Malaluan, Nervin C. BSCS 3C

Finals - Assignment 2

Defining a List

A list in Python is an ordered, mutable collection of items. You can store various data types (numbers, strings, booleans, even other lists) within a single list. You create a list using square brackets [] and separate elements with commas.

```
fruits = ["apple", "banana", "cherry"]
numbers = [1, 2, 3, 4, 5]
mixed_list = ["hello", 10.5, True]
```

List Syntax

Elements are enclosed in square brackets [] and separated by commas.
Lists can be empty (denoted by []).

Accessing List Elements

Python uses zero-based indexing to access elements. The first element has index 0, the second has index 1, and so on.
You use square brackets after the list name and the index of the element you want to access or modify.

```
fruits = ["apple", "banana", "cherry"]

first_fruit = fruits[0]  # first_fruit will be "apple"
last_fruit = fruits[2]   # last_fruit will be "cherry"

# You can also use negative indexing to access elements from the end:
last_fruit = fruits[-1]  # last_fruit will also be "cherry"
```

Looping Through a List

Use a for loop to iterate over each element in the list.

```
fruits = ["apple", "banana", "cherry"]

for fruit in fruits:
    print(fruit)  # Prints each fruit on a new line
```

List Length

Use the len() function to determine the number of elements in a list.

```
fruits = ["apple", "banana", "cherry"]

num_fruits = len(fruits)  # num_fruits will be 3
```

Adding Items to the List

Use the append() method to add an element to the end of the list.
Use the insert(index, element) method to insert an element at a specific index.

```
fruits = ["apple", "banana", "cherry"]

fruits.append("orange")  # Add "orange" to the end
fruits.insert(1, "mango")  # Insert "mango" at index 1

print(fruits)  # Output: ["apple", "mango", "banana", "cherry", "orange"]
```

Removing Items from a List

Use the remove(element) method to remove the first occurrence of a specified element (by value).
Use the pop(index) method to remove and return the element at a specific index (or the last element by default).

```
fruits = ["apple", "mango", "banana", "cherry", "orange"]

fruits.remove("mango")  # Remove "mango"
removed_fruit = fruits.pop(2)  # Remove and return the element at index 2 (which is "banana")

print(fruits)  # Output: ["apple", "cherry", "orange"]
```

The list() Constructor

You can create a list from other iterables (like strings or tuples) using the list() constructor.

```
my_string = "hello"
my_list = list(my_string)  # my_list will be ['h', 'e', 'l', 'l', 'o']
```

List Methods

Python lists provide various methods for manipulating and working with elements. Some common methods include:
sort(): Sorts the list elements in ascending order (in-place modification).
reverse(): Reverses the order of elements in the list (in-place modification).
index(element): Returns the index of the first occurrence of a specified element.
clear(): Removes all elements from the list.

Nested Lists

Lists can contain other lists to create multi-dimensional structures.

```
nested_list = [["apple", "banana", "cherry"], ["mango", "kiwi"]]

first_sublist = nested_list[0]  # first_sublist will be ["apple", "banana", "cherry"]
second_fruit = first_sublist[1]  # second_fruit will be "banana"
```