**Chapter 7: Lists and Tuples**

Sequences

* **Sequence**: An object that holds multiple items of data, stored one after the other
  + Can perform operations on a sequence to examine and manipulate the items stored in it
* **List**: is mutable (i.e. a program can change its contents
* **Tuple**: is immutable (i.e. once it is created, its contents cannot be changed

Introduction to Lists

* **List**: An object that contains multiple data items
  + Is mutable, meaning their contents can be changed during a program’s execution
  + **Dynamic data structure**: items can be added or removed and can use indexing, slicing, and other methods
* **Element**: Each item that is stored in a list
* Statement that creates a list of integers:



* List of five strings:



* List can hold different item types



* Can use *print* function to display an entire list

A close up of a number

Description automatically generated🡪

* Built-in *list()* function: Converts certain types of objects to lists
  + Example: *range()* returns an iterable, an object holding a series of values iterated over. Can use a statement to convert the range’s function’s iterable object to a list:



The Repetition Operator

* A black text with black text

  Description automatically generatedWhen multiplying a list on left-hand side of \* symbol and right side is an integer, called *repetition operator*
  + Makes multiple copies of a list and joins them all together

Iterating over a List with the *for* Loop

* A diagram of numbers and a number

  Description automatically generated with medium confidenceA close-up of black text

  Description automatically generatedGeneral format:

Indexing

* **Index**: Another way you can access individual elements in a list
  + Starts at 0
  + 1 less than the number of elements in the list
  + Can use a negative (-) symbol to go to start at the last element in the list

The *len* Function

* *len()* returns length of a sequence
* A number and a number

  Description automatically generated with medium confidenceReturns the value 4 🡪
* Can be used to prevent an IndexError exception

A black text with black text

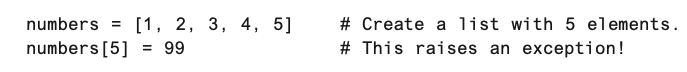
Description automatically generated with medium confidence

Using a *for* Loop to Iterate by Index Over a List

* Can use *len()* with *range()* to get indexes for a list
* A close up of a text

  Description automatically generatedExample:

Lists Are Mutable

* A black text with black text

  Description automatically generated with medium confidenceMeans their elements can be changed
  + Above causes error, numbers list creates 5 elements, indexes 0 through 4. No element at index 5
* Instead, create the list first:

A black text on a white background

Description automatically generated

* + Line 2 creates a list with 5 elements, each element assigned value 0
  + Loop in lines 5 and 6 steps through the list elements, assigning 99 to each one