**Week 6 Discussion**

Please read: [15 things you should know about Dictionaries in Python | by Amanda Iglesias Moreno | Towards Data Science](https://towardsdatascience.com/15-things-you-should-know-about-dictionaries-in-python-44c55e75405c). List three things that you learned in addition to what we learned in class and how they could be used in practice. Why do you think dictionaries are such an important data structure? Please respond to 3 other students' posts.

In addition to what we learned in class, I learned a few important concepts from the article. The first being that by default dictionaries are unordered (unless using a Python version greater than 3.6). However, OrderedDict remembers the order in which its contents are added. This can be incredibly useful in ensuring the order of keys and elements within the dictionary are maintained. For example, if the order of data collected was important, say in a time-series based dataset collecting heart-rate data over a given time for individual patients, this would be incredibly useful. If the data was out of order because the processing of Python occurred across multiple machines that were not all utilizing a version greater than Python 3.6, this could scramble the dataset and render it useless, or at the very least less useful.

Another concept that seems useful is the use of nested dictionaries. The ability to create values in a dictionary which are actually other dictionaries provides a great way to group data. A common use-case could be if there are many dictionaries that need to be created within a large and complex data set. The highest level of dictionary could contain an overarching theme for the data, and the additional nested datasets could be additional values underneath the original dictionary. For example, if our dataset described various modes of transportation, the highest level of dictionary could generically describe boat, car, motorcycle, etc. The dictionaries underneath could describe the make, model, horsepower, and more.

A third takeaway from the article is the ability to use dictionary comprehension to iterate over keys and values. Traditionally, in other programming languages, I have used hash maps to list both the key and value of the result in a for loop. However, dictionary comprehension provides a much-simplified syntax for achieving the same result. For example, if I wanted to pull employee IDs (key) and their associated performance review ratings (value), I could do so with one line of code as opposed to creating an empty dictionary, hash-table, etc. and then iterating over each value and adding the paired key:value to the empty dictionary. This provides a simple way to write for loops, and in my opinion, is much easier to learn and read as opposed to learning the syntax and logic in a traditional for loop.

Overall, dictionaries are incredibly useful tools in Python. They provide flexibility to create relationships between keys and values in a specified container, and can be used in data analysis, transformation, and more.