Ramila Jane B. Abobo
BSCS-3C

FINALS - Assignment #2

Lists

1. Defining a List
   - ✓ It also known as containers or collections, List is an ordered sequence element or sequence of objects. List can contain the different data types like integers, strings and floats. Indexing starts at 0 for the first element, 1 for the second element, and so on. It is used to store the item and also if defined by the having values between square brackets [ ].

2. List Syntax
   - ✓ The syntax of list is my_list = [element1, element2, element3, ...]

3. Accessing List Elements
   - ✓ In accessing the list, you can use the index within square brackets **[],** The Indexing starts from **0** for the first element, **1** for the second element, and so on. Also, slicing allows you to take a subset of elements out of a list.

4. Loop through a List
   - ✓ Python has several ways to loop through a list, including list comprehensions, while loops, and for loops.
   - ✓ **Using a for Loop:**

```
numbers = [1, 2, 3, 4, 5]

# Using a for loop to iterate through the list

for number in numbers:

 print(number)
```

   - ✓ **Using List Comprehensions:**

```
numbers = [1, 2, 3, 4, 5]

# Using list comprehension to print each element

[print(number) for number in numbers]
```

   - ✓ **Using a while Loop**

```
numbers = [1, 2, 3, 4, 5]

i = 0
```

```
# Using a while loop to iterate through the list
while i < len(numbers):
 print(numbers[i])
   i += 1
```

5. List Length
   - ✓ Python's built-in len() method can be used to determine a list's length. The number of elements in the given list is returned by the len() method.

```
my_list = [10, 20, 30, 40, 50]
# Find the length of the list
length = len(my_list)
print("Length of the list:", length)
```

6. Add Items in the List
   - ✓ When adding items to a list in Python, you can use the insert() function to add an item at a given index or the append() method to add an item at the end of the list.

```
my_list = []
# Adding items to the list
my_list.append("apple")
my_list.append("banana")
my_list.append("orange")
# Adding an item at a specific index
my_list.insert(1, "grape")
print(my_list)
```

7. Remove Item from a List
   - ✓ Python offers two methods for removing items from lists: the remove() method, which you should use if you know the item's value, and the pop() method, which you should use if you know its index.

```
# Creating a list
my_list = ["apple", "banana", "orange", "grape"]
```

```
# Removing an item by value
my_list.remove("banana")
# Removing an item by index
removed_item = my_list.pop(1)
print(my_list)
print("Removed item:", removed_item)
```

8. The List () Constructor
   - ✓ To make a new list in Python, use the list() constructor. The list() constructor can be used to generate a new list given the items of an iterable (such as a string, tuple, or other list() as an argument.

```
# Using list() constructor to create a list
my_list = list("hello")
print(my_list)  # Output: ['h', 'e', 'l', 'l', 'o']
```

9. List Methods
   - ✓ Lists are objects in Python that include several built-in functions that let you interact with and manipulate lists.
   - ✓ append(): Adds an element to the end of the list.
   - ✓ extend(): Appends elements from another list to the end of the current list.
   - ✓ insert(): Inserts an element at a specified position in the list.
   - ✓ remove(): Removes the first occurrence of a specified value from the list.
   - ✓ pop(): Removes and returns the element at a specified index in the list.

10. Nested Lists
    - ✓ A nested list in Python is a list that shows up as an element inside another list. This enables you to design a data structure that can represent multidimensional data or several hierarchical layers.

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(nested_list[0])  # Output: [1, 2, 3]
print(nested_list[1][1])  # Output: 5
```