

**To:** Matthias Felleisen  
**From:** Gino Jacob and Emily Miller-McGlone  
**Subject:** review of not so hot tools for our super hot project

We wrote Santorini in Python 3.6 and the VSCode editor with the assumption that the language's long-running community would ease our development in terms of searching the right libraries and debugging code.

The language's untyped nature is a double-edged sword. On one hand, there's the flexibility in defining data structures, and on the other, enforcing them is a challenge. It was crucial that we accurately represent the data signature of every function and method we wrote with adequate documentation. We stuck to the informally written contracts but found that the data pipeline was still vulnerable to our own errors.

Python's API reflects the same issue. Documentation for the built-in TCP socket server, I/O, and multi-threading libraries lacks formal data definitions, but luckily VSCode has strong Python extensions that ease API references. Regardless, we had to consult with StackOverflow multiple times to understand libraries and debug code.

We used third party libraries for JSON parsing and the timeout functionality. In terms of JSON parsing, it felt that there was always an insurmountable caveat. Our first library could parse multiple values perfectly but somehow ignored JSON strings completely. We had to resort to the built-in library and hack around its inability to parse multiple values.

Python is a popular backend language, and as such, its support for establishing TCP servers and clients were much more intuitive than we thought. There was no need to find a higher level abstraction of the built-in socket library since all that was needed was a simple JSON data exchange between servers and clients.

As for our programming environment, exploratory exercises were quick and easy with Python and VSCode, mainly because the language has a built-in REPL and the editor is lightweight with an integrated terminal window. Running the program itself and tests was fast because VSCode is not a memory and CPU hog.