""""

append() - Add an element to the end of the list

extend() - Add all elements of a list to the another list

insert() - Insert an item at the defined index

remove() - Removes an item from the list

pop() - Removes and returns an element at the given index

clear() - Removes all items from the list

index() - Returns the index of the first matched item

count() - Returns the count of the number of items passed as an argument

sort() - Sort items in a list in ascending order

reverse() - Reverse the order of items in the list

copy() - Returns a shallow copy of the list

"""

# create a list

# arr = []

# print(type(arr))

#

# arr\_value = [1, 2, 3, 4, 5]

# print(arr\_value)

#

# my\_list = [1, "hello", 1.5]

# print(my\_list)

#

# nested\_list = [1, "hello", [1, 2, 3, 4]]

# print(nested\_list)

"""Accessing a list"""

alphabet = ["a", "b", "c", "d", "e", ["x", "y", "z"]]

# index

# print(alphabet[0])

# print(alphabet[-1])

# print(alphabet[5]) # IndexError: list index out of range

# print(alphabet[5][0]) # nested list access

# print(alphabet[-1][0]) # nested list access

"""Slice list in python"""

# # list[start\_index : end\_index] note: start\_index is inclusive but end\_index is exclusive

# number = [1, 2, 3, 4, 5, 6, 7, 8, 9]

# # 0 1 2 3 4 5 6 7 8

#

# # print element 3rd to 6th

# print(number[2:6])

#

# # elements 5th to end

# print(number[4:])

#

# # elements beginning to end

# print(number[:])

""" Add or update list elements"""

# even\_list = [2, 4, 6, 9, 10]

# odd\_list = [1, 3, 5, 7, 9]

# print(even\_list)

# update 9 with 8

# even\_list[3] = 8

# print(even\_list)

# add both even and odd list

# use + operator

# all\_ele\_list = odd\_list + even\_list

# print(all\_ele\_list)

# use extend method

even\_list = [2, 4, 6, 9, 10]

odd\_list = [1, 3, 5, 7, 9]

# even\_list.extend(odd\_list)

# print(even\_list)

# odd\_list.append(even\_list)

# print(odd\_list)

# I wish to append a value atr the end of list

# eg: add 11 at the end of odd list

# odd\_list.append(11)

# print(odd\_list)

# all\_list = odd\_list.append(even\_list)

# print(all\_list)

""" initialise a list"""

# print([0] \* 3) # o/p [0, 0, 0]

""" insert method of list"""

# eg = [1, 3]

# eg.insert(1, 2)

# print(eg)

"""delete method of a list"""

# del\_list = [1, 3, 5, 7, 9]

# print(del\_list)

# del one item by index

# del del\_list[1]

# print(del\_list)

#

# del del\_list[2]

# print(del\_list)

# del multiple items

# del del\_list[1:3]

# print(del\_list)

#

# # delete a list

# del del\_list # this step deletes the entire list

# print(del\_list) # NameError: name 'del\_list' is not defined

"""remove and pop method"""

# re\_po\_list = ["a", "b", "c", "d", "e"]

# remove a value in list

# print("before remove= ", re\_po\_list)

# re\_po\_list.remove("a")

# print("after remove =", re\_po\_list)

# pop method

# print("before pop= ", re\_po\_list)

# # re\_po\_list.pop() # if no index specified it will remove last element

# re\_po\_list.pop(2) # if index specified it will remove index element

# print("after pop= ", re\_po\_list)

"""clear method """

# re\_po\_list.clear()

# print("after clear= ", re\_po\_list)

# index() - Returns the index of the first matched item

# re\_po\_list = ["a", "b", "c", "d", "e"]

# print(re\_po\_list.index("d"))

# count() - Returns the count of the number of items passed as an argument

# re\_po\_list = ["a", "b", "c", "d", "e", "e"]

# print(re\_po\_list.count("e"))

# sort() - Sort items in a list in ascending order

elements = [1, 3, 2, 6, 5]

# print(elements)

# elements.sort()

# print(elements)

# reverse() - Reverse the order of items in the list

elements.reverse()

# print(elements)

# copy() - Returns a shallow copy of the list

b = elements.copy()

print(b)

""" iterate over a list"""

for i in range(len(b)-1):

print("before =", b[i], b[i + 1])

b[i], b[i+1] = b[i+1], b[i]

print("after =", b[i], b[i + 1])

print(b)