Pete Wurster; J00204848 04 Sept 2019
Fall 2019; CIS270; Lab01

As an interpreted (rather than compiled) language, Python’s typing is neither strict nor static. Since Python’s typing is dynamic, it has no problem re-assigning a variable’s type mid-program:

```
someVar = 15                  //set the var to Int
print(type(someVar))          //prints ‘15’
someVar = “fifteen”           //set the var to Str
print(type(someVar))          //prints ‘fifteen’
```

Professor Liss states that static and strict typing schemes are wholly separate concepts, but I disagree. To put it in terms as I see them, I would assert that, with regards to Swift, “Static extends Strict.” In both schemas, variables are treated equally strictly once the type is determined. The major difference between the two is ‘when’ a variable’s type is set. This ‘when’ is essentially a function of how advanced the compiler is: the compiler-program is either designed smart enough to infer type at the time of compiling OR typing needs to be done explicitly by the programmer ahead of compiling.

Static and strict typing could be advantageous because I assume that they allow for the most efficient use of memory to retain the variable for use. For example: if you know that you only need 8-bits to store small Int variables, that means you could store 4 such variables in the same space it would take up to hold a single float value. Using up less memory in this way is bound to result in better performance on memory-hungry systems. Another advantage of strict typing is that your code (once successfully complied) is less likely to encounter illegal operations during execution.
The main disadvantage I see with strict-typing is that it requires more careful consideration be given to which variables will be used for what operations later in the program.

Since almost every major operating system is written in a variant of the C programming language, I would say that strict typing is the preferred standard. However, loosely-typed languages like Javascript and PHP allow web-software developers to do amazing work on-the-fly. Dynamically-typed languages like Python also allow flexibility that opens the door for less-experienced programmers (like me) to jump right into action.

The major differences in my two programs was that the Python code worked right the first time, and my Swift code did not! The biggest challenge I faced was figuring out how to run ANY Swift code in the first place. Both the xCode and Playground IDEs are wholly unfamiliar, and I struggled with null values while retrieving user input in order to test my code.

Works Cited

No other works to cite. All opinions expressed are formed by myself, based on knowledge accumulated from the 11 CIS/MATH courses I have completed in the past 2 years.