Lists are used to store multiple items in a single variable. 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.

In [2]:
```
#data type of a list
mylist = ["apple", "banana", "cherry"]
print(type(mylist))
```

```
<class 'list'>
```

In [3]:
```
#creating a list
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

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

In [4]:
```
#lists can allow duplicates
thislist = ["apple", "banana", "cherry", "apple", "cherry"]
print(thislist)
```

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

In [7]:
```
#length of list
thislist = ["apple", "banana",1,1.0,]
print(len(thislist))
```

```
4
```

In [ ]:
```
#list can be any datatype inside it
list1 = ["abc", 34, True, 40, "male"]
```

In [8]:
```
#list can be constructed by using list function
thislist = list(("apple", "banana", "cherry")) # note the double round-brackets
print(thislist)
```

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

In [12]:
```
#access list items
thislist = ["apple", "banana", "cherry"]#index always starts from 0
print(thislist[2])
print(thislist[1])
```

```
cherry
banana
```

In [13]:
```
#Negative Indexing
#Negative indexing means start from the end

#-1 refers to the last item, -2 refers to the second last item etc.
thislist = ["apple", "banana", "cherry",1,2,"sh"]
print(thislist[-1])
```

```
sh
```

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

In [14]:
```
#range of index
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])#start:end
```

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

In [17]:
```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:-1])
```

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

Check if Item Exists To determine if a specified item is present in a list use the in keyword:

In [19]:
```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
  print("Yes, 'apple' is in the fruits list")
else:
  print("NO")
```

```
Yes, 'apple' is in the fruits list
```

Change List Items

In [24]:
```
thislist = ["apple", "banana", "cherry"]
thislist[0] = "blackcurrant"
print(thislist)
#used to change the items in a list of specified index
```

```
['blackcurrant', 'banana', 'cherry']
```

Insert Items

In [29]:
```
#The insert() method inserts an item at the specified index:
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "watermelon")
print(thislist)
```

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

Append Items

In [32]:
```
#To add an item to the end of the list, use the append() method
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

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

Extend List

In [33]:
```
thislist = ["apple", "banana", "cherry"]
tropical = ["mango", "pineapple", "papaya"]
thislist.extend(tropical)
print(thislist)
```

```
['apple', 'banana', 'cherry', 'mango', 'pineapple', 'papaya']
```

Remove List Items

In [34]:
```
#The remove() method removes the specified item.
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

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

Remove Specified Index

In [35]:
```
#The pop() method removes the specified index.
thislist = ["apple", "banana", "cherry"]
thislist.pop(0)
print(thislist)
```

```
['banana', 'cherry']
```

In [37]:
```
#The del keyword also removes the specified index
thislist = ["apple", "banana", "cherry"]
del thislist
```

In [38]:
```
thislist = ["apple", "banana", "cherry"]
del thislist[0]
print(thislist)
```

```
['banana', 'cherry']
```

In [39]:
```
#The clear() method empties the list.
thislist = ["apple", "banana", "cherry"]
thislist.clear()
print(thislist)
#del thislist olp
# .clear() olp []
```

```
[]
```

Loop Through a List

In [46]:
```
thislist = ["apple", "banana", "cherry"]
for i in thislist:
  print(i,end=" ")
```

```
apple banana cherry
```

In [48]:
```
#you can also loop through the list items by referring to their index number.

#Use the range() and len() functions to create a suitable iterable.
thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)): #for 2 in range 3
  print(thislist[i],end=" ")  #list[2]
```

```
apple banana cherry
```

Using a While Loop

In [52]:
```
#Remember to increase the index by 1 after each iteration.
thislist = ["apple", "banana", "cherry","dhd","gbdfhb",'shyam',"ygdyd"]
i=0
while i < len(thislist):
  print(thislist[i])
  i = i + 1
```

```
cherry
dhd
gbdfhb
shyam
ygdyd
```

Looping Using List Comprehension

In [53]:
```
thislist = ["apple", "banana", "cherry"]
[print(x) for x in thislist]
```

```
apple
banana
cherry
```

Out[53]:
```
[None, None, None]
```

List Comprehension

In [54]:
```
#with out list comprehension
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist =[]
for x in fruits: # ith index --
    if "a" in x:
      newlist.append(x)

print(newlist)
```

```
['apple', 'banana', 'mango']
```

In [57]:
```
#with list comprehension
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if "a" in x]
n=[a for a in fruits if "a" not in a]
print(newlist)
print(n)
```

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

newlist = [expression for item in iterable if condition == True]

In [59]:
```
newlist = [x for x in range(10) if x >5]
print(newlist)
```

```
[6, 7, 8, 9]
```

In [60]:
```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist1 = [x.upper() for x in fruits]
newlist2 = [ x for x in fruits if x == "banana"]
print(newlist1)
print(newlist2)
```

```
['APPLE', 'BANANA', 'CHERRY', 'KIWI', 'MANGO']
['banana']
```

Sort Lists

In [70]:
```
#List objects have a sort() method that will sort the list alphanumerically
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort()
print(thislist)
```

```
['banana', 'kiwi', 'mango', 'orange', 'pineapple']
```

In [73]:
```
thislist = [100, 50, 65, 82, 23]
thislist.sort(reverse=True)
print(thislist)
```

```
[100, 82, 65, 50, 23]
```

In [72]:
```
#To sort descending, use the keyword argument reverse = True
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort(reverse = True)
print(thislist)
```

```
['pineapple', 'orange', 'mango', 'kiwi', 'banana']
```

Reverse Order

In [74]:
```
#What if you want to reverse the order of a list, regardless of the alphabet?

#The reverse() method reverses the current sorting order of the elements.
thislist = ["banana", "Orange", "Kiwi", "cherry"]
thislist.reverse()
print(thislist)
```

```
['cherry', 'Kiwi', 'Orange', 'banana']
```

Copy a List

You cannot copy a list simply by typing list2 = list1, because: list2 will only be a reference to list1, and changes made in list1 will automatically also be made in list2.

There are ways to make a copy, one way is to use the built-in List method copy().

In [75]:
```
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()
print(mylist)
```

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

In [76]:
```
#with out using copy
thislist = ["apple", "banana", "cherry"]
mylist = list(thislist)
print(mylist)
```

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

Join Lists

In [78]:
```
#using + method
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]

list3 = list1 + list2
list1.extend(list2)
print(list3)
print(list2)
```

```
['a', 'b', 'c', 1, 2, 3]
[1, 2, 3]
```

In [ ]:
```
#using apend method
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

for x in list2:
  list1.append(x)

print(list1)
```

In [ ]:
```
#using extend method
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

list1.extend(list2)
print(list1)
```

In [ ]:
```

```