newtFire {dhlds}
Maintained by: Elisa E. Beshero-Bondar (ebb8 at pitt.edu) Last modified: Sunday, 06-Aug-2017 18:16:43 EDT. Powered by firebellies.

XPath Exercise 3

To begin, download this XML file from our Pacific Voyage project that I have linked here: ForsterGeorgComplete.xml. Open the file in <oXygen/> and work with the XPath 2.0 Window. Respond to the XPath questions below in a text file, and upload to Courseweb for this assignment when you're finished. (Please use an attachment! If you paste your answer into the text box, CourseWeb may munch your brackets.) Some of these tasks are thought-provoking, and even difficult. If you get stuck, do the best you can, and if you can't get a working answer, give the answers you tried and explain where they failed to get the results you wanted. Sometimes doing that will help you figure out what's wrong, and even when it doesn't, it will help us identify the difficult moments.

These tasks require the use of path expressions, predicates, and the functions count(), not(), name(), position(), last(), and distinct-values(), but they should not require any other XPath functions. There may be more than one possible answer. To read about these functions, you should consult *The XPath Functions We Use Most* page and if you have the Michael Kay text, it may be useful to you here. As always, consult our class notes and our introductory guide *Follow the XPath!*.

With the Georg Forster file open in oXygen and using the XPath 2.0 browser window in oXygen, construct XPath expressions that will do the following. **Be sure to give the FULL XPath expression you used in your answer, and don't just report your results.** This way, if the answer is incorrect, we can help explain what went wrong.
** Notation: For ease in recognition, from now on when we refer in discussion to an attribute name, we'll precede it with an at sign (@). In other words, when we write about the @type attribute below, the name of the attribute is actually type (without an at sign).

1. Working with @type:
    1. Write an XPath using a function that returns a count of number of times we've used @type attributes in the Georg Forster file.
    2. What's the XPath to return the parent elements (whatever they are) of @type?
    3. Modify the XPath in your previous statement to return in the bottom results window the **names** of those parent elements
    4. Modify the XPath expression once more to return a list of only the distinct-values of those parent elements.

2. Working with attributes of ANY kind:
    1. Write an XPath expression to return all the attributes of any kind, anywhere in this file.
    2. Using the name() function, build on your previous XPath expression to return the names of these attributes.
    3. Now, return only the distinct-values of those attribute names: What XPath expression does this?
    4. Now, what if we wanted to find all the **parent elements** (without knowing what they are) of any attributes in use within the **body** element of the file? Write the XPath expression.
    5. What's the XPath to return the distinct-values of the **names** of those parent elements in the body of the file.
    6. How many distinctly different element names are holding attributes of any kind? (What expression returns this as a count?)
3. Working with the count() and position() functions in predicates: (count(), position(), last()): (You'll need to look up how to look for a count() of something and set it equal to, greater than, or less than a particular number.)
    1. Write an XPath expression that returns the **last** paragraph in the ENTIRE Georg Forster file that contains **more than one** latitude record, coded as <geo select="lat">. (Hint: This builds on things we showed you in the XPath Exercise 2 homework. Note that you should only get ONE result here!)
    2. Modify the expression so you return the **first** paragraph in the ENTIRE Georg Forster file that contains **more than two** latitude records. (There's no such thing as a first function, but remember how we found the first, second, and third books and chapters in past XPath exercises? The same working with position numbers applies here.) Again, you should only get one result in your results window.
    3. Now, how would we write XPath to find **the very last paragraph in Book 2** that contains **more than 1 latitude record**? As before, you should only get one result for this.
    4. Now, can you write an XPath expression that finds **the very first** paragraph holding **more than two latitude** records, that also holds **more than one placeName element**?
    5. How would you modify the previous expression to return the contents of the placeName elements in that paragraph? What are the placeNames?
4. **Optional Bonus Challenge Question:** Try out an XPath function on this file that we haven't yet assigned. Look up functions in the links posted here in our instructions, or in the Michael Kay book if you have it. Explain what you tried, give your XPath expression, and describe its results in your return window.