newtFire {dh}
 Last modified: Monday, 14-Feb-2022 17:51:34 UTC.
Maintained by: Elisa E. Beshero-Bondar (eeb4 at psu.edu). Powered by firebellies.

Regex Exercise (Test): Autotag the Radio Script of *The Maltese Falcon*

- Our newtFire tutorial on Autotagging with Regular Expressions (Regex)
- Regular-Expressions.info Tutorial: a mine of helpful detail on regular expression matching,

The test

- For this test you need to download the Maltese Falcon file from this page (right-click on the file and save it to your computer).
- After you have the file downloaded and opened the file in oXygen, open the Find/Replace window.
- **Read this exam carefully: This is different from the homework assignments because sometimes we are giving you expressions to test and explain what they are matching.**
- Open a new markdown or text file to record your responses to each of the questions. **Record a response to each of the following tasks carefully**, since this is the file we will be evaluating for exam credit. These will include global Find-and-Replace operations working with oXygen's Regular Expression search features.. Your goals are to answer each question clearly and precisely, and to produce a well-formed XML document at the end of the exercise. However, even if you have have trouble generating a well-formed XML document, what's most important is that you document the steps you took for partial exam credit.
- We have already verified for you that there are no reserved characters (`&`, `<`, and `<`).
- Also there are no groups of blank lines exceeding 2 (`\n{2,}`).
- **Before you begin**, make sure you have set up your Find and Replace for searching with regular expressions, that case sensitive searching and Wrap around are checked on, and that you are able to see space characters in your oXygen window. You will be adjusting below whether or not to select "Dot matches all" when you use the dot `.` regex.

Your Tasks:

1. **Begin by testing the following regular expression Find and Replace patterns**.

   First, please paste in the following regex pattern in the **Find** window: `^(.+?)(\n\n)`, and in the **Replace** window, paste in this expression: `<sp>\1</sp>\2`. Then, answer the following questions in your markdown or text file:

   1. Experiment with the **"Dot matches all"** setting: Inspect the results when Dot matches all is not checked, and then when it is checked. Then explain here, should you turn on "Dot matches all", or not? Explain why or why not. (4 points)
   2. Explain exactly what `\1` and `\2` are referring to in the replace window. What are these referring to in the Find expression, and why do we need to use these expressions? (4 points)
   3. When you have decided the best course of action, complete the Find and Replace operation to tag the text blocks with `<sp>.....</sp>`.
2. What Find and Replace operation can you use to tag the stage directions (or cues for the actors) in this document? Record your Find and Replace operations, and if you use the dot in your expression, indicate whether you use Dot matches all or not (or whether it makes any difference). (5 points)
3. What Find and Replace operation can you use to tag the speakers? Record your Find and Replace operations, and if you use the dot in your expression, indicate whether you use Dot matches all or not (or whether it makes any difference). (5 points)
4. Add a root element to your document, save it as XML, and note anything you need to do (if anything) to make this a well-formed XML document (showing the green square). (2 points)
5. **Optional: Extra credit** (2 points): Notice that there are some stage directions that are not really meant to be inside character's speeches. Can you use regular expressions to search and replace stage directions that are marked inside `<sp>` elements but are not inside real speeches —just block stage directions meant to stand by themselves? Record your expressions to remove the unnecessary `<sp>` elements from the code as a last step.
6. Submit two files on Canvas to complete this test:
   - Your markdown or text file in which you have recorded your responses these exam questions (saved with `.md` or `.txt` file extension as in `stepFile.md`).
   - Your resulting file, saved with `.xml` file extension as in `MalteseFalcon.xml`.