CHAPTER 8 – STRINGS AND LISTS

8.1

A sequence is an object that holds multiple items of data, stored one after the other, its main types in python are: strings, lists, tuples.

Sequences can be mutable (lists) or immutable (strings, tuples).

8.2

A string is a sequence of characters, it can contain alphanumeric characters and symbols.

Strings must be enclosed in quotes ‘ ‘ or double quotes “ “.

8.3

The main operators for strings are:

+ 🡪 concatenates two strings of the same data type (str)

\* 🡪 (repetition operator) repeats a string a certain number of times

in 🡪 returns True if a string is contained in another string, False otherwise

not in 🡪 opposite of in

is 🡪 returns True if two strings are identical, False otherwise

is not 🡪 opposite of is

8.4

A list is a collection of data, each piece of data is called an element.

The elements of a list are enclosed in square brackets and divided by commas.

The data types in a list do not have to be uniform.

To view the contents of a list the print function can be used:

print (nameoflist)

The list function returns characters of a string or of iterable objects turning them into a list:

list (iterableobject)

8.5

The main operators for lists are:

+ 🡪 concatenates two lists respecting order of insertion

\* 🡪 creates copies of a list and merges them

in 🡪 returns True if an (entire) element is found within a list, False otherwise

not in 🡪 opposite of in

8.6

An index is an integer which identifies the position of each element within a sequence

Indexing starts from 0 on the left and ends at lengthofthesequence-1 or starts from -1 on the right and ends at -lengthofthesequence on the left

To access an item within a sequence the syntax is:

Nameofsequence [index]

If the sequence is mutable the same syntax can be used to modify the element of the list:

Nameofsequence [index] = newvalue

8.7

Slicing is a technique used in Python to select multiple elements of a sequence.

The syntax to select a slice of a sequence is:

Nameofsequence [startindex : endindex : step]

The start index is included, the end index is excluded.

If start is omitted, the slice starts from index=0, if end is omitted the slice ends at the last element if step is omitted it is equal to 1.

If one parameter is negative, then they must all be or, if step is negative, start and end can be positive but they must be inverted.

8.8

Methods are functions that belong to a certain object, their syntax is different than that of functions because the name of the object must always be specified.

The general syntax for methods is:

Nameofvariable.nameofmethod(arguments)

The len function calculates the length of a string:

len (string)

The min function returns the smallest value of a sequence (minimum values are special characters, the alphabetical order) and the max function returns the highest value:

min (sequence)

max (sequence)

The main string methods are:

.upper() 🡪 transforms in uppercase

.lower()🡪transforms in lowercase

.capitalize()🡪first character is uppercase, all others lowercase

.strip()🡪removes all whitespace characters or the argument

.find(sub)🡪finds the subrstring of the argument in the string and returns the first index. Start and end are optional arguments and if sub is not found it returns -1.

.replace(old,new)🡪replaces all occurences of old with new

.startswith(prefix)🡪True if the string starts with prefix, false otherwise. Start and end are optional

.endswith(suffix)🡪True if the string ends with suffix, false otherwise. Start and end are optional

.count(sub)🡪returns the number of occurences of sub in the string

.split(iterable)🡪separates a string in words. The optional argument separator can be used instead of the space. The optional argument maxsplit allows to set a maximum number of splits.

.join(iterable)🡪concatenates all elements of an iterable object containing strings only. A string to be used as separator must be specified: separator=’’ 🡪separator.join(iterable)

8.9

The sorted function returns a new sorted list in ascending order:

sorted(list)

The sum function returns the sum of all elements of a list (they must all be numbers):

sum(list)

8.10

The main list methods are:

.append(element)🡪appends new element to the list

.insert(index, element)🡪inserts the element in the index position

.remove(element)🡪removes the first occurrence of the element

.pop([index])🡪removes and returns the element at the index. If index is omitted it pops the last element

.extend(list2)🡪appends list2 to the list

.index(element)🡪returns the index of the first occurrence of the element

.sort()🡪sorts the list in ascending order (for descending order use .sort(reverse=True)

.reverse()🡪reverses the elements of the list

.clear()🡪removes all the elements of a list

.count(element)🡪counts the number of times element is found in the list

.copy()🡪returns a copy of the list

8.11

Traversing is sequential access to all elements of a string, list or iterable object.

8.12

Lists of lists can be created, the first index refers to the list and the second to the element of that specific list.