10.1   Maps and Dictionaries

Python's **dict** class is arguably the most significant data structure in the language. It represents an abstraction known as a ***dictionary*** in which unique ***keys*** are mapped to associated ***values***. Because of the relationship they express between keys and values, dictionaries are commonly known as ***associative arrays*** or ***maps***. In this book, we use the term *dictionary* when specifically discussing Python's dict class, and the term *map* when discussing the more general notion of the abstract data type.

As a simple example, Figure 10.1 illustrates a map from the names of countries to their associated units of currency.
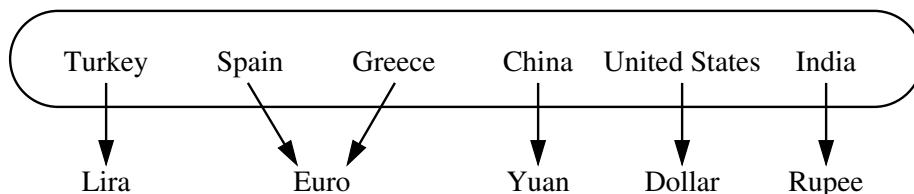

Figure 10.1: A map from countries (the keys) to their units of currency (the values).

We note that the keys (the country names) are assumed to be unique, but the values (the currency units) are not necessarily unique. For example, we note that Spain and Greece both use the euro for currency. Maps use an array-like syntax for indexing, such as currency['Greece'] to access a value associated with a given key or currency['Greece'] = 'Drachma' to remap it to a new value. Unlike a standard array, indices for a map need not be consecutive nor even numeric. Common applications of maps include the following.

- A university's information system relies on some form of a student ID as a key that is mapped to that student's associated record (such as the student's name, address, and course grades) serving as the value.
- The domain-name system (DNS) maps a host name, such as www.wiley.com, to an Internet-Protocol (IP) address, such as 208.215.179.146.
- A social media site typically relies on a (nonnumeric) username as a key that can be efficiently mapped to a particular user's associated information.
- A computer graphics system may map a color name, such as 'turquoise', to the triple of numbers that describes the color's RGB (red-green-blue) representation, such as (64,224,208).
- Python uses a dictionary to represent each namespace, mapping an identifying string, such as 'pi', to an associated object, such as 3.14159.

In this chapter and the next we demonstrate that a map may be implemented so that a search for a key, and its associated value, can be performed very efficiently, thereby supporting fast lookup in such applications.