XPath Exercises

Preparation

1. The files we will need for this are:

- cooper-locations-no-nmspc.kml
- 3733437.xml
- cooper-piracy.xml

2. For each exercise, import the file you want into BaseX.
a. Run BaseX
b. Go to Database > New
c. Browse to the file you want to explore and click OK.

Exercise 1: Navigating the XML Tree with our KML file.

With cooper-locations-no-nmspc.kml loaded in BaseX…

Try out XPath, by entering the following commands into the query bar (with the XQuery option selected):

//Placemark	//Placemark/matches(name, "L")
//Placemark[1]	//Placemark[matches(name, "L")]
//Placemark[2]	*What's the difference between those two?*
//Placemark[Point]	
//Placemark[Polygon]	count(//Placemark/matches(name, "L"))
//Placemark/Point	count(//Placemark[matches(name, "L")])
//Placemark/Point[2]	//Placemark[matches(name, "^L")]
//Placemark[name="London"]	
//Placemark[not(name="London")]	
count(//Placemark[not(name="London")])	

Exercise 2: Querying a Bibliographical Database

With 3733437.xml loaded into BaseX, try to answer the following questions:

How many references are there?

For the next questions, it is important to note that attributes are a little tricky to access in XPath. They can be used as filters, as in //Source[@pubType], but they can't be accessed directly like elements can. So //Source/@pubType won't retrieve anything. But you can get the value of an attribute with the data() function, like this: //Source/data(@pubType) Try it out!

How many different types of publication are referenced?
Hint: This will require a function called distinct-values()

How many of each type of publication are there?
Hint: Without a way to make a loop, you will have to do this one at a time.

Exercise 3: Getting to know some Pirates

With cooper-piracy.xml loaded into BaseX…

Write an XPath expression to retrieve the defendants' names (given and surname) and ages.

Hint 1: Recall that we can retrieve the value of an attribute is with the data() function. e.g. data(@class)
Hint 2: It is possible to use the pipe | to create a union of two different "steps" in a path. For example, //(persName | date) would retrieve all the persName and all the date elements from the xml source.