![[00-Index-of-Python-Fundamentals]]

List

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.

Lists are created using square brackets:

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

List Items

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

List items are indexed, the first item has index `[0]`, the second item has index `[1]` etc.

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.**
Changeable	The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.
AllowDuplicates	Since lists are indexed, lists can have items with the same value:

```python
# Lists allow duplicate values:

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

Functions to use with List

Function	
`len()`	To determine how many items a list has
`type()`	lists are defined as objects with the data type 'list'
`list()`	use the list() constructor when creating a new list (optional)

```python
# Determines how many items a list has
print(len(thislist))

# List items can be of any data type
list1 = ["apple", "banana", "cherry"]   # String data types
list2 = [1, 5, 7, 9, 3]                 # int data types
list3 = [True, False, False]            # boolean data types

# A list can contain different data types:
list4 = ["abc", 34, True, 40, "male"]

# From Python's perspective,
# lists are defined as objects with the data type 'list'
print(type(thislist)) # Output Should be : <class 'list'>

# The `list()` Constructor :
# It is also possible to use the list() constructor when creating a new
list.

morefruits = list(("apple", "banana")) # note the double round-brackets
print(morefruits)
```

Access List Items | ReadMore

Access Items	`thislist[1]`	**List items are indexed and you can access them by referring to the index number**
Negative Indexing	`thislist[-1]`	Negative indexing means start from the end
		`-1` refers to the last item, `-2` refers to the second last item
Range of Indexes	`thislist[2:5]`	specifys a range of indexes by specifying where to start and where to end the range
Range of Negative Indexes	`thislist[-4:-1]`	Specify negative indexes if you want to start the search from the end of the list

> **Note:** Always Remember, The first item has index 0.

PYTHON SCRIPT

```
# Access List items by referring to the index number

# Rememnber First Item, is indexed at '0'

thislist = ["apple","banana","cherry"]

print("0.Displaying the whole List : ",thislist)
print("1.Accessing the 2nd Item : ",thislist[1])

# Negative indexing means counting starts from the end
print("2.Counting from the End : ",thislist[-1])

# Know the current list of items before, adding new ones.
print("3.Total Items before adding new : ",len(thislist))

# First, lets add(append) some more list items
morefruits = ("orange","kiwi","melon","mango")
thislist.extend(morefruits)
print("4.List with New Fruits Items : ",thislist)

# Find out the total items in the list
print("5.Total item in the list after adding more : ",len(thislist))
```

OUTPUT

```
# OUTPUT

0.Displaying the whole List :  ['apple', 'banana', 'cherry']
1.Accessing the 2nd Item :  banana
2.Counting from the End :  cherry
3.Total Items before adding new :  3
4.List with New Fruits Items :  ['apple', 'banana', 'cherry', 'orange',
'kiwi', 'melon', 'mango']
5.Total item in the list after adding more :  7
['cherry', 'orange', 'kiwi']
```

Check if Item Exists

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

```
# Check if "apple" is present in the list:
thislist = ["apple","banana","cherry"]
if "apple" in thislist:
        print("Yes, 'apple is in the fruit-list'")
```