**Summary: *Learning Python*, pp. 239-254**

Pages 239 to 253 in the textbook *Learning Python* by Mark Lutz give an overview of the Python list and dictionary objects and describe common functions that can be used with each of these objects. As mutable objects, lists and dictionaries may be changed after they’re created. This makes them more versatile than immutable objects such as strings, booleans, and integers. Lists can be indexed, sliced, and concatenated, much like strings, but can also support index assignment operations (e.g. slice assignment) as well as type-specific method calls (e.g. sort and append). Dictionaries, like lists are also mutable objects which have their own type-specific method calls (e.g. update and get). However, although lists and dictionaries are both mutable objects, there are some key differences between them. For instance, lists are ordered, whereas dictionaries are unordered, which serves to make dictionaries more flexible than lists. Rather than accessing values by their indices, as is done in the case of lists, dictionary values are accessed by their corresponding keys. For this reason, lists and dictionaries have different applications, and dictionaries are the best storage option when large amounts of data must be aggregated.