

This is why I commit to Python and rely on it to help me succeed.

We chose Python in part because we have both worked in Python before and have even pair programmed in it together. In addition, Python is a very popular language, with a lot of documentation available online.

Python supports running programs both on the school's lab machines (which run UNIX) and on our laptops from the command line with a single command from within the directory which houses the program file without any code changes. In a UNIX command line, the command that would run the program is "python program\_file\_name", while in a Windows command line, the command requires the file extension and is thus "python program\_file\_name.py". In addition, Runp is a downloadable Python tool that allows one to run/test specific functions from the file within the command line.

Python supports entering values in the console through the function "input()". You can also print to the console using "print()". Python has the ability to read and write to files as well. In addition, you can import "socket" and use the library to work with TCP/IP sockets.

Python supports modular programming in that each Python file acts as a module which can have little or no dependency on other Python files. Variables and functions defined within the scope of a Python file cannot be accessed outside of that file unless through controlled imports into other files. Additionally, modularity is supported within Python files with the use of functions and classes.

Python supports reading and writing JSON with the "json" library included in the Python Standard Library. Functions from the library can be accessed within a Python file by including the line "import json" at the top of the file.

Python supports loading code dynamically by using the method "\_\_import\_\_()". In order to dynamically load a class from a module, the following lines of code can be implemented:

```
dyn_module = __import__(module)
dyn_class = getattr(dyn_module, class)
```

Python supports automatic unit testing through the "unittest library". Additionally, existing Python tools like "Coverage.py" can be used to determine test coverage.

Python has an IDE named PyCharm which provides the ability to interact directly with the Python interpreter via a console which is useful for exploratory programming. We will be using this IDE to run in Python.