

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

Below are the requirements given for the project and our rationale for choosing Python to satisfy them:

*(i) support for running programs on the school's lab machines (UNIX) and your laptops from the command line with a single command without code changes;*

Python programs can be run on unix using the command `python <program>.py`, where `<program>` is the name of the file we wish to run.

*(ii) support for UNIX-style STDIN, standard I/O and TCP/IP sockets;*

Python has a `stdin` in the `sys` package which can read STDIN when we use `sys.stdin`. It also has STDOUT capabilities with `sys.stdout` and the `print()` function to print to STDOUT. Within the `socket` package python allows creating a TCP/IP socket using `socket.socket(socket.AF_INET, socket.SOCK_STREAM)`.

*(iii) modular programming (think modules, functors, packages etc.);*

Along with Python's standard library, there are many packages and modules we can import to allow us to execute different tasks. Also we can import modules we write into our main files.

*(iv) reading and writing JSON;*

With the `json` package, python can easily write to json files using the `json.dump()` function and likewise read from a json file using the `json.load()` function.

*(v) loading code dynamically;*

By using `__import__(<module>)`, where `<module>` is a string with the name of the module we want to load, we can load modules dynamically.

*(vi) automatic unit testing and test coverage*

The `unittest` module in python allows for automated unit testing when the file is run by including `unittest.main()` at the end of a python script.

*(vii) an IDE with support for exploratory programming*

PyCharm offers an IDE that supports python and allows for exploratory programming. Also python can be run on a terminal interactively and testing small pieces of code is possible.