

# Introduction to XML

David J. Birnbaum



## Before we begin

Launch <oXygen/>

If you haven't yet installed <oXygen/>:

- Go to <http://www.oxygenxml.com>
- Download and install
- Follow instructions to request free 30-day evaluation license

# Session 1: Introduction to XML

El'Manuscript, Vienna, 2018-09-16

David J. Birnbaum  
djbpitt@gmail.com  
<http://www.obdurodon.org>

# Outline

## Sample projects

- Manuscript description: Repertorium
- Manuscript transcription: Codex Suprasliensis

## XML and text

- Overview
- Pseudo-markup and markup
- Elements
- Attributes
- Well-formedness

## The creation of a digital text

Editing XML in <oxyen/>

Hands-on practice editing XML

# Sample projects

## Manuscript description

- Repertorium of Old Bulgarian literature and letters
- Institute of Literature, Bulgarian Academy of Sciences
- <http://repertorium.obdurodon.org>

## Codex Suprasliensis

- Institute of Literature, BAS
- <http://suprasliensis.obdurodon.org> (edition)
- <http://csup.ilit.bas.bg/> (project)

# Overview

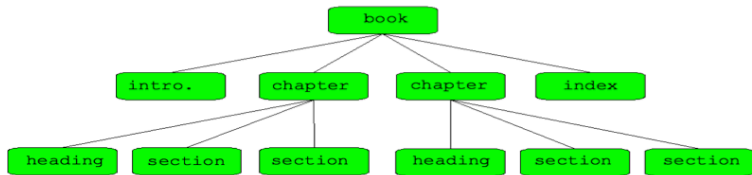
OHCO: ordered hierarchy of content objects

Three views of XML

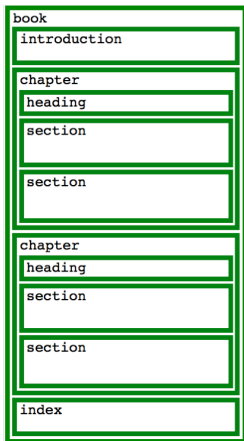
- Tree (hierarchy of objects)
- Nested boxes (objects contain other objects)
- Serialization (string of characters)

# 1. Tree (hierarchy of objects)

---



## 2. Nested boxes (objects contain other objects)





### 3. Serialization (string of characters)

```
<book>
  <introduction> ...</introduction>
  <chapter>
    <heading> ...</heading>
    <section> ...</section>
    <section> ...</section>
  </chapter>
  <chapter>
    <heading> ...</heading>
    <section> ...</section>
    <section> ...</section>
  </chapter>
  <index> ...</index>
</book>
```

## Hamlet, First quarto, 1603

*Enter Hamlet.*

*Cor.* Madame, will it please your grace  
To leaue vs here?

*Que.* With all my hart. *exit.*

*Cor.* And here *Ofelia*, reade you on this booke,  
And walke aloofe, the King shal be vnscene.

*Ham.* To be, or not to be, I there's the point,  
To Die, to sleepe, is that all? I all:  
No, to sleepe, to dreame, I mary there it goes,  
For in that dreame of death, when wee awake,  
And borne before an euerlasting Iudge,  
From whence no passenger euer returnd,  
The vndiscovered country, at whose sight  
The happy smile, and the accursed damn'd.  
But for this, the ioyfull hope of this,  
Whol'd beare the scornes and flattery of the world,  
Scorned by the right rich, the rich curst of the poore?

The

*Enter Hamlet.*

*Cor.* Madame, will it please your grace

To leaue vs here?

*Que.* With all my hart. *exit.*

*Cor.* And here Ofelia, reade you on this booke,

And walke aloofe, the King shal be vnseene.

*Ham.* To be, or not to be, I here's the point,

To Die, to sleepe, is that all? I all:

No, to sleepe, to dreame, I mary there it goes,

For in that dreame of death, when wee awake,

And borne before an euerlasting ludge,

From whence no passenger euer retur'nd,

The vndiscovered country, at whose sight

The happy smile, and the accursed damn'd.

But for this, the ioyfull hope of this,

Whol'd beare the scornes and flattery of the world,

Scorned by the right rich, the rich curssed of the poore?

The

# Pseudo-markup

## *Hamlet*

- Stage directions
- Speeches
- Speakers
- Other characters
- Metrical lines

## General

- Paragraph spacing and indentation
- Centering and bolding of titles
- Hanging indentation for bibliographic lists
- Italics for emphasis, foreign words, book titles, etc.

# The XML view of content and markup

Content is the textual data

- Transcribed from source (e.g., a manuscript)
- Created by the editor (e.g., manuscript catalogue)

Markup describes the role of different data components

No *pseudo markup* in your content

- No editorial parentheses, square brackets, angle brackets, slashes and backslashes, italics, etc.

# XML building blocks

Textual (character data) content

Elements

- Structural components of the document

Attributes

- Properties of elements

# Elements

Elements have matching start and end tags

- `<title> ... <title/>`

Some elements are empty and self-closing, e.g., `<bookmark/>`

Element names must begin with a letter and may contain letters, digits, and underscores (no spaces; no other punctuation)

- Underscore: `<personal_name>`
- Camel case: `<personalName>`

Attribute names have the same constraints as element names

# The “X” in XML

## eXtensible Markup Language

- The user determines the tag set
- Pro: you determine how to characterize your data
- Con: you are responsible for determining how to characterize your data

## You decide

- What to tag
- How to tag it (what to call it)



## Three types of markup

Descriptive: what the object is

- E.g., emphasized
- `<em>yes !</em>`

Presentational: what the object looks like

- E.g., italicized
- `<i>yes !</i>`

Procedural

- What the machine should do
- E.g., unload the Roman font film strip and load the italic one

# Why Digital Humanities projects use descriptive markup

Same formatting may represent various semantics

- Italics: emphasis, foreign, book title, etc.

Same semantics may be formatted variously

- Emphasis: italic, bold, loud (audio device), etc.

Separation of levels: content and presentation

- Encode descriptively
- Transform to presentational final form for rendering (HTML, PDF, etc.)

Multipurposing

- Format the same content objects in different ways for different purposes

# Texts and trees

## Why XML looks at texts as trees

- Computers can traverse trees quickly
- Documents *are* hierarchical, right?

## Hierarchical challenges

- Multiple, overlapping hierarchies
  - Physical hierarchy: folios, lines
  - Intellectual hierarchy: texts (with subelements: chapters, sections, paragraphs, etc.)
- Relationships at a distance
  - Cross-references and other internal pointers
  - References and pointers to other (external) documents

# Attributes

Qualifying information about elements

Encoded inside the start tag, after the element name

- Attribute name="value" pair
- `<place xml:lang="fr">Paris</place>`
- `<title type="journal">Scripta &amp;  
e-Scripta</title>`

Attribute names are subject to the same rules as element names

Attribute values must be quoted (matching single or double straight quotation marks)

## An XML document must be *well-formed*

Single root element

Proper nesting (no overlapping tags)

- Good: `<em><foreign>oui!</foreign></em>`
- Bad: `<em><foreign>oui!</em></foreign>`

Name and name start characters for element and attribute names

Attribute values must be quoted (single or double)

Reserved characters must be encoded as entities

- `& = &amp;`
- `< = &lt;`
- `> = &gt;`

Indentation is for human convenience

# Sample

```
<book>
  <author>Michael Kay</author>
  <title edition="4">XSLT 2.0 and XPath 2.0 Programmer's
Reference</title>
  <published date="2008">
    <publisher>John Wiley & Sons, Inc.</publisher>
    <pubPlace>10475 Crosspoint Boulevard, Indianapolis, IN
46256</pubPlace>
  </published>
  <ISBN num="978-0-470-19274-0"/>
  <dedication>
    <i>
      <b>To Anyone Who Uses This Book To Make the World a
Better Place</b>
    </i>
  </dedication>
</book>
```

## What's wrong?

```
<author>Michael Kay</author>
<title edition = 4>XSLT 2.0 and XPath 2.0
  Programmer's Reference</title>
<published date = 2008>
  <publisher>John Wiley & Sons, Inc.</publisher>
  <pubPlace>10475 Crosspoint Boulevard, Indianapolis,
    IN 46256</pubPlace>
</published>
<ISBN num="978-0-470-19274-0">
<dedication>
  <i><b>To Anyone Who Uses This Book
    To Make the World a Better Place</i></b>
</dedication>
```

# Creating a digital text

## In theory

- Document analysis, then ...
- Schema development, then ...
- Markup

## In practice

- The preceding is a cycle, and not a sequence
- Markup is part of the process of document analysis

## Nonetheless

- Start with document analysis, not with angle brackets



## Why use an XML editor?

<oXygen/> (<http://www.oxygenxml.com>)

XML-aware

- Real-time and on-demand validation
- Completion hinting
- Multiple views
- (Schema-aware ... stay tuned)

IDE (integrated development environment)

- XSLT (eXtensible stylesheet language transformations)
- Debugger
- Other development tools

## Editing XML in <oXygen/>, p. 1

### Create a new file

- File -> New -> New document -> XML document
- Short cuts: Ctrl+n; leftmost icon at top of screen

### Create an element

- Type a start tag (in angle brackets)
- <oXygen/> automatically creates the matching end tag

### Change an element

- Change the start tag; the end tag changes automatically to match

## Editing XML in <oXygen/>, p. 2

### Wrap text in an element

- Select the text, type Ctrl+e (for 'element'), type the element name
- To use the same wrapper as last time, select and type Ctrl+/,

### Split an element

- Put the cursor at the split point and type Shift+Alt+d

### Pretty-print (wrap) the text

- Shift +Ctrl+p; pretty-print (indentation) icon

# Hands on

- Choose a document with a regular structure
- Copy into a new XML document in <oXygen/>
- Mark it up in XML

## Choose a document with a regular structure

Google a recipe for your favorite food

Find a menu from your favorite restaurant

Encode a letter by Oscar Wilde

<http://law2.umkc.edu/faculty/projects/fttrials/wilde/lettersfromwilde.html>

Encode a sonnet by William Shakespeare

<http://www.shakespeares-sonnets.com/all.php>

... or choose your own text

## Copy into a new XML document in <oXygen/>

- Select all of the text (Ctrl+a, or use the mouse)
- Copy (Ctrl+c)
- Open a new document in <oXygen/> (Ctrl+n, select “XML document”)
- Paste (Ctrl+v)

## Mark it up in XML

- Imagine a research or other context where you're marking up your text for a reason
- Identify and tag *major structural components*
- Identify and tag *small, in-line items* that might be useful