It can be made even more difficult with an unfortunate combination of assumption and working problems out on your own. Without a mentor especially, it can be rather difficult to ever even know if one way you are doing something is wrong. We are certainly all guilty of going into our code at a later date and refactoring because we are all learning constantly how to do things in a better way. Fortunately, with the right amount of awareness, correcting these mistakes can make you a significantly better programmer.

The greatest way to become a greater programmer is to overcome mistakes and problems. There is always a better way of doing something, it's finding that specific better way that is challenging. It's easy to get used to doing one thing or another, but sometimes a bit of a shake-up is needed to really get the ball rolling on becoming a great engineer.

Default arguments in Python are evaluated *once,* and the evaluation takes place when the function definition is executed. Given that these arguments are evaluated once, each element inbound is used in each call, which means that the data contained in the variable is mutable across each time it is accessed within the function.

Though this mostly applies to the statistical/DS/ML side of Python users, having immutable data is universally important depending on the circumstances.