Problem #3

Question:

If you could change 1 thing about your favorite framework/language/platform (pick one), what would it be?

Response:

This question was actually very interesting to me. I often find that I only wish I could change an aspect of programming when I am not currently aware of why that aspect exists in its current state. For instance, a very basic example is that, to a new programmer, the use of the semicolon to end a line in Java may seem useless when the programmer can simply press enter to go to the next line. However, after learning that the semicolon is used to end lines in Java to signal to the complier where one instruction ends and the next instruction begins, the programmer can better appreciate the use of this aspect that may have caused a roadblock in the past. One can see how this logic can apply to other examples such as allocating memory in the C programming language or array indexing starting at 1 rather than the typical 0 in the MATLAB programming language. This logic again follows to show that many seemingly convenient changes, such as being able to initialize an array of unknown size in Java, could come with unwarranted consequences. Therefore, I will attempt to answer this question in such a way that the response does not change the foundation of an existing programming language.

Changes that can be made in this fashion are minimal and would mainly suit personal convenience rather than improved function. As a result, my response is catered towards my own personal preference. If could change one thing about the Python programming language, I would have the main method/function, if it exists in the script, be run automatically as it is in Java or C. It would not be required for the main method to exist as not every Python script needs one; however, in my experience, should the main method exist, having the main driver method be run automatically would greatly facilitate my coding. Creating the main method and then using

```
if __name__ == '__main__':
    main()
```

to call the method feels very redundant. This could possibly have consequences that I have not foreseen; however, I believe that this is the safest, yet most personally practical, change I could make to the Python programming language.