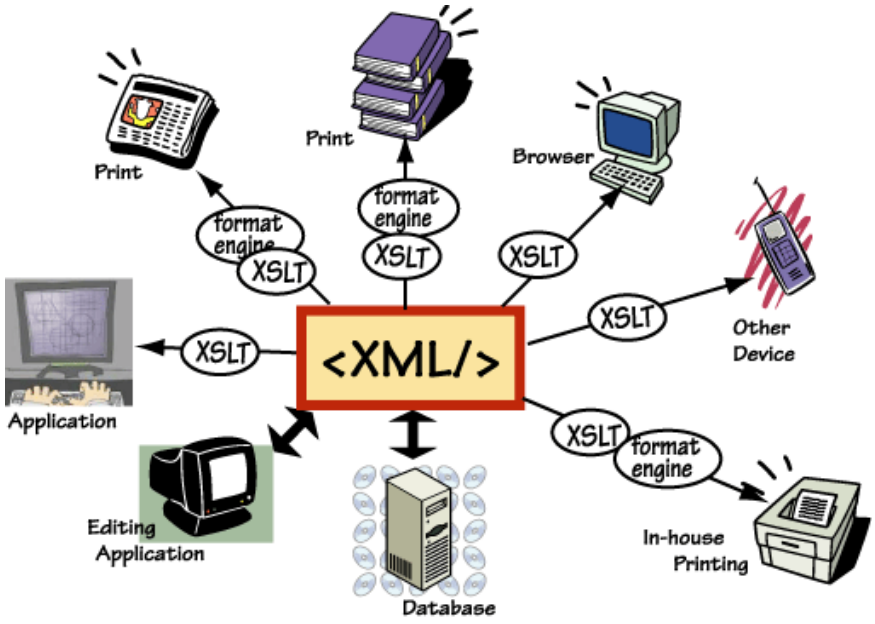
slide 16

How XSL-FO Programs Get Created

- You could write XSL-FO code by hand but
  - it was not designed for that
  - nobody does
- You could use a GUI tool to
  - drag and drop your wishes
  - writes XSL-FO behind the scenes
- But XSL-FO is typically made using XSLT
  - which is an XML transformer
  - changes your XML tags to
  - XSL-FO tags with your content inside them

slide 17

XSL-FO Creates Print (and Web) Output



slide 18

XSL-FO Supports Big Batch Production

- Write instructions once, run on many files
- Hands-free page make-up
- Consistency of generated content
- Format complex documents in seconds
- Run unattended to make consecutive output files
- Save typesetter time on routine tasks
  (some pesky handwork replaced by work for the *software*)

slide 19

Lot of XSL-FO Processing is Batch Mode

- Run XML to pages (according to stylesheet)
- Look at output (PDF, etc.)
  - as printed pages
  - using online viewer
- For formatting you don't like (bad widow, graphic placement)
  - alter the XML document
  - run again
  - alter the XML document
  - run again
  - alter...

slide 20

Flowing XSL-FO into a Composition Engine

(May be the best of both worlds)

- Run XSL-FO stylesheet making formatting objects
- Open results in a composition engine
- Change XML and reflow if content changes
- Make non-content tweaks inside the composition system
  - Graphics placement
  - Column balancing
  - Floats, keeps, widow, orphans
- You get *both*
  - clean validatable (XML) input and batch processing *and*
  - ability to tweak pages by hand

(There are composition systems that accept XSL-FO input)

slide 21

Aside: To Understand XSL-FO You Really Must Understand XSLT

Extensible Stylesheet Language Transformation

(Since most XSL-FO is created via XSLT)

- "The XML Transformation Language"
- Provides transformation and manipulation functions for XML files
- Designed to make XML into something else (HTML, different XML, etc.)
- A W3C Recommendation (original 1999)

slide 22

What XSLT Does is “Transform”

Transform means change

- Reads XML documents and writes *something else*
  - HTML for browsers
  - a different XML tag set
  - typesetting driver file (InDesign, QuarkXPress, FrameMaker)
  - interchange file (RTF, RDF, EDI, etc.)
  - a flat ASCII file (plain text, comma separated etc.)
  - an XSL-FO document (input to formatting process)

slide 23

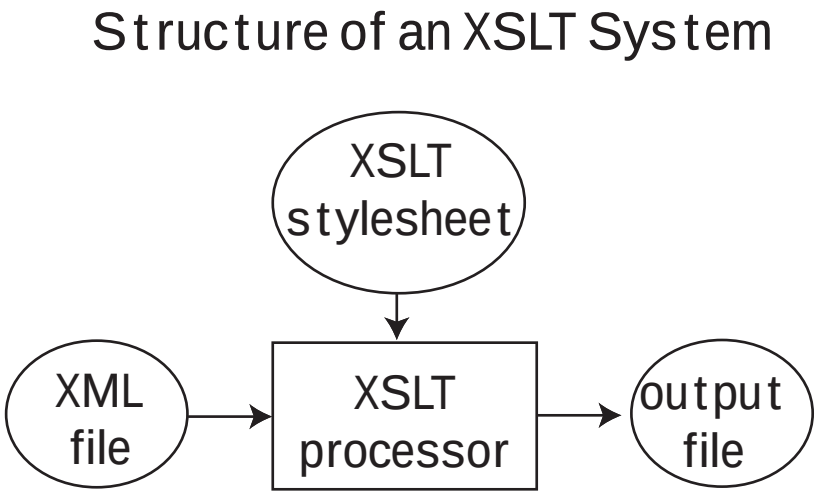
The Very Basics of XSLT Transforms

- Transform
  - Does *not* change the input file
  - Creates one (or more) new output files
- Does *not* make non-XML into XML
- Two basic requirements
  - known XML source (tag set, schema, DTD)
  - known target (such as HTML or XSL-FO)

slide 24

How an XSLT Transform Works

- Reads XML document(s) (tags and text)
- Uses an XSLT stylesheet/transform (a program)
- Runs using XSLT processing software (called an "XSLT Engine")
- Produces output document(s)



(this output file could be an XSL-FO file)

slide 25

Read an XML document

```
<employee-record type="dog" empno="9">
<name>
<first>Sasparilla</first>
<last>Usdin</last></name>
<affiliation>
<title>Deputy in Charge of Chewables</title>
<company>Mulberry Technologies</company>
<location><city>Rockville</city>
<state>MD</state><zip>20850</zip></location>
<email-name>sassy</email-name>
</affiliation>
<height unit="in">36</height><weight unit="lb">70</weight>
</employee-record>
```

slide 26

Transform It into HTML

slide 27

How XSL-FO Works

- XSL-FO is an XML tag set
- XSLT is the middle of XSL-FO
  (how you make XSL-FO from XML)
- How XSL-FO engines work
- Architecture of an XSL-FO system

slide 28

XSL-FO is an XML Vocabulary (Tag Set)

- XSL-FO vocabulary is a set of tags that describe:
  - the layout geometry of a page
    (into which you pour content)
  - a set of formatting objects:
    - that say how to put content on the pages
    - that describe how the document should be rendered

```
<fo:block font-family="Helvetica">
```

slide 29

So an XSL-FO Document is

- An XML document
- Tagged in the XSL-FO tag set
- That contains your text and graphic content
  wrapped in **f**ormatting **o**bject tags (F - O, get it?)

slide 30

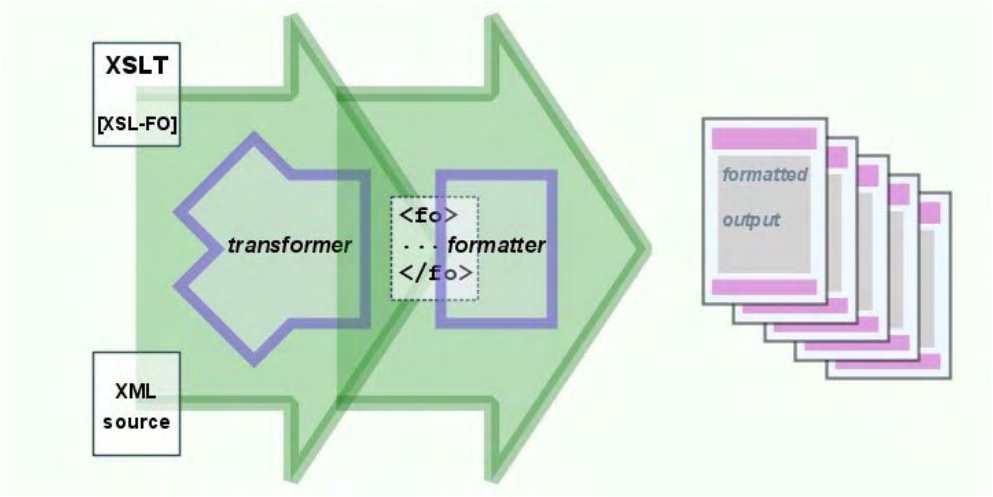
XSLT (Transforming) Builds XSL-FO-tagged Files

- Start with your XML file (tags and text)
- Use XSLT to transform from
  - original XML tags (`<paragraph>`, `<list>`, `<bold>`) into
  - XSL-FO tags (`<fo:block>`, `<fo:list-block>`, `<fo:inline>`)

Then XSL-FO software makes pages from those XSL-FO tags

slide 31

How XSL_FO Formatting Works



slide 32

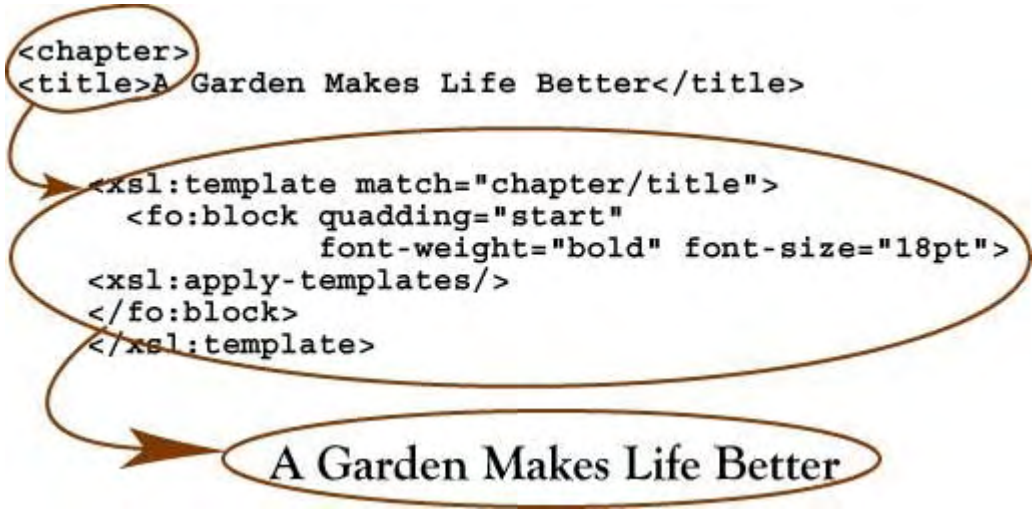
The Complete XSL-FO Picture

In XSL-FO processing

- Tags and content in an XML document
- Are transformed
  - *from* their XML tags
  - *into* XSL-FO tags
    - which describe the styling
    - and hold the content
- XSL-FO rendering engine
  - understands these tags
  - makes pages from them (PDF, Postscript, RTF, etc.)
- And printed/displayed using a display engine
  (like Adobe Acrobat for the PDF or Word for the RTF)

slide 33

XML is Turned into FO-XML is Turned into Pages



slide 34

Role of the XSL-FO Engine

- Understands the XSL-FO vocabulary

```
<fo:block>
<fo:wrapper>
<fo:wrapper font-style="italic">
```

- Produces display for paper (we don't know/care how)
- An XSL-FO programmer needs to know
  which XSL-FO tags to use to make the pages he/she wants

slide 35

How Using XSL-FO Compares to Using HTML

HTML is one way to display XML

- HTML is a vocabulary of tags into which XML can be translated
- HTML software (a browser) makes the display from those tags
- We know the vocabulary to make the screen effect we want

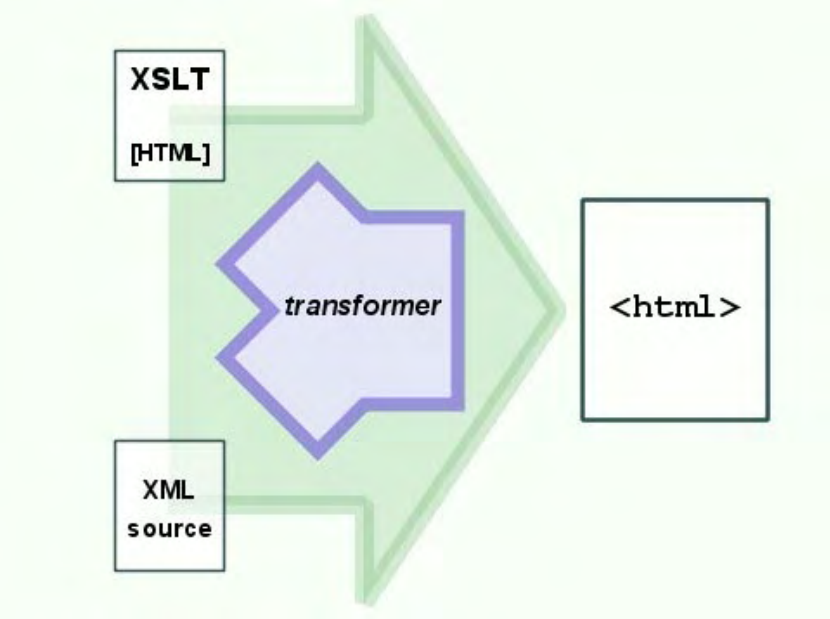
```
<p>...</p>
<b>Wow!</b>
<span style="font-style: italic">...</span>
```

(Native HTML does not enable complex page layout, multiple columns, running heads, keeps and breaks)

slide 36

Using HTML to Make a Display

(we make HTML; browser makes display)



HTML Example

- Start with XML-tagged source

```
<recipe>
  <title>Waffles with Syrup</title>
  ...
</recipe>
```

- Use XSLT to transform that XML into HTML source

```
<h1>Waffles with Syrup</h1>
```

- And the browser (like IE or Firefox) understands the HTML tags (`<h1>`) and displays

slide 38

XSL-FO is Another Way to Display XML

- XSL-FO is a vocabulary of tags into which XML can be translated
- XSL-FO software (a rendering engine) makes the display from those tags
- We know the vocabulary to make the print effect we want

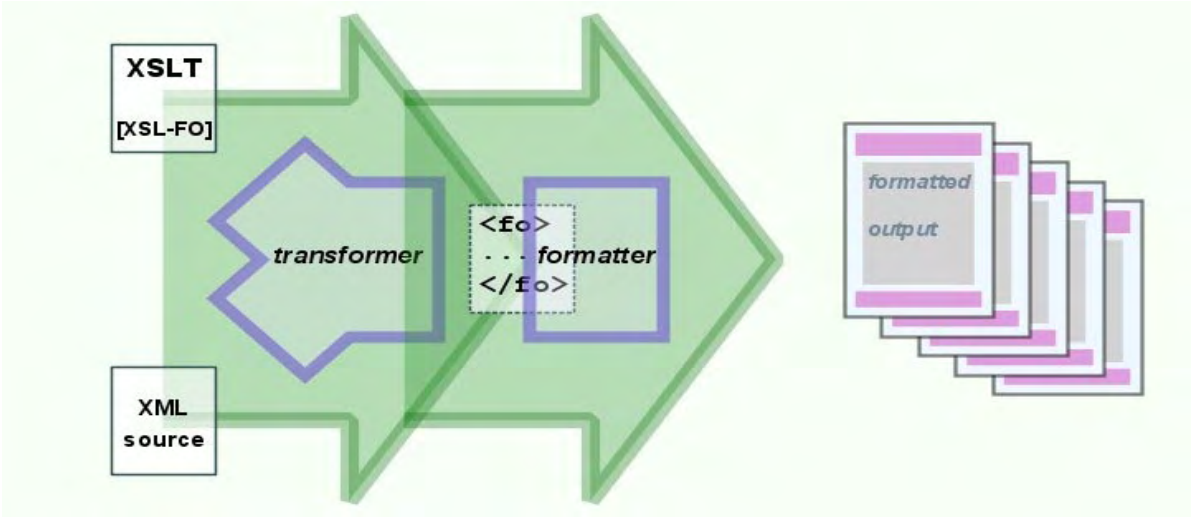
```
<fo:block>...</fo:block>
<fo:inline font-style="bold">Wow!</fo:inline>
<fo:inline font-style="italic">...</fo:inline>
```

(XSL-FO can describe more complex, sophisticated page layouts than HTML)

slide 39

Using XSL-FO to Make a Display

(typically a PDF page)



XSL-FO Example

(one more step than in the HTML)

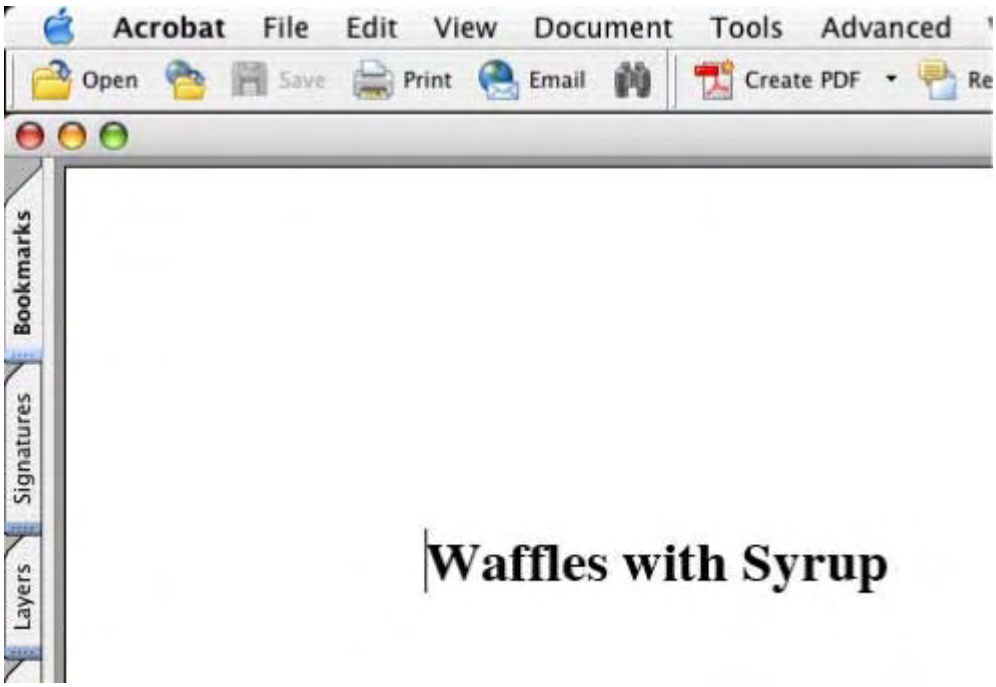
- The source XML

```
<recipe>
  <title>Waffles with Syrup</title>
  ...
</recipe>
```

- Use XSLT to transform that XML into

```
<fo:block space-after="4pt">
   <fo:wrapper font-size="14pt" font-weight="bold">
     Waffles with Syrup
   </fo:wrapper>
</fo:block>
```

- Rendering software understands the XSL-FO tags and creates PDF
- And the display engine (like Acrobat) understands the PDF and displays



slide 41

Features of XSL-FO Formatting

- XSL-FO
  - is a typesetting application
  - has a complex, extensible Page Model (multicolumn)
  - has sophisticated text typography (hanging indents, multi-font)
  - uses typographic concepts and terminology
- XSL-FO usually created by transformation
- Formatting is based on XML tagging
- Internationalization designed in

slide 42

XSL-FO Designed to Deal with Typesetting Complexity

- Multiple columns
- Endnotes and footnotes
- Block-level behaviors (leading, margins, padding, space before/after)
- Text-level behaviors (font family, style, weight, size, color)
- Hyphenation and justification
- Marginalia
- Rules and leaders
- Page markers and links
- Autogenerated cross-references, Tables of Contents, indexes

slide 43

XSL-FO's Page Model

- XSL-FO describes page layout
  - page size
  - margins and columns and gutters
  - headers and footers
  - side-bars, etc
- Can describe sequences of different page layout templates
  - First page followed by later pages
  - Recto/verso alternating

slide 44

XSL-FO Uses Typographic Terminology

and frequently Cascading Stylesheet (CSS) names

- The XSL-FO spec and XSL-FO stylesheets talk about
  - floats and keeps
  - blocks and inline
  - masters and sequences
  - font, point-size, leading, font-weight, serif, cursive
  - margins, leading, line-height, padding
- You need to know some typography to write XSLT/XSL-FO stylesheets (at least the terminology)

slide 45

XSL-FO Usually Created by Transformation

- XSL-FO is typically made by XSLT
- XSLT is a transformer (changing source tags to XSL-FO tags)
- Since you're transform anyway, take advantage
  - Rearrange the order of the XML
  - Delete things from the XML
  - Duplicate things in the XML
  - Add things to the XML
  - Select only the parts you want for publication

slide 46

XSL-FO Formatting Flows from the XML

In XSL-FO, you can make formatting distinctions based on

- Element type
- Attribute value
- Element or attribute content
  (`<title>`s containing "Introduction")
- Many more refined tests
  - Element position
    (first, last, *n*th member of a `<list>`)
  - Element context
    (chapter title, figure title, table title)
  - Grouping elements (lists versus items in the list)

slide 47

Internationalization and Accessibility

- XML character set is internationalized (Unicode)
- XSL-FO supports non-Western writing directions
  - Writing mode describes layout and direction of writing, e.g.
    - Left-to-right-top-to-bottom (*lr-tb*)
    - Top-to-bottom-right-to-left (*tb-rl*)
  - Relative frame of reference
    (not “top”, “bottom”, “left” and “right” but
    “before”, “after”, “start” and “end”)
- Formatting properties for audio media

slide 48

XSL-FO Works Through Stylesheets

- A stylesheet is a computer program
- Before programming
  - start with design
  - design is still king
- The program itself (a transformation)
  - Page geometry
  - Formatting Objects (what and how)
  - Where does the content go anyway?

slide 49

The XSL-FO Big Picture

(format-specification, transforming, and rendering are separate steps)

- This whole thing works because
  - there is software (XSL-FO Rendering Engines)
  - hard-wired to understand XSL-FO tags
  - which uses those tags to make pages
- XSL-FO tags describe (in a system-independent way)
  - text formatting objects with styling
  - page geometry
- And there is a transform (XSLT) to turn your XML tags into XSL-FO tags

"XSL-FO is the intermediate form between media-neutral XML and media-dependent output" - Stephen Deach

slide 50

All XSL-FO Programs Start with Page Design

(all typesetting does)

- Page layout(s) must be designed
  - first page/recto/verso pages
  - headers and footers
  - Whitespace and margins
- Text typography must be designed
  - fonts
  - text size and leading
  - title design and indents
- **This is still a people process!**

Be prepared to accept small tweaks in your original design (minor XSL-FO or engine limitations

slide 51

Human Page Design is Translated into XSL-FO Formatting Objects

- Formatting Objects Describe Page Layout
  - page size
  - margins, columns
  - headers, footers, side-bars. etc.
  - Different page layout templates (*masters*) can be sequenced, e.g.
    - first page followed by later pages
    - recto / verso alternating
- Formatting Object with their properties control
  - text placement
  - fonts, sizes, line height, leading
  - hyphenation, widows / orphans, etc.

slide 52

So What is a Formatting Object?

- An XML element
- With properties that are XML attributes

Examples of Formatting Objects

TYPE OF FORMATTING OBJECT	DESCRIPTION	SAMPLE TAGS

Inline-level FOs	These things are in the same line as the things around them, they do not add new structure or change writing direction, think emphasis, leader dots, font families and size	`<fo:inline>`
		`<fo:wrapper>`
		`<fo:page-number>`

Block-level FOs	Things that are separate from the things around them, not runin. Blocks include paragraphs, titles, and block quotes. Blocks stack one top of each other (or below each other) like children's blocks.	`<fo:block>` `<fo:block-container>`
FOs for Lists	Lists are complex block-type objects, typically composed of list items, which are typically composed of some sort of prefix character (like a bullet, number, letter which XSL calls a label) and then the body of the list item.	`<fo:list-block>` `<fo:list-item>`

FOs for Tables	Rows and columns are different from simple blocks; and therefore have their own special FOs	`<fo:table>`
		`<fo:table-body>`
		`<fo:table-caption>`

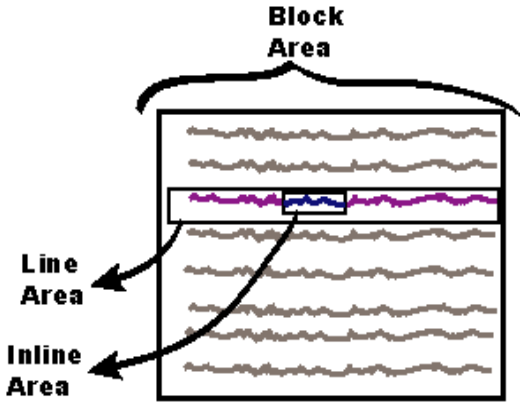
Pages and Layout Objects	Top-level elements that define what the page(s) will look like and give you page sequences like making recto and verso pages different	`<fo:simple-page-master>` `<fo:page-sequence-master>`
Out of Line FOs (includes links)	FOs for footnotes and marginalia, stuff not in the regular stream of text	`<fo:footnote>` `<fo:footnote-body>`

(There are other objects such as graphics and floats)

slide 54

There are Block-type FOs and Inline FOs

- Almost everything is a simple block or an inline
- Lists and tables are blocks with special behaviours



slide 55

Components of an XSL-FO Document

(the, usually virtual, documents made during the transform)

An XSL-FO document has two major parts

- Both parts are enclosed in a top-level element `<fo:root>`
- Part One contains page descriptions (layouts and page masters)
  - general layout of each potential page
  - instructions to the rendering engine on which page templates to use when
  - inside element `<fo:layout-master-set>`
- Part Two contains page sequences (content flows)
  - assign content to pages
  - assign formatting styles (properties) to content
  - inside one or more `<fo:page-sequence>` elements

slide 56

Page Descriptions (Layouts)

(There will be sets of these)

- Describe dimensions and aspects of “one page”
  - in print, this is a single print page (8 ½ by 11, A4, etc.)
  - for web design these can be BIG pages with scrollable regions (HTML-style page)
  - Provides
    - A name, so we can send content to that named page-design
    - page width
    - page margins
    - dimensions of page regions
- Inside element `<fo:simple-page-master>`

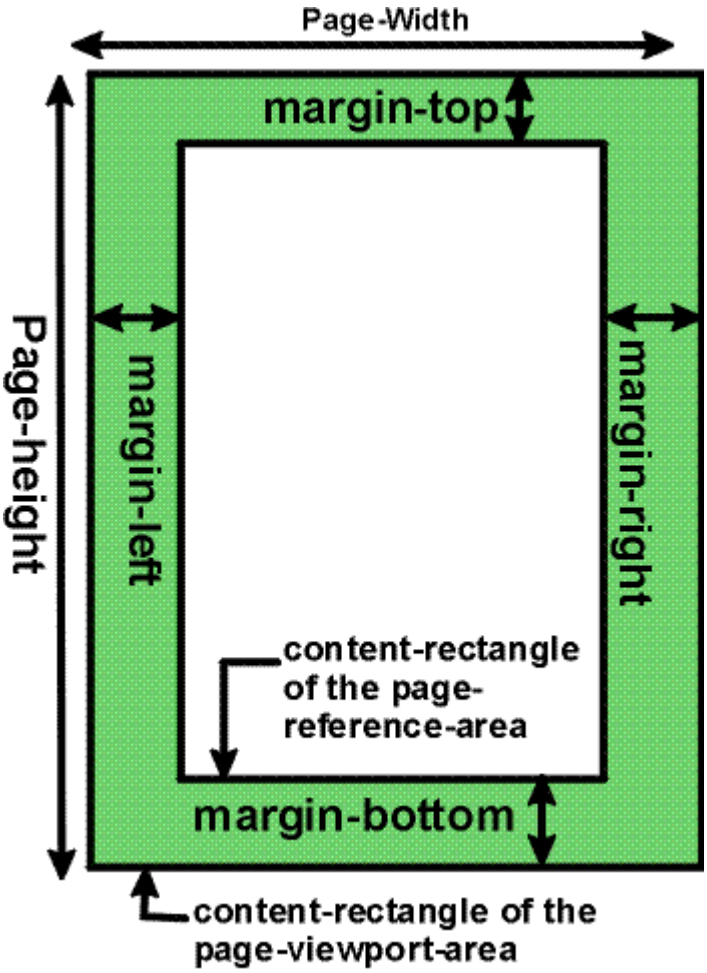
slide 57

Multiple Page Masters

- You can have more than one `<fo:simple-page-master>`
  - with different identifiers
  - Useful if you have regular *sequences* of page layouts
  - E.g., chapter first page followed by alternative left and right hand pages

XSL-FO Page Terminology

- Pages have a height and width, plus four margins (top, bottom, left, right)
- How these properties map to page depends on writing mode and orientation



Graphic from the XSL CR 21 November 2000, 6.4.12

slide 59

Content Rectangle

- Is the area where the text will actually go. That includes:
  - not just the body text with
  - columns and gutters, but also
  - headers and footers
- Is divided into regions
- Regions have names and text can be placed into regions

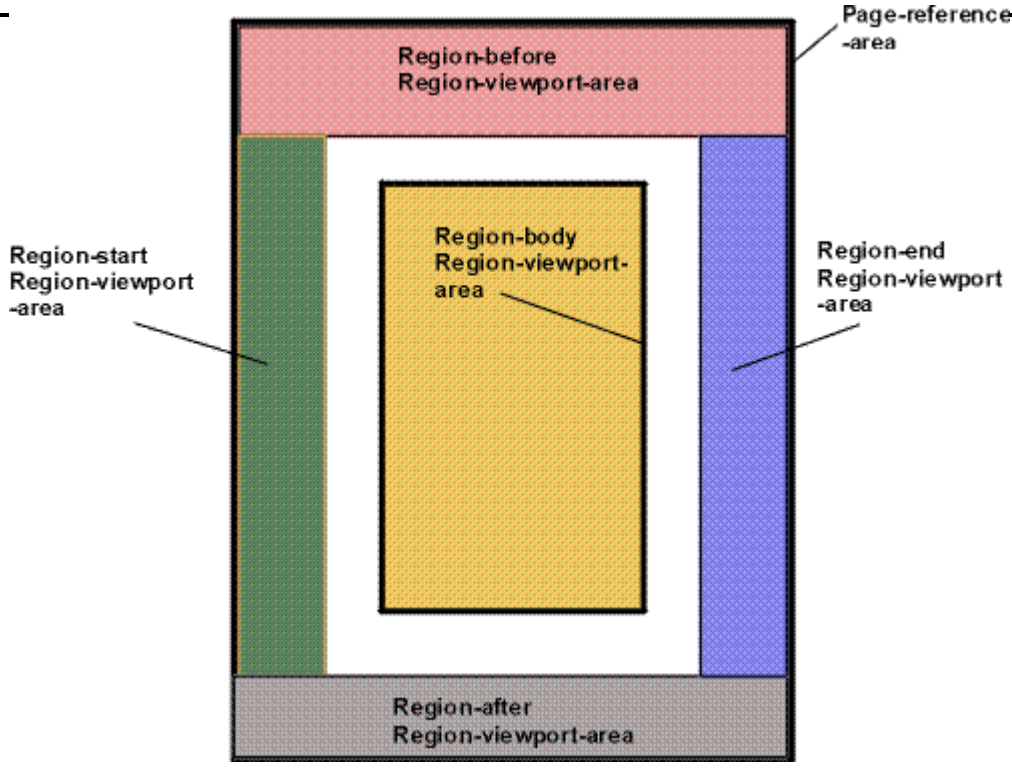
(Not just the body text, this is the whole print region)

slide 60

Geometry of Regions Inside the Content Rectangle

- Content area contains five *region-viewport-areas*
- These are set *inside* the page in the space between the page margins
- Provide space for
  - any content flows (such as paragraphs) or
  - *static content* (such as headers or footers)

Graphic from the XSL CR 21 November 2000, 6.4.12



slide 61

Technical Note: Regions

- Regions can overlap, based on
  - size they are defined to be (`extent`)
  - margin of neighboring region
  - amount of overlap = extent - margin
- Whether text will overflow or be scrolled, clipped, etc., can also be controlled through properties
- The *region-body* region (but not other regions) may contain columns
- Depending on orientation
  (for vertical Japanese, right-to-left Hebrew, etc.)
  - “top” may become left
  - “right” become top, etc.

slide 62

Up to This Point

- We have described pages and pages geometry
- Put page layout is only half the story
- Where is the text, the content?
- Where does the content go?

slide 63

Where the Content Goes

- That's the second half of the stylesheet (program)
- Content goes into Page Sequences
- `<page-sequence>` is a wrapper element for content
- As Dave Pawson expressed it
  "The `page-sequence` element contains the content to fill a sequence of pages"
- Page sequences contain Formatting Objects such as `<fo:block>`s

Page Sequences

- Contain
  - *static content* (the same on every page)
    - for headers and footers (page numbers)
    - name the region they will go into
  - a *flow*
    - the primary stream of content
      (goes from page to page)
    - contains block-level formatting objects (e.g. `<fo:block>`)
    - these objects will “flow” onto the page
      (through the page sequence)
    - names the region into which content will be flowed

slide 65

Content Formatting Happens Inside the Flow

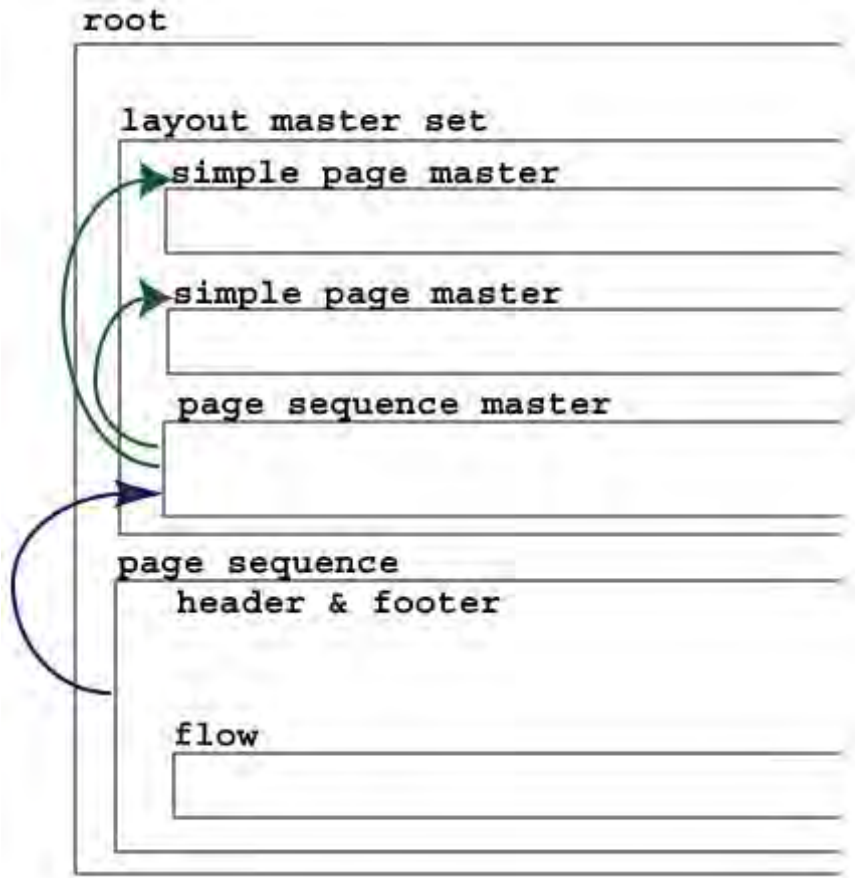
- Various Formatting Objects are placed *inside* the `flow`...
- Usually blocks `<fo:block>`
- Blocks are used to format simple structures
  - Paragraphs
  - Titles
  - Lines
  - Captions
  - Display quotes
  - Section heads, etc. etc.
- Other Formatting Objects are available for more complex structures (lists, tables, footnotes, etc. are more complex)

slide 66

More Complex Pagination

- Point to page-masters (so you know what kind of page gets the content)
- Used for
  - recto/verso differences
  - first page different from all others
  - intentionally blank pages have no header or footer
- In other words: to vary which page masters get chosen when
- Inside the element `<fo:page-sequence-master>`
- Technically
  - specify a sequence of `<simple-page-master>`s
  - that are used for a given `<page-sequence>` (to flow content)

Organization of a Typical XSL-FO Document



slide 68

Blocks Take Typographic Properties

(simplified)

Blocks define the way the text will look through *properties*

- Font
  `font-family`, `font-size`, `font-weight`
- Color
  `color`, `background-color`
- Spacing
  `space-before`, `space-after`
- Line height
- Indents and margins

Blocks may be nested, and inherit properties from ancestors.

slide 69

Block Properties are Attributes of the `<fo:block>` Element

```
<fo:block
          text-align="start"
          margin-left="1.5in"
          space-after="6pt"
          line-height="18pt"
          font-family="sans-serif"
          font-weight="bold"
          font-size="16pt"
          font-weight="bold">Hello, World!</fo:block>
```

slide 70

The Actual Text goes *Inside the Block*

- This XML

```
<recipe><title>Carrots with Ginger Sauce</title>...
```

- Might be put inside a block like this

```
<fo:block font-weight="bold" font-size="14pt">Carrots with
Ginger Sauce</fo:block>
```

slide 71

Let's Watch a Stylesheet Grow

See the formatting of a recipe (or 1000 recipes) spring into shape:

1. Basic page, undifferentiated text
2. Main blocks of content separated
3. Inline elements formatted
4. Ingredients as a table
5. Directions as a list
6. Levels of title distinguished

The XML We Started From

```
<recipe>
<title>Fancy Apple Pie</title>

<class name="type of dish">dessert pie</class>
<class name="main ingredient">apple</class>

<component>
<title>The Pastry</title>

<ingredients>

<ingredient>
<quantity>1 1/2</quantity>
<measure>c</measure>
<foodstuff>flour</foodstuff>
</ingredient>

<ingredient>
<quantity>1/2</quantity>
<measure>tsp</measure>
<foodstuff>salt</foodstuff>
</ingredient>

<ingredient>
<quantity>1/2</quantity>
<measure>c</measure>
<foodstuff>unsalted butter, cold</foodstuff>
</ingredient>

<ingredient>
<quantity>1</quantity>
<measure>tsp</measure>
<foodstuff>sugar</foodstuff>
</ingredient>

<ingredient>
<quantity>3-4</quantity>
<measure>Tbsp</measure>
<foodstuff>cold water</foodstuff>
</ingredient>

<ingredient>
<foodstuff>flour (for sprinkling)</foodstuff>
</ingredient>

</ingredients>

<directions>
<step><p>Make the pie crust in your usual
way. Bake blind 14 min at 400F.</p></step>
```

```
</directions>
</component>

<component>
  <title>The Filling</title>

<ingredient-list>
<ingredient>
<quantity>4</quantity>
<foodstuff>tart apples, peeled</foodstuff>
</ingredient>

<ingredient>
<quantity>1</quantity>
<measure>Tbsp</measure>
<foodstuff>lemon juice</foodstuff>
</ingredient>

<ingredient>
<quantity>2</quantity>
<measure>Tbsp</measure>
<foodstuff>unsalted butter</foodstuff>
</ingredient>

<ingredient>
<quantity>2/3</quantity>
<measure>c</measure>
<foodstuff>sugar</foodstuff>
</ingredient>

<ingredient>
<quantity>3</quantity>
<foodstuff>eggs</foodstuff>
</ingredient>

<ingredient>
<quantity>1</quantity>
<measure>c</measure>
<foodstuff>heavy cream</foodstuff>
</ingredient>

<ingredient>
<measure>pinch</measure>
<foodstuff>nutmeg</foodstuff>
</ingredient>

<ingredient>
<measure>pinch</measure>
<foodstuff>cinnamon</foodstuff>
</ingredient>

</ingredient-list>

<directions>
```

```
<step><p>Grate the apples, and sprinkle with lemon
juice to prevent browning.</p>
</step>

<step><p>Cook the apples and sugar gently in the
butter just to boiling. Simmer until the apples
are soft and little liquid remains. Cool.</p>
</step>

<step><p>Beat the eggs. Beat in the cream, nutmeg,
and cinnamon. Mix in the apples and pour the
mixture into the pie shell.</p>
</step>

<step><p>Bake until the custard is just set, a
bout 30 min.</p>
</step>

</directions>

</component>

<source>Adapted from Angela.</source>

<yield>Serves 6</yield>

<nutrition><p>...</p></nutrition>

</recipe>
```

slide 73

Here is the Unformatted Text Poured onto a Page

Recipe Collection: Breads and Soups Acorn Soup soup stock pinon nuts 1 Tbsp butter 1/2 c pinon nuts 6 Tbsp chopped onions 7 c chicken stock 1 tsp finely ground black pepper 1 qt milk chopped mint to sprinkle on top Saute the pinon nuts and onions. Add the stock, salt, and pepper. Bring to a boil, then reduce the heat to medium and cook until reduced by half, about 15 min. Add the milk and simmer gently until the soup is reduced to 5 cups. Blend to a smooth consistency. Garnish with the mint, and serve. Susan Luscious Acorn Soup is easy to make and greatly heartening on a wintry day. Cream of Squash Soup soup acorn squash whipping cream 2 large acorn squash 1/4 cup butter 1 c chopped onion 6 c chicken broth 2 tsp lemon zest, grated 2 tsp fresh lemon juice 1/4 tsp pepper 1 cup milk Bake the squash. Scoop out the flesh and chop. Saute the onion. Add chicken broth and squash; bring to a boil, then simmer 10 min. Add lemon zest, lemon juice, and pepper. Blend until smooth. Return to heat. Add the milk, and heat gently. Alice Fancy Apple Pie dessert pie apple The Pastry 1 1/2 c flour 1/2 tsp salt 1/2 c unsalted butter, cold 1 tsp sugar 3-4 Tbsp cold water flour (for sprinkling) Make the pie crust in your usual way. Bake blind 14 min at 400F. The Filling 4 tart apples, peeled 1 Tbsp lemon juice 2 Tbsp unsalted butter 2/3 c sugar 3 eggs 1 c heavy cream pinch nutmeg pinch cinnamon Grate the apples, and sprinkle with lemon juice to prevent browning. Cook the apples and sugar gently in the butter just to boiling. Simmer until the apples are soft and little liquid remains. Cool. Beat the eggs. Beat in the cream, nutmeg, and cinnamon. Mix in the apples and pour the mixture into the pie shell. Bake until the custard is just set, about 30 min. Adapted from Angela. This is very like some recipes for pumpkin pie. Serves 6 Cardamon Bread yeast bread 2 pkts active dry yeast 1/4 c warm water 2 c warm milk 3/4 c sugar 1/2 c butter 1-1/2 tsp salt 3/4 tsp ground cardamon 2 eggs 7-8 c all-purpose flour Dissolve yeast in warm water. Add milk, sugar, butter, salt, cardamon, eggs and 3 c flour. Beat until smooth. Stir in enough remaining flour to form a soft dough ball, and knead. Let rise until doubled. Punch down, divide into halves, shape, and let rise. Bake 350° for 25-30 min. Becky 2 loaves Sweet bread yeast bread 1 c milk 3/4 c butter 1 pkt active dry yeas 1/4 c warm water 2 eggs 1 egg, separated 6 c bread flour 3/4 c white sugar 2 tsp lemon zest 1/4 tsp ground mace 1/4 tsp ground nutmeg Scald the milk; add the butter to melt; cool to lukewarm. Dissolve the yeast in warm water and let work. Beat together two eggs with the extra egg yolk. In a separate bowl, combine 4 c flour, sugar, salt, lemon zest, mace and nutmeg. Beat in the milk, yeast mixture and beaten eggs. Let rise until double, about 2 hours. Knead in the remaining cup of flour. Divide, shape, and let rise until double. Brush with the remaining egg white. Bake at 325F 30-40min. James 2 loaves 20 Calories258 Total Fat8.6g Cholesterol51mg Sodium87mg Total Carbs32.8g Dietary Fiber1.1g Protein6.5g Saffron Bread 1-1/2 c milk 1 c butter 1 c white sugar 2 tsp saffron 1/2 c hot water 1/2 oz active dry yeast (2 packets) 2 eggs 2 tsp salt 1/2 tsp ground nutmeg 1 tsp ground cinnamon 2 Tbsp grated lemon zest 6 c all-purpose flour Scald milk; add butter and sugar, and let cool. Soak the saffron in hot water. Combine the milk mixture with the saffron water and eggs. Stir in the yeast, salt, nutmeg, cinnamon, lemon zest and 4 c flour. Add the remaining flour slowly, stopping when the dough holds together. Knead and let rise, about 1 hour. Punch down, divide into three, and shape. Bake at 350F a little under an hour. Mark Rye Bread yeast bread 1/4 c cracked rye 1/4 c water 3/4 c milk 1 c water 1 tsp salt 1/4 c brown sugar 1 egg 3 Tbsp butter 4-1/3 c bread flour 4 tsp gluten flour 1-3/4 tsp active dry yeast 1 Tbsp milk Soak the cracked rye flour in 1/4 c water until most of the water has been absorbed. Start the bread in your usual way. Let rise about half an hour. Punch down, form 2 loaves, brush with milk, and bake at 350F about 45min. Adele This rye bread uses cracked rye and white flour instead of rye flour.

First, Make Some Block Objects

Fancy Apple Pie

The Pastry

1 1/2 c flour
1/2 tsp salt
1/2 c unsalted butter, cold
1 tsp sugar
3-4 Tbsp cold water
flour (for sprinkling)

Make the pie crust in your usual way. Bake blind 14 min at 400F.

The Filling

4 tart apples, peeled
1 Tbsp lemon juice
2 Tbsp unsalted butter
2/3 c sugar
3 eggs
1 c heavy cream
pinch nutmeg
pinch cinnamon

Grate the apples, and sprinkle with lemon juice to prevent browning.
Cook the apples and sugar gently in the butter just to boiling. Simmer until the apples are soft and little liquid remains. Cool.
Beat the eggs. Beat in the cream, nutmeg, and cinnamon. Mix in the apples and pour the mixture into the pie shell.
Bake until the custard is just set, about 30 min.

Add Formatting for the Inline Objects

Fancy Apple Pie

The Pastry

1 1/2 *c* flour
1/2 *tsp* salt
1/2 *c* unsalted butter, cold
1 *tsp* sugar
3-4 *Tbsp* cold water
flour (for sprinkling)

Make the pie crust in your usual way. Bake blind 14 min at 400F.

The Filling

4 tart apples, peeled
1 *Tbsp* lemon juice
2 *Tbsp* unsalted butter
2/3 *c* sugar
3 eggs
1 *c* heavy cream
pinch nutmeg
pinch cinnamon

Grate the apples, and sprinkle with lemon juice to prevent browning.
Cook the apples and sugar gently in the butter just to boiling. Simmer until the apples are soft and little liquid remains. Cool.
Beat the eggs. Beat in the cream, nutmeg, and cinnamon. Mix in the apples and pour the mixture into the pie shell.
Bake until the custard is just set, about 30 min.

Make the Ingredients List into a Table

Fancy Apple Pie

The Pastry

1 1/2	c	flour
1/2	tsp	salt
1/2	c	unsalted butter, cold
1	tsp	sugar
3-4	Tbsp	cold water
		flour (for sprinkling)

Make the pie crust in your usual way. Bake blind 14 min at 400F.

The Filling

4		tart apples, peeled
1	Tbsp	lemon juice
2	Tbsp	unsalted butter
2/3	c	sugar
3		eggs
1	c	heavy cream
	pinch	nutmeg
	pinch	cinnamon

Grate the apples, and sprinkle with lemon juice to prevent browning.
Cook the apples and sugar gently in the butter just to boiling. Simmer until the apples are soft and little liquid remains. Cool.
Beat the eggs. Beat in the cream, nutmeg, and cinnamon. Mix in the apples and pour the mixture into the pie shell.
Bake until the custard is just set, about 30 min.

Display the Steps as a List

Fancy Apple Pie

The Pastry

1 1/2	c	flour
1/2	tsp	salt
1/2	c	unsalted butter, cold
1	tsp	sugar
3-4	Tbsp	cold water
		flour (for sprinkling)

1. Make the pie crust in your usual way. Bake blind 14 min at 400F.

The Filling

4		tart apples, peeled
1	Tbsp	lemon juice
2	Tbsp	unsalted butter
2/3	c	sugar
3		eggs
1	c	heavy cream
	pinch	nutmeg
	pinch	cinnamon

1. Grate the apples, and sprinkle with lemon juice to prevent browning.
2. Cook the apples and sugar gently in the butter just to boiling. Simmer until the apples are soft and little liquid remains. Cool.
3. Beat the eggs. Beat in the cream, nutmeg, and cinnamon. Mix in the apples and pour the mixture into the pie shell.
4. Bake until the custard is just set, about 30 min.

Distinguish Among Different Levels of Title

Fancy Apple Pie

The Pastry

1 1/2	*c*	flour
1/2	*tsp*	salt
1/2	*c*	unsalted butter, cold
1	*tsp*	sugar
3-4	*Tbsp*	cold water
		flour (for sprinkling)

1. Make the pie crust in your usual way. Bake blind 14 min at 400F.

The Filling

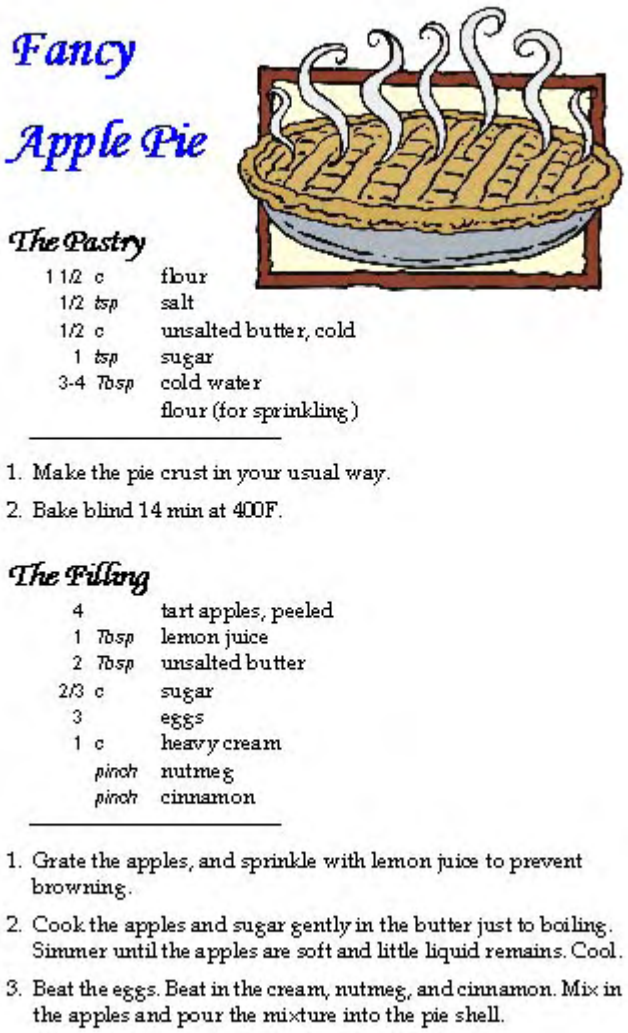
4		tart apples, peeled
1	*Tbsp*	lemon juice
2	*Tbsp*	unsalted butter
2/3	*c*	sugar
3		eggs
1	*c*	heavy cream
	pinch	nutmeg
	pinch	cinnamon

1. Grate the apples, and sprinkle with lemon juice to prevent browning.
2. Cook the apples and sugar gently in the butter just to boiling. Simmer until the apples are soft and little liquid remains. Cool.
3. Beat the eggs. Beat in the cream, nutmeg, and cinnamon. Mix in the apples and pour the mixture into the pie shell.
4. Bake until the custard is just set, about 30 min.

Now Make It Prettier

the stylesheet writer could

- Add rules
- Add graphics
- Play with fonts
- Make it the design your designer dreamed up



Fancy Apple Pie

The Pastry

1 1/2	c	flour
1/2	*tsp*	salt
1/2	c	unsalted butter, cold
1	*tsp*	sugar
3-4	*Tbsp*	cold water
		flour (for sprinkling)

1. Make the pie crust in your usual way.
2. Bake blind 14 min at 400F.

The Filling

4		tart apples, peeled
1	*Tbsp*	lemon juice
2	*Tbsp*	unsalted butter
2/3	c	sugar
3		eggs
1	c	heavy cream
	pinch	nutmeg
	pinch	cinnamon

1. Grate the apples, and sprinkle with lemon juice to prevent browning.
2. Cook the apples and sugar gently in the butter just to boiling. Simmer until the apples are soft and little liquid remains. Cool.
3. Beat the eggs. Beat in the cream, nutmeg, and cinnamon. Mix in the apples and pour the mixture into the pie shell.

slide 80

Who's Using XSL-FO and What is It Good For

- Who's using XSL-FO now?
- What do they gain?
- When does XSL-FO not make sense?
- Why isn't everyone using XSL-FO?
- When does XSL-FO make very good sense?

slide 81

XSL-FO Software Vendors Name Their Clients

(small random sample)

Visa International	Bank of America	Mastercard
European Customs Union	Pratt & Whitney	Nike Customer Service Center
Bell South	TransCanada Pipelines Limited	SAS
Cisco	Siemens AG	Semantec
Credit Swiss	Hewlett-Packard	State of Oregon (legislature)
Princess Cruise Lines	State of Iowa Motor Vehicles	Intermountain Gas Company
ConnXion	Minneapolis Public Schools	Boeing
SAP	Ami	Airbus
Lockheed Martin	HSBC	Network Appliances

slide 82

What the Commercial XSL-FO Vendors Claim

- They can do 70-85% of traditional high-end composition
- 2002 — XSL-FO could probably meet 30-40% of publishing needs
- 2005 — 80–85% (they will claim higher)
- That their clients have "*modified their [formatting] requirements* to enable them to live with the [XSL-FO] limitations"

slide 83

Companies are Using XSL-FO When

They want advantages of XML and

- Speed is more important than extreme quality
- High-end composition is too expensive
- Volume is too large for speed of human layout
- Need batch pagination (multitudinous documents pulled from a database and poured into templates)
- Output process has to be automated ("lights out")
- Need fast proofs, before conventional page-layout (WYSIROTS)

slide 84

XSL-FO Really Shines When

- Speed of production is critical (so many transactions that if people intervene it could not be run in real time) (credit card statements)
- Large volume of similar documents
- All (or most) layout decisions can be stated as rules
- People checking the final output does not add value (bank statements)
- Document format is easily repeatable
- Content can be validated before processing
- Multilingual content that changes writing direction frequently

slide 85

Ideal XSL-FO Candidates

(XSL-FO is a Great Report-writer)

- Content-driven document
- Format on-demand for a customer
- In large batches
- For example
  - credit card and bank statements
  - catalogs, directories, indexes, listings
  - bank statements
  - insurance policies and claims
  - contracts
  - legal agreements
  - investment portfolios
  - telephone books
  - hospital systems reports
  - patient medical charts

slide 86

Candidates also Include

(It's Not Just For Reports)

- Bills, resolutions, reports of state legislatures
- Journal articles and other short materials
- Textbooks
  - where there is page after page of tight textual material
  - where the same basic content (and ancillary products) varies state by state
- Books with regional variations
- Manuals (software, owner, maintenances)
- Reference books

slide 87

Why Isn't Everybody Using XSL-FO?

By design, XSL-FO is desk-top publishing, *not fine typography*

- XSL-FO spec doesn't express all of typography
- XSL-FO engines may only produce desktop quality (or less)
  - quality of H&J
  - page layout and vertical justification
  - floats, keeps, widows, and orphans still a little weak (improving every year)
  - no CMYK, few pantones

slide 88

When XSL-FO Is Not Useful

- Your content is not in XML, and is not likely to be
- Layout designers add value spread-by-spread
  (*Wired* Magazine, picture books)
- Your formatting requirements aren't supported by:
  - the XSL-FO vocabulary
  - an existing XSL-FO engine (e.g., support for massive tables varies)
  - your tagging

(last bullet is the Achilles heel of XSL-FO [or any other] automation
the computer can't know what you don't tell it)

slide 89

Really Bad XSL-FO Candidates Include

- Any content where page design does not flow from the tagging
- Grade school math texts (every spread is individual layout art)
- High-design magazines
- Most advertising circulars
- Material that is art not mass production
- Fancy newspaper layout with multiple continuations
- One-off broadsheets

slide 90

You Might *Not* Want XSL-FO If…

- Content is well-formed XML with no DTD or schema
  (too much variation; too many surprises)
- Page-limited productions, where text will be altered to fit the space
- Most text changes are made *after* layout
  (Unless you are flowing your FO into an interactive tool that let's you tweak)
- DTD or schema changes constantly
  (XSL-FO stylesheet would need constant modification to keep up)
- Content is highly variable in structure
  (structure has loads of options)

slide 91

Additional Planning Challenges

- Solely or mostly hands-off process doesn't suit every layout
- Writing typographic instructions for a computer to perform is
  - very different than *doing* the layout
  - very different from making a mockup or naming Quark styles
  - a logic task (programmer) not a graphic design task
- You need both a typographic designer and a programmer
  (rarely the same person)
- Desktoppers trained on *applications* (QuarkXPress, Photoshop)
  not in typography

slide 92

XSL-FO Tools

- Rendering engines
- XSL-FO editors
- Online previewers
- Composition systems that accept XSL-FO input
- Two cost categories
  - free and open source
  - commercial

slide 93

Free XSL-FO Formatting Engines

(representative stand-alone, free command-line engines)

- FOP
  - an Apache project
  - built into several XML editing applications, e.g., Oxygen, TopXML, Editix
- Sun's xmlroff. Open-source.
- IBM's XFC (XSL Formatting Objects Composer)
- SourceForge's jFor project
- Unicorn Formatting Objects: C++ application that makes TeX
- Chive Apoc XSL-FO renderer (a full port of FOP for .NET)

slide 94

Commercial XSL-FO Formatting Engines

(representative commercial GUI software)

- AntennaHouse XSL Formatter
  (plugins for XMLSPY, others in the works)
- RenderX XEP Rendering Engine
  (built into XSLFast, which can produce as PDF or deliver to InDesign, QuarkXPress, FrameMaker)
- Scriptura and Scripturaxbos
- Ibex
- Ecrion XF Rendering Server
- Xinc (Swing XSL-FO viewer)

slide 95

Page-layout Applications that Understand XSL-FO

(selected)

- Arbortext 3b2-FO
- XyEnterprise XPP
- Arbortext Epic
- FrameMaker within Adobe Document Server

slide 96

More Support Tools All the Time

- DTD-specific XSL-FO application for NewsML/NITF
- RTF to XML or to XML + XSL-FO
- HTML to XSL-FO
- CSS to XSL-FO
- HTML to PDF (using FOP)
- Microsoft Word's HTML or their XML to XSL-FO
- Document servers from Saffron, Adobe
- ebXML/UBXML stylesheet generators

slide 97

You Get What You Pay For

- Commercial engines
  - are more powerful than free ones
  - have better language/alphabet/hyphenation support
  - have been built into editors and other products
- Free, shareware, and open source
  - are still quite useful
  - have been built into editors and other products

There are Software Limitations

- Different products have different strengths
- Nobody implements *all* of the very large XSL-FO spec
- Products add non-compatible, non-spec *useful* extensions
- XSL-FO stylesheets are not 100% interchangeable
  - looks great from product A
  - looks lousy from product B
  - blows up in product C
- Support varies for
  - forms and fill-in-the-blank
  - equations (MathML or TeX or both)
  - tables
  - writing direction and hyphenation

slide 99

XSL-FO Spec Sets Levels of Conformance

- Three Levels
  - **basic** — Minimum level of pagination or aural rendering
  - **extended** — Everything else for sophisticated pagination except some conveniences
  - **complete** — Everything (Example: font-selection-strategy)
- Not as useful as they sound
  - because there are parts nobody does
  - so actual levels are really higher than reported levels
- Applications generally provide
  - Basic conformance
  - Some part of Extended conformance; and
  - Conformance table
    including any extensions

(Check with your vendor, read the conformance table, then ask their clients)

slide 100

Conformance Examples

(as of December 2005)

- XEP has not yet implemented `table-omit-footer-at-break`
- AntennaHouse has not yet implemented `font-selection-strategy`
- fop has not yet implemented all of the Basic conformance items
- Almost no one had implemented any aural properties so far

(In general what they *do have* is more important than what they *don't*)

The Wrapup

slide 101

What Does It Take to *Create* XSL-FO

- An XSL-FO engine (software purchase)
- Source documents in XML that need making into pages
- Designers (pages still need to look readable and pleasing)
- Programmers to write the XSLT/FO transforms (can be contracted out)
- Desire to change to an XML workflow
  - get simple results very easily
  - with work, can develop into something quite sophisticated

slide 102

What You Need to Know to Write an XSL-FO Stylesheet

- XML source tags (what are they, what do they mean)
- Target page design (look, behavior)
- The XSL-FO vocabulary (how to make the desired effects)
- Some XSLT, maybe quite a lot (how to do the transform)
- Relation between XML source and desired effects
  (typically expressed as a specification, like a typesetting specification)

slide 103

What Does It Take to *Maintain* an XSL-FO Stylesheet

(a lot less)

- Source documents in XML
- Changes you need to make to the formatting
- Knowledge of typography to accomplish those changes
- Enough XSLT to *understand* the XSL-FO stylesheets
- Enough XSL-FO vocabulary to produce the changes
- An XSL-FO engine to run the new stuff

slide 104

How do You Train an XSL-FO Programmer

(Hint: It is more critical to know typesetting than programming)

- They must know XML and XSLT
- Download an evaluation XSL-FO engine (free for a while)
- Read any tutorials and examples that come with the engine
- Program something simple (not live data)
- Download the XSL spec from W3C and print it (it's big!)
- Take a course (many available)
- Buy
  - Michael Kay's XSLT Programmer's Reference (and/or haunt Zvon)
  - One of the XSL-FO books (for reference)
- Read some GOOD code (e.g., Ken Holman's UbXML stylesheets)
- Read the XSL list and XSL-FO lists and ask questions

* URLs for all the above are in the appendices

slide 105

There is Other Cool XSL-FO Material

(no time to mention in 3 hours)

- XSL-FO plus SVG for dynamic web-publishing
- XSL-FO for bar codes
- XSL-FO drives the screen display of an XML editing tool
- XSL-FO plays chess (or at least displays chess)

slide 106

The Hype Behind XSL-FO

(let's re-examine those myths)

Specify your pages *once* and make *many* outputs (PDF, Postscript, web, audio all one stylesheet)	Not on my planet. You want lots of different stylesheets; not just one PDF but many.
Publish lights-out, no human necessary.	For many types of content, not all.
All pagination systems will accept XSL-FO input.	Hasn't happened yet, but some do and more all the time.
These pages are *just as good as* fine typesetting.	Well…no.
These pages are good enough.	Quite possibly.
Cheaper, faster, better	Probably, definitely, and (hey) two out of three ain't bad! Who defines "better" anyway?

slide 107

Some Real XSL-FO-produced Pages

- We'll show some made with XSL-FO to illustrate
  - range
  - complexity
  - what we could find that was not behind fire walls
- If there's time we'll make some pages
  It's fast and easy

slide 108

Some Real XSL-FO-Made Pages

- Conference papers, for example:
  `http://www.idealliance.org/proceedings/xml05/ship/160/piez-xml2005-ideadb.PDF` ("XSLT throughout the document lifecycle")
- Short PDF samples from RenderX, e.g., Color table in PDF
  `http://xep.xattic.com/xep/testsuite/features/color.pdf`
- Short PDF samples from Antennahouse, e.g., glyph orientation
  `http://www.antennahouse.com/XSLsample/pdf-v32/GlyphOrientation.pdf`

slide 109

Information Resources: Where to Learn More

slide 110

Books

- *XSL Formatting Objects: Developer's Handbook*, Doug Lovell, 2002
- Elliotte Rusty Harold, chapter. 18 of *The XML Bible*
  `http://www.cafeconleche.org/books/bible2/chapters/ch18.html`
- *XSL-FO: Making XML Look Good in Print* , Dave Pawson, 2002
- *Practical Formatting Using XSL-FO*, Ken Holman

slide 111

Articles and Reports

- Gilbane Report: Overview article (Jan/2004)
  `http://www.gilbane.com/gilbane_report.pl/94/XSLFO_Ready_for_Prime_Time.html`
- About.com's article on XSL-FO
  `http://webdesign.about.com/cs/xslinformation/a/aaxslfo.htm`
- 3Schools XSL-FO tutorial
  `http://www.w3schools.com/xslfo/default.asp`
- PerfectXML.com tutorial **`http://www.perfectxml.com/XSLFO.asp`**
- XML Pitstop's list of many articles
  `http://www.xmlpitstop.com/ListResourcesByType/DispContentType/XSL-FO/PageNumber/1.aspx`
- W3C's XSL-FO resource page **`http://www.w3.org/Style/XSL/`**
- Antennahouse tutorial and examples
  `http://www.antennahouse.com/XSLsample/XSLsample.htm`
- Ecrion tutorial **`http://www.ecrion.com/XF/TechnicalResources/XSL-FO%20Tutorial.aspx`**
- DevX article on using XSL-FO with Java servlet
  `http://www.devx.com/xml/Article/16430/1954?pf=true`

- Ibex tutorial on using Ibex with SQL Server
  http://www.xmlpdf.com/ibex-sql.html
- XSL-FO Chef's Tool Exhibition
  http://www.idealliance.org/proceedings/xml03/slides/xslfoshowcase
- "What Is XSL-FO and When Should I Use It?" (Seybold Report, Dec. 2002)
  http://www.adobe.com/products/server/documentserver/pdfs/adobeseybold_xsl-fo.pdf

---

slide 112

Online Resources

- W3C list of processors, with notes: **http://www.w3.org/Style/XSL/**
- Diderot Track XSL-FO list of processors, with notes and conformance comparison table:
  http://www.diderottrack.nl/en.articles.xslfo.html
- XSL Info site **http://www.xslfo.info/**
- The lists:
  - W3C mailing list **http://lists.w3.org/Archives/Public/www-xsl-fo/**
  - XSL-List (XSLT and XSL-FO)
    http://www.mulberrytech.com/xsl/xsl-list/
  - Yahoo XSL-FO group **http://groups.yahoo.com/group/XSL-FO/**

---

slide 113

XSL-FO Stylesheets Online

- AntennaHouse sample files
  http://www.antennahouse.com/XSLsample/XSLsample.htm
- Docbook **http://docbook.sourceforge.net/projects/xsl/**; also a book about Docbook XSL **http://www.sagehill.net/book-description.html**

- DocBook stylesheet
  Wiki**http://wiki.docbook.org/topic/DocBookXslStylesheets**
- RenderX FO-features demonstration with the XSL-FO stylesheets
  http://www.renderx.com/featurestest.html
- AntennaHouse WordMLtoFO stylesheet ($980)
  http://www.antennahouse.com/product/wordmltofo.htm
- Crane Softwrights XSL-FO for UBL
  http://www.cranesoftwrights.com/resources/ublss/
- Microsoft/RenderX XSL-FO files for converting WordML to XSL-FO
  http://msdn.microsoft.com/office/default.aspx?pull=/library/en-us/odc_wd2003_ta/html/OfficeWordWordMLtoXSL-FO.asp

---

slide 114

For Programmers

- FAQs and Specs
  - News, examples, software links, and book links at xslfo.info
    http://www.xslfo.info/
  - ZVON XSL-FO Reference (indexed to the Rec)
    http://www.zvon.org/xxl/xslfoReference/Output/
  - The XSL-FO Recommendation **http://www.w3.org/TR/xsl/**
  - Dave Pawson's XSL-FO QA snippets
    http://www.dpawson.co.uk/xsl/sect3/index.html
  - Instructions for installing two free XSL-FO processors, FOP and PassiveTeX:
    http://www.sagehill.net/docbookxsl/InstallingAnFO.html
- ListServs to ask Questions
  - XSL-List (XSLT and XSL-FO)
    http://www.mulberrytech.com/xsl/xsl-list/
  - Yahoo XSL-FO group **http://groups.yahoo.com/group/XSL-FO/** or
  - W3C FO list archives **http://lists.w3.org/Archives/Public/www-xsl-fo/**

Colophon

- Slides and handouts created from single XML source
- Slides projected from HTML which was created from XML using XSLT
- Handouts created from XML:
  - Source XML transformed to Open Office XML
  - Open Office XML opened in Open Office
  - Pagination normally adjusted
- Slideshow materials available at:
  `http://www.mulberrytech.com/slideshow`

The Most Basic Formatting Properties

Block Properties (`<fo:block>`)	
margin-top -bottom -left -right	[7.10.1 - .4]
border-before -after -start -end	[7.7.7 - .30]
padding-before -after -start -end with -color -style -width	[7.7.31 - .38]
space-before -after; start- end-indent	[7.10.5 - .8]
line-height	[7.15.4]
text-align, text-align-last	[7.15.9 - .10]
text-indent	[7.15.11]
break-before -after	[7.19.1 - .2]
keep-together -with-next with-previous	[7.19.3 - .5]
orphans, widows	[7.19.6 - .7]

Text Properties (`<fo:wrapper>`)	
font-family	[7.8.2]
font-size	[7.8.4]
font-style	[7.8.7]
font-weight	[7.8.9]
text-decoration	[7.16.4]
text-transform	[7.16.6]
color	

There are many other properties for blocks and wrappers.

Note in particular that an `<fo:block>` *will* also accept font-related attributes.

Tip: Check the compliance table for the XSL-FO formatting engine you use for both its limitations and its extensions.