

Informatics 1 Data & Analysis

Tutorial 5

Week 7, Semester 2, 2013

- You must prepare for the tutorial by attempting the questions on this worksheet in advance. Bring with you a copy of your work, including printouts of code and other results.

If you cannot do some questions, write down what it is that you find challenging and use this to ask your tutor in the meeting.

It's important both for your learning and other students in the group that you come to tutorials properly prepared. If you have not attempted the exercise sheet, then you may be sent away from the tutorial to do it elsewhere.

- Some exercise sheets contain material marked with a star ★. These are optional extensions.
- Data & Analysis tutorial exercises are not assessed, but they are a compulsory and important part of the course. If you do not do the exercises then you are unlikely to pass the exam.
- Attendance at tutorials is obligatory: if you are ill or otherwise unable to attend one week then email your tutor, and if possible attend another tutorial group in the same week.

1 Introduction

This tutorial exercise is about XML, using material covered in lectures 9–11. There are several stages, during which you will write some expressions (DTD, XPath) and run some command-line tools (validating a document against a DTD, executing XPath expressions). Please start on this exercise in good time to complete it before the tutorial. If you need assistance, speak to a lab demonstrator or ask the student helpers at InfBASE.

You will need two files from the course web page, *restaurants.xml* and *restaurants.xq*, and you will also need access to a DICE command line. The simplest way to get this is in the computing labs, but it's also possible to connect remotely. For more information about remote working see the relevant pages from Informatics Computing Support.

<http://computing.help.inf.ed.ac.uk/connecting-home-overview>

2 Exercises

Question 1

Draw the tree structure of the XPath data model for the following XML document:

```
<restaurants>
  <restaurant name="La Pasteria" >
    <address>13 Alonisos Street, Patras, 261 35</address>
    <cuisine>Italian</cuisine>
    <phoneno>2610325833</phoneno>
```

```
</restaurant>
  <restaurant name="Kalamarakia" >
    <address>21 Poseidonos Street, Patras, 264 45</address>
    <cuisine>Greek</cuisine>
    <phoneno>2610428066</phoneno>
  </restaurant>
</restaurants>
```

Question 2

Design a DTD for an XML database of restaurants. This should be written to validate the small XML example above, and also the longer document *restaurants.xml* available from the course web page.

Question 3

Validate the given *restaurants.xml* file against your DTD. You can do this using the command `xml-xparse` on a DICE machine. Copy the *restaurants.xml* file and add your DTD inline using a DOCTYPE declaration. Then run the following command in a terminal window on a DICE machine:

```
xml-xparse restaurants.xml
```

This will check that the XML file is well-formed, and validates against the DTD you have given. Try changing some parts of the XML and the DTD to see how `xml-xparse` responds.

Question 4

Write path expressions to extract the following textual information from an XML document called *restaurants.xml* matching your restaurants DTD.

- (a) Retrieve the addresses of all restaurants.
- (b) Retrieve the names of all restaurants.
- (c) Retrieve the address of the *Kalpna* restaurant.
- * (d) Retrieve the phone numbers of all restaurants serving Indian food.
- * (e) Retrieve the names of all restaurants which serve Italian food.

Question 5

In this part you will execute your XPath expressions against the *restaurants.xml* file provided using the *Galax* query tool. Download the file *restaurants.xq* from the course web page and open in a text editor. For each of your XPath expressions, follow the instructions below.

- (a) Add your XPath query in the space allocated in *restaurants.xq*. You will need to make sure your query correctly references the XML file to be checked: the easiest way is to start all paths with `doc("restaurants.xml")/`
- (b) Make sure any other queries are commented out with `(: and :)`
- (c) Save the file.
- (d) In a terminal window, from a directory containing both *restaurants.xq* and *restaurants.xml*, enter the command

```
galax-run restaurants.xq
```

- (e) Add the result as further comments in *restaurants.xq*.

Question 6

How would you represent this data in a relational database? Use the SQL data definition language (DDL) to define an SQL schema for representing the same data.

3 Discussion

- (a) Suggest some strengths and weaknesses of the relational approach compared to XML.
- (b) Given these strengths and weaknesses, name a context in which you would prefer to use XML.
- (c) Given these strengths and weaknesses, name a context in which you would prefer to use a relational database.

★ 4 Extension

Write a Java program to carry out the queries from Question 4. There are two different ways you could approach this.

- Use a Java XML parser to read in the file as a tree — the Document Object Model (DOM) — and then navigate that tree to find the information required.
- Use the Java XPath API to directly execute your answers from Question 4.

Both of these will use the Java API for XML Processing (JAXP) and some combination of DOM, SAX or StAX interfaces. You might find the following article helpful.

<http://www.ibm.com/developerworks/library/x-javxpathapi/index.html>