


<oo> → <dh> Digital humanities

Maintained by: David J. Birnbaum (djbitt@gmail.com) 

Last modified: 2015-10-11T18:32:10+0000

XPath assignment #3

You can find an XML (TEI) version of Shakespeare's *Hamlet* at <http://dh.obdurodon.org/bad-hamlet.xml>. We've deliberately damaged some of the markup in this edition to introduce some inconsistencies, but the file is well-formed XML, which means that you can use XPath to explore it. You should download this file to your computer (typically that means right-clicking on the link and selecting "save as") and open it in <oXygen/>.

After you've completed your homework, save your answers to a file and upload it to CourseWeb as an attachment. (Please use an attachment! If you paste your answer into the text box, CourseWeb may munch the angle 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 functions. There may be more than one possible answer.

Using the *Bad Hamlet* document and the XPath browser window in <oXygen/>, construct XPath expressions that will do the following (give the full XPath expressions in your answers, and not just the results):

1. What XPath expressions will find the last stage direction <stage> in the entire document? (Note: there should be only one!)
2. What XPath expression will find the last member in the cast list at the beginning of the document and return the value of the @xml:id attribute that is associated with it?
3. What XPath expression will find all <sp> elements with more than 8 line (<l>) subelements? You'll need to use the count() function (Kay 733–34).
4. Building on your answer to the preceding question, what XPath expression will tell you how many line subelements each of those speeches actually has?
5. Building on your answers to the preceding two questions, what XPath expression will find the speakers of all speeches that have more than 8 line subelements? Once you've found the speeches that have more than 8 lines, you can find the speakers of those speeches by just adding another path step, but you'll get some duplication, since a single person may have more than one long speech. Your answer to this question should get rid of the duplicates, and return just a list of names of speakers without duplication. You'll need to use the distinct-values() function (Kay 749–50).

Optional bonus questions

1. Question #1, above, asked how you to provide an XPath that would find the last stage direction (<stage>) in the play. What XPath would find the last line (<l>) in the play? What XPath would find the last stage direction or line (that is, whichever of the last stage direction and last line comes last)? You'll need to use the union operator (Kay 628–31).

2. Question #2, above, asked you to provide an XPath that would find the `@xml:id` associated with the last cast member in the cast list. What's the difference between an XPath that returns the `@xml:id` attribute itself and an XPath that returns just the *value* of the `@xml:id` attribute? That is, what are the two XPath expressions and what object does each of them return? You'll need to use the `data()` or `string()` function (Kay 741–43, 877–79).
3. Question #3, above, asked you to provide an XPath that would find all of the speeches (`<sp>` elements) with more than 8 line (`<l>`) subelements. What are the XPaths to find speeches with more than 8 line *child* elements and speeches with more than 8 *descendant* line elements? How do those results differ? If there are descendant line elements that are not children of a speech, what is their parent? If you don't know the types of their parent elements in advance, what XPath expression will tell you?