

## 8: LISTS

### 8.1: A LIST IS A SEQUENCE

Like a string, a list is a sequence of values. In a string, the values are characters; in a list, they can be any type. The values in list are called elements or sometimes items.

### 8.2: LISTS ARE MUTABLE

The syntax for accessing the elements of a list is the same as for accessing the characters of a string: the bracket operator. The expression inside the brackets specifies the index. Remember that the indices start at 0:

### 8.3: TRAVERSING A LIST

The most common way to traverse the elements of a list is with a for loop.

### 8.4: LIST OPERATIONS

The + operator concatenates lists.

### 8.5: LIST SLICES

The slice operator also works on lists.

### 8.6: LIST METHODS

Python provides methods that operate on lists. For example, append adds a new element to the end of a list:

### 8.7: DELETING ELEMENTS

There are several ways to delete elements from a list. If you know the index of the element you want, you can use pop.

### 8.8: LISTS AND FUNCTIONS

There are a number of built-in functions that can be used on lists that allow you to quickly look through a list without writing your own loops:

### 8.9: LISTS AND STRINGS

A string is a sequence of characters and a list is a sequence of values, but a list of characters is not the same as a string. To convert from a string to a list of characters, you can use list.

### 8.10: PARSING LINES

Usually when we are reading a file we want to do something to the lines other than just printing the whole line. Often we want to find the "interesting lines" and then parse the line to find some interesting part of the line.

### 8.11: OBJECTS AND VALUES

To check whether two variables refer to the same object, you can use the is operator.

### 8.12: ALIASING

The association of a variable with an object is called a reference. In this example, there are two references to the same object.

### 8.13: LIST ARGUMENTS

When you pass a list to a function, the function gets a reference to the list. If the function modifies a list parameter, the caller sees the change. For example, delete\_head removes the first element from a list.

### 8.14: DEBUGGING

### 8.E: LISTS (EXERCISES)

### 8.G: LISTS (GLOSSARY)