

# Digital Methods in Humanistic Research

## Schedule

### Text Encoding Basics: eXtensible Markup Language (XML)

**Tuesday, January 13** Class:

- What is (computational) Digital Humanities?
- Introduction to XML: hierarchy, well-formedness, elements and attributes
- Using oXygen to mark up Lord of the Rings quotes.

Homework:

- Install oXygen on your personal computer (licence key provided on Canvas)
- Install Visual Studios Code
- Read An even gentler introduction to XML
- XML exercise 1: Copy the text of *one* letter by either Oscar Wilde (please close and ignore the pop-up) or Anton Chekhov and mark it up in XML using and employing whatever markup you consider appropriate. In this and all other assignments, follow our file-naming conventions.

**Thursday, January 15** Class:

- Types of XML markup: descriptive, presentational, procedural
- AI and coding
- Project guidelines

Homework:

- Submit questionnaire on research interests
- XML exercise 2: Mark up a text of your choice (any genre, manageable size, foreign languages welcome)
- Read the linked articles in “Learning Git: Part One”.

### Associated Recitation (Tuesday or Thursday)

- Sample course project
- Homework workshop

**Tuesday, January 20** Class:

- Command line: Friend, not foe
- Conceptual introduction to Git
- What is a “schema”?

Homework:

- Complete “Learning Git: Part II,” which teaches you how to set up a GitHub account.
- Read You aren’t a beginning XML developer anymore to help prepare for the XML test.

### Schemas: RelaxNG

**Thursday, January 22** Class:

- Relax NG (Pasta recipe)
- Quiz: XML basics (handwritten preface to the XML test)

Homework:

- XML test (all tests completed over the weekend, due before next class)

### **Associated Recitation (Tuesday or Thursday)**

- Form project teams
- Go over project team norms: weekly meeting times, communication format, etc.

**Tuesday, January 27** Class:

- Relax NG, cont.

Homework:

- Relax NG exercise 1: Write a Relax NG schema for one of the XML documents you created for an earlier assignment (XML exercise 1, 2, or 3; you may modify your XML if you wish) and upload both your XML and your schema file.
- Complete “Learning Git: Part III,” which teaches you how to install Git locally on your personal computer.

**Thursday, January 29** Class:

- Advanced RelaxNG Methods: Dungeon Crawl Exercise (pull or download base files from class Git repo)

Homework:

- Read our Relax NG content models tutorial to learn about modeling mixed content and empty elements.
- Relax NG exercise 2: Choose a small text, perform document analysis, write a schema, mark up the text according to the schema, and upload both your XML and your schema file.

### **Associated Recitation (Tuesday or Thursday)**

- Assembling a corpus and initial document analysis

### **String Matching: Regular Expressions (Regex)**

**Tuesday, February 3** Class:

- Lecture: Regex, the superpowered search /replace language
- Lecture: “language” versus “program / app”
- Walkthrough: Regular expressions (Sonnet 60)

Homework:

- Read the regex mini-tutorial at <http://www.regular-expressions.info/quickstart.html>.
- Regex exercise 1
- Complete “Learning Git: Part IV,” which finishes teaching you the full workflow that you will rely on for your projects

**Thursday, February 5** Class:

- Regular expressions (word list)
- AI-assisted regular expressions
- Quiz: Schema basics (handwritten preface to the RelaxNG test)

Homework:

- Complete RelaxNG test over the weekend
- Read Regex tips (Read the first half, before the Using regex in XSLT section, to acquaint yourself with what’s there, but don’t try to memorize it all at once. You can skip the second half, about regex in XSLT.)

### **Associated Recitation (Tuesday or Thursday)**

- Git: practice and troubleshooting

**Tuesday, February 10** Class:

- Advanced Regex (Scarlet Letter)

Homework:

- Regex exercise 2

### **Web Technologies: HTML, CSS, Javascript**

**Thursday, February 12** Class:

- Lecture: XML versus its subsets (HTML, TEI)

Homework:

- Read Learn CSS layout
- HTML/CSS exercise 1: Create and upload your first HTML and CSS pages Github Pages (see “Getting started with Git Pages”). The content and look of your webpage is up to you, but should demonstrate usage of several XHTML element types as well as CSS styling. Please submit the URL of your page to Canvas.

### **Associated Recitation (Tuesday or Thursday)**

- Thinking about User Experience

**Tuesday, February 17** Class:

- Lecture: Do’s and don’ts of AI assistance for web design
- Walkthrough: CSS Box Model & Flexbox
  - files for in-class exercise: `git clone https://github.com/pickettj/teaching.git`

Homework:

- HTML/CSS exercise 2: Enhance and expand your HTML web page and associated CSS from the last assignment (focus this time on structural elements) and upload them as new files with new names (see the additional instructions about formatting, naming, and uploading your files). *Do not overwrite your files from the first HTML assignment.*
- Read What can XPath do for me?

### **Navigating Textual Markup: XPath**

**Thursday, February 19** Class:

- Lecture: Hierarchical “paths”
- Walkthrough: XPath: overview, paths, axes (fables.xml)

Homework:

- XPath exercise 1

### **Associated Recitation (Tuesday or Thursday)**

- The Text Encoding Initiative

**Tuesday, February 24** Class:

- Lecture: XPath predicates and functions
- Walkthrough: Develop and test (!) path expressions one step at a time

Homework:

- Read The XPath functions we use most
- XPath exercise 2

**Associated Recitation (Tuesday or Thursday)**

- Web design workshop:
  - A complete guide to Flexbox
  - W3Schools CSS Grid layout module

**Thursday, February 26** Class:

- Walkthrough: XPath

Homework:

- XPath exercise 3

**Tuesday, March 3** Class:

- Walkthrough: XPath

Homework:

- Complete XPath test

**Associated Recitation (Tuesday or Thursday)**

- Advanced XPath: work through XPath exercise 4

**Textual Transformation: XSLT**

**Thursday, March 5** Class:

- Lecture: How to look stuff up
- Lecture: XSLT and XPath Overview

Homework:

- Read “XSLT Basics”

**Tuesday, March 17** Class:

- XSLT templates
- XPath expressions and XPath patterns

Homework:

- Read Thinking in algorithms
- XSLT exercise 1

**Thursday, March 19** Class:

- Lecture: XSLT program structure and design

Homework:

- XSLT exercise 2
- Read XSLT, part 2: conditionals and push and pull
- Read Developing an XSLT stylesheet

**Associated Recitation (Tuesday or Thursday)**

- First XSLT homework troubleshooting workshop

**Tuesday, March 24** Class:

- Walkthrough: XSLT homework

Homework:

- XSLT exercise 3
- Read Attribute value templates

**Thursday, March 26** Class:

- Walkthrough: XSLT push and pull

Homework:

- Read Modal XSLT
- XSLT exercise 4

**Associated Recitation (Tuesday or Thursday)**

- Project workshop: from XML to your project website using XSLT

**Tuesday, March 31** Class:

- XSLT: Continued practice

Homework:

- XSLT exercise 5

**Thursday, April 2** Class:

- XSLT: Map Value Pairs
- XSLT: Batch Processing – clone git repo with example files

Homework:

- XSLT Test

**Associated Recitation (Tuesday or Thursday)**

- XSLT for projects

## **Text to Images: Scalable Vector Graphics (SVG)**

**Tuesday, April 7** Class:

- SVG

Homework:

- SVG exercise 1
- Read Viewbox tutorial: an introduction to the SVG coordinate space

**Thursday, April 9** Class:

- David Birnbaum Guest Lecture: XQuery “Drive-by” Tutorial

Homework:

- Read XQuery Drive-by

**Tuesday, April 14** Class:

- SVG: stooges starting point

Homework:

- SVG exercise 2 (Remember that you should upload the XSLT stylesheet that you used to transform your document, not the SVG output of the transformation)

## **Associated Recitation (Tuesday or Thursday)**

- SVG for projects workshop

**Thursday, April 16** Class:

- SVG: bubble chart starting point

Homework:

- SVG exercise 3
- SVG test

## **Associated Recitation (Tuesday or Thursday)**

- Project Workshop

**Tuesday, April 21** Class:

- Project Showcase

**Thursday, April 23** Class:

- Crash course on XML for geospatial mapping

Homework:

- Project sprint