

Introduction to XML and Linked Data

School of Library and Information Studies ♦ University of Wisconsin at Madison
Dorothea Salo ♦ salo@wisc.edu ♦ 4261 Helen C. White Hall ♦ (608) 265-4733
Spring 2013 ♦ Online



To the extent possible under law, the person who associated CCo with this work has waived all copyright and related or neighboring rights to this work. This work is published from: United States.

Description

This course offers an introduction to the principles and syntax(es) of XML and linked data, sufficient to permit further exploration of their use in library contexts such as digital-projects metadata and cataloging. By its conclusion, learners will be able to:

- ✓ Create, edit, parse, validate, and repair an XML document
- ✓ Create a valid MODS record
- ✓ Read basic RDF in Turtle and RDF/XML syntax, as well as in graph form
- ✓ Find and use URIs from common library/archives vocabularies and other datastores (such as VIAF)

Textbooks and software

I recommend but do not require the purchase of *XML in a Nutshell* (by Elliotte Rusty Harold and W. Scott Means, O'Reilly Media 2004; should be readily available used).

You **will need** an XML-editing environment to take this class. If you have computer-programming experience already, a programmer's editor plus command-line or web-based parsing/validation tools may suffice, but most people will prefer the experience of an environment designed for XML. If you can afford to buy an editor, I recommend oXygen/ (cross-platform; <http://www.oxygenxml.com/>), which many librarians and archivists use. All XML-editing screenshots I create for the course will be from oXygen/.

If you need a free editor, I suggest installing one of the following:

- ✓ XML Notepad (Windows; <https://xmlnotepad.codeplex.com/>)
- ✓ XML Copy Editor (Windows or Ubuntu Linux: <http://xml-copy-editor.sourceforge.net/>)
- ✓ EditiX XML Editor Free (cross-platform; <http://free.editix.com/index.html>)
- ✓ Jaxe (cross-platform; <http://sourceforge.net/projects/jaxe/>)

You may of course use any capable text/programmer's/XML editor that you prefer. Do not, however, try to do assignments for this course in a word processor!

Communication

Please use the Learn@UW forums to ask *all* questions that do not involve personal circumstances. You are encouraged to answer your colleagues' questions when I don't get to them first. This includes:

- ✓ syllabus questions and clarifications
- ✓ questions about the material
- ✓ questions and clarifications about assignments

I *cannot* respond to these questions when asked by email. Please allow your curiosity to benefit your fellow learners.

Grading and due dates

This is a hands-on, assignment-based class. You are expected to work through assigned problemsets *during the week they are assigned*, turning in screenshots, documents, or other evidence of completion as instructed. You should expect to spend six hours per week *minimum* reading materials and working on assignments.

I will review a given week's assignments on **Wednesday** of the following week. ***I will not review them after that; assignment extensions are not available.*** If you know you cannot give sufficient time to the course while it is ongoing, please discuss a refund with UW Continuing Education within the first week of class. You may, however, work ahead if you wish; I will do my best to have at least two weeks' additional material posted.

Weekly learning goals

Weekly work may vary from the lists below, as I learn from you what concepts and techniques need more time, scaffolding, or practice.

All readings are linked from the course pages in Learn@UW.

Week 1: Basic XML syntax

Learning objectives: Elements, tags, attributes, attribute values. Legal tag and attribute names. Root element. Nesting. XML well-formedness; fixing well-formedness errors. XML parsing.

Week 2: Namespaces, DTDs, schemas, XML validation

Learning objectives: What DTDs and XML schemas are. How to reference them within an XML document. How to find and use documentation and sample instances for individual XML languages. XML namespaces. XML validation; fixing validation errors. (N.b. creating and reading DTDs and schemas is out of scope for this course!)

Week 3: More on MODS

Learning objectives: Producing valid MODS instances. Crosswalking MODS from MARC/AACR2/ISBD; differences in philosophy. Using MODS in digital libraries.

Week 4: What is linked data? RDF, URIs, vocabularies

Learning objectives: What is RDF, and what problems does it try to solve? RDF triples: subject, predicate, object. URIs and literals in RDF/linked data. Examples of library vocabularies retooling toward RDF/LD (VIAF, Dewey Decimal Classification, Dublin Core).

Week 5: Turtle syntax, RDF graphs

Learning objectives: Translating between RDF triples and an RDF graph. Namespace abbreviations in RDF graphs. Basic Turtle RDF syntax. What happens to “the record” in RDF/LD? The “open world assumption.”

Week 6: Basic RDF/XML syntax

Learning objectives: Reading RDF/XML. Translating between Turtle and RDF/XML. RDF/XML use in library/archive contexts.