```
# list , tuple , set , Dictionary

# list: Lists are used to store multiple items in a single variable.
    #  List items are ordered, changeable, and allow duplicate values.
    #  Lists are created using square brackets

student = ["atharva", "santosh", "akshay", "krutika", "om", "shilpa", 12 , 34.878, True]
print(type(student))

# eg:
thislist = ["apple", "banana", "cherry"]
print(thislist)

# List Length
# To determine how many items a list has, use the len() function:
thislist = ["apple", "banana", "cherry"]
print(len(thislist))

# List Items - Data Types
# List items can be of any data type:

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

print(list1)
print(list2)
```

```
print(list3)


# eg.2
list1 = ["abc", 34, True, 40.6, "male"]
print(list1)

# type()
list1 = ["abc", 34, True, 40.6, "male"]
print(type(list1))

student = ["atharv", "akshay" , "santosh"]
student = list(("athara", "santosh", "akshay"))
print(student)

# The list() Constructor
# It is also possible to use the list() constructor when creating a  list.
# eg.1
thislist = list(("apple", "banana", "cherry")) # note the double round-brackets
print(thislist)

# access the list items.

# List items are indexed and you can access them by referring to the index number:
#eg.1
thislist = ["apple", "banana", "cherry", "dragonfruit" , "Orange", "Grapes", "Watermelon"]
print(thislist[5])

# Negative Indexing:
```

Negative indexing means start from the end

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

eg:1

thislist = ["apple", "banana", "cherry", "dragonfruit" , "Orange", "Grapes", "Watermelon"]

print(thislist[-5])

Range of Indexes

You can specify a range of indexes by specifying where to start and where to end the range.

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

eg.1

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:6])

eg.2

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:])

eg.3

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[:4])

Range of Negative Indexes

Specify negative indexes if you want to start the search from the end of the list:

eg.1

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:-1])

# eg.2
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:])

#eg.3
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:-3])
```

Check if Item Exists

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

```
#eg.1
thislist = ["apple", "banana", "cherry", "orange"]
if "apple" in thislist:
    print("yes apple in this list")
```

change the value

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

```
# eg.
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)
```

Change a Range of Item Values:

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

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

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

# Insert Items
# To insert a new list item, without replacing any of the existing values, we can use the insert() method.
# eg.1
thislist = ["apple", "banana", "cherry"]
thislist.insert(2, "watermelon")
print(thislist)

# Append Items:
# To add an item to the end of the list, use the append() method:
# eg.1
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
```

```
print(thislist)


#Extend List
# To append elements from another list to the current list, use the extend() method.
#eg.1
student_class = ["atharva" , "krutika", "archie", "bhumika"]
student_school = ["shyam", "pravina", "aashie", "kinmay"]
student_class.extend(student_school)
print(student_class)


#eg.2
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)


# Remove Specified Item
#The remove() method removes the specified item.


# eg.1
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)


# eg.2
thislist = ["apple", "banana", "cherry"]
thislist.pop(2)
print(thislist)
```

```
# If you do not specify the index, the pop() method removes the last item.
# eg.1
thislist = ["apple", "banana", "cherry"]
thislist.pop()
print(thislist)

# The del keyword also removes the specified index:
# eg.1
thislist = ["apple", "banana", "cherry"]
del thislist[0]
print(thislist)

# The del keyword can also delete the list completely.
# eg.1
thislist = ["apple", "banana", "cherry"]
del thislist

# Clear the List
thislist = ["apple", "banana", "cherry"]
thislist.clear()
print(thislist)

# sort list
# Sort the list alphabetically:
# eg.1
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort()
print(thislist)
```

Sort the list numerically:

eg.1

```
thislist = [100, 50, 65, 82, 23]
thislist.sort()
print(thislist)
```

Sort Descending:

eg.1

```
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort(reverse = True)
print(thislist)
```

eg.2

```
thislist = [100, 50, 65, 82, 23]
thislist.sort(reverse = True)
print(thislist)
```

Reverse Order:

eg.1

```
thislist = ["banana", "Orange", "Kiwi", "cherry"]
thislist.reverse()
print(thislist)
```

case senstive sort

By default the sort() method is case sensitive, resulting in all capital letters being sorted before lower case letters:

```
thislist = ["banana", "Orange", "Kiwi", "cherry"]
```

```
thislist.sort()
print(thislist)

# Case Insensitive Sort:
thislist = ["banana", "Orange", "Kiwi", "cherry"]
thislist.sort(key = str.lower)
print(thislist)

# Copy a List
# you can use the built-in List method copy() to copy a list.
# eg.1
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()
print(mylist)

# Use the list() method
# Another way to make a copy is to use the built-in method list().
# eg.1
thislist = ["apple", "banana", "cherry"]
mylist = list(thislist)
print(mylist)

# Use the slice Operator ":"
# eg.1
thislist = ["apple", "banana", "cherry"]
mylist = thislist[:]
print(mylist)
```

```
# Join Two Lists
# One of the easiest ways are by using the + operator.
# eg.1
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]
list3 = list1 + list2
print(list3)

# Use the extend() method to add list2 at the end of list1:
# eg.1
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]
list1.extend(list2)
print(list1)
```

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list