Alyanna Marie N. Bilazon
CS3C

Lists
Defining a List
> In Python, a list is a collection of items enclosed in square brackets []. Lists can contain elements of different data types, such as numbers, strings, booleans, and even other lists.
>
```python
my_list = [1, 2, 3, 4, 5]
```

List Syntax
>
```python
my_list = [item1, item2, item3, ...]
```

Accessing List Elements
> List elements can be accessed using their index positions, starting from 0. You can also use negative indexing to access elements from the end of the list.
>
```python
my_list = [1, 2, 3, 4, 5]
print(my_list[0])  # Output: 1
```

Loop through a List
> You can use a for loop to iterate through each element in a list and perform operations on them.
>
```python
my_list = [1, 2, 3, 4, 5]
for item in my_list:
    print(item)
```

List Length
> The length of a list can be obtained using the len() function, which returns the number of elements in the list.
>
```python
my_list = [1, 2, 3, 4, 5]
print(len(my_list))  # Output: 5
```

Add Items in the List
> You can add items to a list using various methods, such as the append() method to add an item at the end, the insert() method to insert an item at a specific position, or using the + operator to concatenate two lists.
>
```python
my_list = [1, 2, 3]
my_list.append(4)
print(my_list)  # Output: [1, 2, 3, 4]
```

Remove Item from a List

➢ Items can be removed from a list using methods like remove() to remove a specific item, pop() to remove an item at a specific index, or del statement to remove an item by index.

➢
```python
my_list = [1, 2, 3, 4, 5]
my_list.remove(3)
print(my_list)   # Output: [1, 2, 4, 5]
```

The List () Constructor

➢ You can also create a list using the list() constructor.

➢
```python
my_list = list((1, 2, 3))
print(my_list)   # Output: [1, 2, 3]
```

List Methods

➢ Lists have built-in methods such as sort(), reverse(), count(), and more, which allow you to perform various operations on lists.

➢
```python
my_list = [1, 2, 3]
my_list.append(4)
print(my_list)   # Output: [1, 2, 3, 4]
```

Nested Lists

➢ Lists can contain other lists as elements, creating nested structures.

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