01. LISTS

M2U5P1 COMPUTER SCIENCE

WHAT ARE LISTS?

LISTS

- Lists are used to store multiple items in a single variable
- Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

Sequence Types:	`list`, `tuple`, `range`
Mapping Type:	`dict`
Set Types:	`set`, `frozenset`

LIST

- Lists are created using square brackets:

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

- List items are ordered, changeable, and allow duplicate values.

ORDERED

- When we say that lists are ordered, it means that the items have a defined order, and that order will not change
- If you add new items to a list, the new items will be placed at the end of the list
- There are some list methods that will change the order, but in general: the order of the items will not change

LIST ORDER

??? ??? ???

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

LIST ORDER

0 ??? ???

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

LIST ORDER

0 1 ???

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

LIST ORDER

```
                    0          1          2
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

LIST ORDER

- Lists always start from 0, which means that the first item in the list will have an index of 0

```
                    0          1          2
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

CHANGEABLE

- The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

ALLOW DUPLICATES

- Since lists are indexed, lists can have items with the same value:

```
thislist = ["apple", "banana", "cherry", "apple", "cherry"]
print(thislist)
```

LIST LENGTH

- To determine how many items a list has, use the len() function:

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
```

```
3
```

LIST ITEMS - DATA TYPES

- List items can be of any data type:

```
list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
```

- A list can contain different data types:

```
list1 = ["abc", 34, True, 40, "male"]
```

ACCESSING LIST ITEMS

ACCESS ITEMS

- List items are indexed and you can access them by referring to the index number:

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

ACCESS ITEMS

- List items are indexed and you can access them by referring to the index number:

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

- Keep in mind that the first item has an index of 0

NEGATIVE INDEXING

- Negative indexing means start from the end

```
thislist = ["apple", "banana", "cherry"]
print(thislist[-1])
```

NEGATIVE INDEXING

- Negative indexing means start from the end

```
thislist = ["apple", "banana", "cherry"]
print(thislist[-1])
```

- -1 refers to the last item, -2 refers to the second last item etc.

RANGE OF INDEXES

- You can specify a range of indexes by specifying where to start and where to end the range.
- When specifying a range, the return value will be a new list with the specified items.

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])
```

- What items will be in the new list?

RANGE OF INDEXES

- You can specify a range of indexes by specifying where to start and where to end the range.
- When specifying a range, the return value will be a new list with the specified items.

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])
```

- The search will start at index 2 (included) and end at index 5 (not included).

RANGE OF INDEXES – FROM BEGINNING

- By leaving out the start value, the range will start at the first item
- This example returns the items from the beginning to, but NOT included, "kiwi":

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])
```

RANGE OF INDEXES – FROM BEGINNING

- By leaving out the start value, the range will start at the first item
- This example returns the items from the beginning to, but NOT included, "kiwi":

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])
```

```
['apple', 'banana', 'cherry', 'orange']
```

RANGE OF INDEXES – UNTIL END

- By leaving out the end value, the range will go on to the end of the list
- This example returns the items from "cherry" and to the end:

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:])
```

RANGE OF INDEXES – UNTIL END

- By leaving out the end value, the range will go on to the end of the list
- This example returns the items from "cherry" and to the end:

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:])
```

```
['cherry', 'orange', 'kiwi', 'melon', 'mango']
```

CHECK IF ITEM EXISTS

- To determine if a specified item is present in a list use the in keyword:

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
  print("Yes, 'apple' is in the fruits list")
```

CHANGE ITEM VALUE

- To change the value of a specific item, refer to the index number:

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "kiwi"

print(thislist)
```

CHANGE ITEM VALUE

- To change the value of a specific item, refer to the index number:

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "kiwi"

print(thislist)
```

```
['apple', 'kiwi', 'cherry']
```

CHANGE A RANGE OF ITEM VALUES

- To change the value of items within a specific range, define a list with the new values, and refer to the range of index numbers where you want to insert the new values:

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]
thislist[1:3] = ["blackcurrant", "watermelon"]
print(thislist)
```

CHANGE A RANGE OF ITEM VALUES

- If you insert ***more*** items than you replace, the new items will be inserted where you specified, and the remaining items will move accordingly:

```
thislist = ["apple", "banana", "cherry"]
thislist[1:2] = ["blackcurrant", "watermelon"]
print(thislist)
```

CHANGE A RANGE OF ITEM VALUES

- If you insert *less* items than you replace, the new items will be inserted where you specified, and the remaining items will move accordingly:

```
thislist = ["apple", "banana", "cherry"]
thislist[1:3] = ["watermelon"]
print(thislist)
```

ANY QUESTIONS?

THE END