Functional Programming Coursework - Report

500 words max

For this report, I am comparing my experiences with Haskell and

Python. In particular, I have used the Scalpel scraping library in

Haskell, and the equivalent scrapy library in Python.

Code size:

The code I have provided is relatively concise. I have designed the filtering functions in a way that is both short and concise, although I think some could have been further optimised if I had a better understanding of certain syntactic elements within Haskell. The lengthiest part of my implementation is the body of the mostfrequentwordonpage function, which essentially consists of a succession of let-expressions. I think I was influenced by the syntax of other languages when I tried to implement the main function; Although I managed to produce a working solution, I am certain I could have implemented a more functional approach for this part of the program. Furthermore, I have used many external libraries in order to clarify and shorten the code through the use of functions such as filter or map. Overall, I am satisfied with the final state of my solution, given that Haskell was my first introduction to the functional programming style.

Code readability:

I have tried to make my code as readable as possible. The variables have meaningful and consistent names. I also ensured that the functions are clearly designed and have coherent signatures. I have written comments to help potential readers follow the sequence of operations executed by the program. Some time was required to understand some of the basic types and concepts of functional programming, but I now feel that my code was easier to read than some other styles such as object-oriented programming. This is partially due to the clear function signatures as well as the absence of tightly coupled hierarchy of classes.

Code maintainability:

I have written comments to make the code more understandable and easier to maintain. I have also tried to avoid complicated nested types in order to facilitate potential debugging of the code. The most common error I encountered was “Could not match types”, which was quite frustrating at first. I managed to slowly work my way through these errors as my understanding of Haskell types increased, but it was crucial for me that the types remained simple and understandable (by avoiding long sequences of nested types). The functional programming style is also generally easier to test and debug due to the restrictions and standards imposed on pure functions (no hidden or inconsistent states).

Toolchain support:

With the skeleton of the project provided, I had no problem with basic Stack commands like run, build or test. However, I was much less productive coding in Haskell than I would have been for the same task in Python. This is due to my lack of experience as well as the smaller amount of available online resources: there are a few times where I felt stuck as I could not find anyone experiencing the same problem online. I found myself returning to the same pages and resources much more often than I would normally to make sure I had not missed anything.