

Hints and additional information for Programming Assignment 1: Flesch Index with BasicDocument.java

Part 1: BasicDocument.java

Hints:

- Notice the helper method `getTokens` in the Document superclass. It takes a string which is a regular expression representing the regular expression of the “tokens” it’s trying to find. Consider passing a different pattern for each method. If you get really stuck here there is a support video that can help.
- It is possible to write regex's to count the number of syllables directly (we did it with a combination of 3, but it might also be possible to write a single one), but this is not necessarily the best approach. If you find yourself doing mental gymnastics to come up with a single regex to count syllables directly, that's usually an indication that there's a simpler solution (hint: consider a loop over characters--see the next hint below). Just because a piece of code (e.g. a regex) is shorter does not mean it is always better.
- Under the rules for what defines a syllable, the words “the”, “fly”, “yes”, “cave” and “double” all have 1 syllable, but “segue” has two syllables. Notice that this isn’t exactly correct (“double” actually has 2 syllables), but it’s close enough for our purposes. Here are some more examples with the number of syllables your method should return to help you: “contiguous” (3 syllables), “sleepy” (2 syllables), “obvious” (2 syllables), “toga” (2 syllables). Notice that our rules get a lot wrong, especially when you have more than 2 vowels in a row, but these are the rules we will test you against.
- Consider making a helper method in the Document class (i.e. the superclass) that counts the number of syllables in a single word. This will come in useful in next week's assignment as well. We have a stub method already in Document.java. You should not need to create `countWords` or `countSentences` methods in Document.java because these methods will not be useful next week. For reasons of efficiency, which we'll mention more in next week's assignment, you should not create `Matcher` or `Pattern` objects inside your `countSyllables` method. Just use a loop to loop through the characters in the string and write your own logic for counting syllables.

Explanation for the first provided test case (if you have questions about others, we encourage you to ask in the forums):

Test case string: “many??? Senteeeeeeeeeeencences are”

Number of sentences: 2 There is one sequence of characters ending with a series of end of line punctuation marks, as well as the last sequence of characters which also counts as a sentence (despite not ending with an end of line punctuation mark.)

Number of words: 3 There are three sequences of contiguous alphabetical characters (many, Senteeeeeeeeeeencences, are).

Number of syllables: 6 `getNumSyllables()` should evaluate to 6 syllables because there are six contiguous sequences of vowels which fit our definition of a syllable. They are:

- many: 2 (a, y)
- Senteeeeeeeeeeencences: 3 (e, eeeeeeeeeee, e)
- are: 1 (a) -- note the lone e on the end does not count in this case.