Python has an interesting data structure called a "list" that is much more than a mere list. In fact, a Python list actually encompasses the functionality of almost every list-based data structure in this lesson.

Behind the scenes a Python list is built as an array. Even though you can do many operations on a Python list with just one line of code, there's a lot of code built in to the Python language running to make that operation possible.

For example, inserting into a list is easy (happens in constant time). However, inserting into an array is O(n), since you may need to shift elements to make space for the one you're inserting, or even copy everything to a new array if you run out of space. Thus, inserting into a Python list is actually O(n), while operations that search for an element at a particular spot are O(1). You can see the runtime of other list operations [here](https://wiki.python.org/moin/TimeComplexity).

Python is a "higher level" programming language, so you can accomplish a task with little code. However, there's a lot of code built into the infrastructure in this way that causes your code to actually run much more slowly than you'd think. Keep this in the back of your mind when using Python. You likely won't need to know the details of how Python works behind the scenes in a programming interview, but you'll seem very impressive if you do!

If you aren't already comfortable with Python lists, you can look through this [lesson](https://developers.google.com/edu/python/lists) about basic Python list manipulation.