10: TUPLES

10.1: TUPLES ARE IMMUTABLE

A tuple1 is a sequence of values much like a list. The values stored in a tuple can be any type, and they are indexed by integers. The important difference is that tuples are immutable. Tuples are also comparable and hashable so we can sort lists of them and use tuples as key values in Python dictionaries.

10.2: COMPARING TUPLES

The comparison operators work with tuples and other sequences. Python starts by comparing the first element from each sequence. If they are equal, it goes on to the next element, and so on, until it finds elements that differ. Subsequent elements are not considered (even if they are really big).

10.3: TUPLE ASSIGNMENT

One of the unique syntactic features of the Python language is the ability to have a tuple on the left side of an assignment statement. This allows you to assign more than one variable at a time when the left side is a sequence.

10.4: DICTIONARIES AND TUPLES

Dictionaries have a method called items that returns a list of tuples, where each tuple is a key-value pair.

10.5: MULTIPLE ASSIGNMENT WITH DICTIONARIES

10.6: THE MOST COMMON WORDS

10.7: USING TUPLES AS KEYS IN DICTIONARIES

Because tuples are hashable and lists are not, if we want to create a composite key to use in a dictionary we must use a tuple as the key.

10.8: SEQUENCES: STRINGS, LISTS, AND TUPLES - OH MY!

10.9: DEBUGGING

in this chapter we are starting to see compound data structures, like lists of tuples, and dictionaries that contain tuples as keys and lists as values. Compound data structures are useful, but they are prone to what I call shape errors; that is, errors caused when a data structure has the wrong type, size, or composition, or perhaps you write some code and forget the shape of your data and introduce an error.

10.E: TUPLES (EXERCISES)

10.G: TUPLES (GLOSSARY)