Download [Macbeth as XML](https://www.ibiblio.org/xml/examples/shakespeare/macbeth.xml) and compare/contrast with both the "flat file" data in the Terrorism data file and the just-the-text version file from Project Gutenberg.

In no more than 250 words, reply to the following:

How is the tree structure of the xml data file different from the flat structure of the csv data file? How are both different from the just-the-text version file? And how might a program impose structure on the just-the-text version?

Type your response in the text field and submit it to the dropbox.

The Macbeth XML file is organized in a tree (or hierarchical) structure, as opposed to the flat (CSV) Terrorism file or the text-only version of Macbeth. The tree structure does not lend itself to a relational database, whereas the flat file can quite easily be organized into tabular format. There may be cases where a tree structure makes sense and is advantageous over a flat file, for instance an HTML (or XML) document. In these cases, it is difficult to conveniently split up the document into separate but consistently formatted data points or rows.

On the other hand, the CSV file can conveniently fit into a tabular format, which allows for easy querying of the document. They make logical sense as each row is a single data point, so can easily be interpreted by humans. However, if there is a redundancy in a column of the data (for example, if each row represents a person, but two people have the name “John”), the data can easily be mis-queried.

Finally, the text-only version lacks any structure that is recognizable by a computer. This makes it difficult to organize into a coherent format, but also means the file itself takes up minimal memory. One could programmatically impose structure on the text with a few basic commands, such as searching for tabs at the start of a paragraph, or quotation marks to indicate speech. In the case of the Macbeth file, you could also use regex to locate when a new character speaks.