Largado, Ardean Josh E.
CS3C

- Defining a List
- List Syntax
- Accessing List Elements
- Loop through a List
- List Length
- Add Items in the List
- Remove Item from a List
- The List () Constructor
- List Methods
- Nested Lists

- Lists can contain elements of different data types, including strings, numbers, and even other lists.

```
# Defining a list of numbers
numbers = [1, 2, 3, 4, 5]
print(numbers)  # Output: [1, 2, 3, 4, 5]

# Defining a list of strings
fruits = ["apple", "banana", "cherry", "date"]
print(fruits)  # Output: ['apple', 'banana', 'cherry', 'date']

# Defining a list with mixed data types
mixed_list = [1, "apple", True, 3.14]
print(mixed_list)  # Output: [1, 'apple', True, 3.14]

# Defining a list of lists (nested list)
nested_list = [[1, 2, 3], ["a", "b", "c"]]
print(nested_list)  # Output: [[1, 2, 3], ['a', 'b', 'c']]
```

-

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

print(fruits[0])  # Output: apple
print(fruits[2])  # Output: cherry
print(fruits[-1])  # Output: date (accessing from the end)
```

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

for fruit in fruits:
    print(fruit)
```

```
apple
banana
cherry
date
```

```
# List definition
fruits = ["apple", "banana", "cherry", "date"]

# Get the length of the list
list_length = len(fruits)

# Print the length of the list
print("Number of elements in the list:", list_length)
```

```
Number of elements in the list: 4
```

Adding Items to a List: You can add items to a list using various methods such as append(), insert(), and list concatenation (+).

```
# List definition
fruits = ["apple", "banana", "cherry"]

# Adding items using append()
fruits.append("date")
print(fruits)  # Output: ['apple', 'banana', 'cherry', 'date']

# Adding items using insert() at a specific position
fruits.insert(1, "grape")
print(fruits)  # Output: ['apple', 'grape', 'banana', 'cherry', 'date']

# Adding items using list concatenation
more_fruits = ["kiwi", "mango"]
fruits += more_fruits
print(fruits)  # Output: ['apple', 'grape', 'banana', 'cherry', 'date', 'kiwi', 'man
```

Removing Item from a List: You can remove items from a list using methods like remove(), pop(), and del.

```python
# List definition
fruits = ["apple", "banana", "cherry", "date"]

# Removing an item using remove()
fruits.remove("banana")
print(fruits)  # Output: ['apple', 'cherry', 'date']

# Removing an item using pop() at a specific index
removed_fruit = fruits.pop(1)
print(fruits)  # Output: ['apple', 'date']
print("Removed fruit:", removed_fruit)  # Output: Removed fruit: cherry

# Removing an item using del at a specific index
del fruits[0]
print(fruits)  # Output: ['date']
```

The list() Constructor: You can create a list using the list() constructor by passing an iterable object such as a string or another list.

```python
# Using list() constructor with a string
word_list = list("Python")
print(word_list)  # Output: ['P', 'y', 't', 'h', 'o', 'n']

# Using list() constructor with another list
num_list = list(range(1, 5))
print(num_list)  # Output: [1, 2, 3, 4]
```