1. What is the output of the following list operation

aList = [10, 20, 30, 40, 50, 60, 70, 80]

print(aList[2:5])

print(aList[:4])

print(aList[3:])

* [20, 30, 40, 50]  
  [10, 20, 30, 40]  
  [30, 40, 50, 60, 70, 80]
* **[30, 40, 50]  
  [10, 20, 30, 40]  
  [40, 50, 60, 70, 80]**

**Explanation:**

Python list collection is ordered and changeable. The list also allows duplicate members. To get a sublist out of the list, we need to specify the range of indexes.  To get a sublist, we need to specify where to start and where to end the range.

**Syntax**: list[start:end] If start is missing it takes 0 as the starting index

2. What is the output of the following list function?

sampleList = [10, 20, 30, 40, 50]

sampleList.append(60)

print(sampleList)

sampleList.append(60)

print(sampleList)

* [10, 20, 30, 40, 50, 60]  
  [10, 20, 30, 40, 50, 60]
* **[10, 20, 30, 40, 50, 60]  
  [10, 20, 30, 40, 50, 60, 60]**

**Explanation:**

The append() method is used to add an item at the end of a list. Also, the list allows duplicate items.

3. In Python, list is mutable

* False
* **True**

**Explanation**:

The list collection is ordered and changeable. A mutable object can be changed after it is created. So we can update or remove elements from a list once it is created.

4. Select all the correct options to copy a list

aList = ['a', 'b', 'c', 'd']

* newList = copy(aList)
* **newList = aList.copy()**
* newList.copy(aList)
* **newList = list(aList)**

**Explanation**:

The copy() method and list() constructor can be used to create a copy of a list. This will create a new list and any changes made in the original list will not reflect in the new list. This is **shallow copying**.

5. What is the output of the following code

aList = ["PYnative", [4, 8, 12, 16]]

print(aList[0][1])

print(aList[1][3])

* P 8  
  Y 16
* P  
  12
* **Y  
  16**

6. What is the output of the following

l = [None] \* 10

print(len(l))

* **10**
* 0
* Syntax Error

7. What is the output of the following

aList = [5, 10, 15, 25]

print(aList[::-2])

* [15, 10, 5]
* [10, 5]
* **[25, 10]**

**Explanation**:

aList[::-2] Start from the end of the list with step value 2.

8. What is the output of the following list assignment

aList = [4, 8, 12, 16]

aList[1:4] = [20, 24, 28]

print(aList)

* [4, 20, 24, 28, 8, 12, 16]
* **[4, 20, 24, 28]**

**Explanation**:

Use the assignment operator (=) to replace an item or a range of items in a List.

9. What is the output of the following code

list1 = ['xyz', 'zara', 'PYnative']

print (max(list1))

* PYnative
* **zara**

10. Select all the correct options to join two lists in Python

listOne = ['a', 'b', 'c', 'd']

listTwo = ['e', 'f', 'g']

* **newList = listOne + listTwo**
* newList = extend(listOne, listTwo)
* **newList = listOne.extend(listTwo)**
* newList.extend(listOne, listTwo)

**Explanation:**

1. The extend() method adds all the elements of an iterable (list, tuple, string) to the end of the list.
2. You can also use the addition operator to join two list in Python

11. What is the output of the following

aList = [1, 2, 3, 4, 5, 6, 7]

pow2 = [2 \* x for x in aList]

print(pow2)

* **[2, 4, 6, 8, 10, 12, 14]**
* [2, 4, 8, 16, 32, 64, 128]

**Explanation:**

Here we used list comprehension to multiply each item of a list by 2.

12. What is the output of the following code

my\_list = ["Hello", "Python"]

print("-".join(my\_list))

* HelloPython-
* **Hello-Python**
* -HelloPython

**Explanation**:

The join() method will join all items in a list into a string, using a hyphen character as a separator.

13. What is the output of the following code?

sampleList = [10, 20, 30, 40]

del sampleList[0:6]

print(sampleList)

* **[]**
* list index out of range.
* [10, 20]

14. What is the output of the following list function?

sampleList = [10, 20, 30, 40, 50]

sampleList.pop()

print(sampleList)

sampleList.pop(2)

print(sampleList)

* [20, 30, 40, 50]  
  [10, 20, 40]
* [10, 20, 30, 40]  
  [10, 20, 30, 50]
* **[10, 20, 30, 40]  
  [10, 20, 40]**

**Explanation:**

The list’s pop() function is used to remove the item present at the specified index, (or the last item if the index is not specified).

15. What is the output of the following list operation

sampleList = [10, 20, 30, 40, 50]

print(sampleList[-2])

print(sampleList[-4:-1])

* **40  
  [20, 30, 40]**
* IndexError: list index out of range

**Explanation:**

Use the range of negative indexes to search from the end of the list.

16. What is the output of the following list comprehension

resList = [x+y for x in ['Hello ', 'Good '] for y in ['Dear', 'Bye']]

print(resList)

* **[‘Hello Dear’, ‘Hello Bye’, ‘Good Dear’, ‘Good Bye’]**
* [‘Hello Dear’, ‘Good Dear’, ‘Hello Bye’, ‘Good Bye’]

**Exercise 1: Reverse a list in Python**

**Given**:

list1 = [100, 200, 300, 400, 500]

**Expected output:**

[500, 400, 300, 200, 100]

**Solution 1**: list function reverse()

list1 = [100, 200, 300, 400, 500]  
list1.reverse()  
print(list1)

**Solution 2**: Using negative slicing

**-1** indicates to start from the last item.

list1 = [100, 200, 300, 400, 500]  
list1 = list1[::-1]  
print(list1)

### Exercise 2: Concatenate two lists index-wise

Write a program to add two lists index-wise. Create a new list that contains the 0th index item from both the list, then the 1st index item, and so on till the last element. any leftover items will get added at the end of the new list.

**Given**:

list1 = ["M", "na", "i", "Ke"]

list2 = ["y", "me", "s", "lly"]

**Expected output:**

['My', 'name', 'is', 'Kelly']

Use the zip() function. This function takes two or more iterables (like list, dict, string), aggregates them in a tuple, and returns it.

list1 = ["M", "na", "i", "Ke"]   
list2 = ["y", "me", "s", "lly"]  
list3 = [i + j for i, j in zip(list1, list2)]  
print(list3)

### Exercise 3: Turn every item of a list into its square

Given a list of numbers. write a program to turn every item of a list into its square.

**Given**:

numbers = [1, 2, 3, 4, 5, 6, 7]

**Expected output:**

[1, 4, 9, 16, 25, 36, 49]

**Solution 1**: Using loop and list method

* Create an empty result list
* Iterate a numbers list using a loop
* In each iteration, calculate the square of a current number and add it to the result list using the append() method.
* numbers = [1, 2, 3, 4, 5, 6, 7]  
  *# result list*res = []  
  for i in numbers:  
   *# calculate square and add to the result list* res.append(i \* i)  
  print(res)

**Solution 2**: Use list comprehension

numbers = [1, 2, 3, 4, 5, 6, 7]  
res = [x \* x for x in numbers]  
print(res)

### Exercise 4: Concatenate two lists in the following order

list1 = ["Hello ", "take "]

list2 = ["Dear", "Sir"]

**Expected output:**

['Hello Dear', 'Hello Sir', 'take Dear', 'take Sir']

list1 = ["Hello ", "take "]  
list2 = ["Dear", "Sir"]  
  
res = [x + y for x in list1 for y in list2]  
print(res)

### Exercise 5: Iterate both lists simultaneously

Given a two Python list. Write a program to iterate both lists simultaneously and display items from list1 in original order and items from list2 in reverse order.

**Given**

list1 = [10, 20, 30, 40]

list2 = [100, 200, 300, 400]

**Expected output:**

10 400

20 300

30 200

40 100

* The zip() function can take two or more lists, aggregate them in a tuple, and returns it.
* Pass the first argument as a list1 and seconds argument as a list2[::-1] (reverse list using list slicing)
* Iterate the result using a for loop

list1 = [10, 20, 30, 40]  
list2 = [100, 200, 300, 400]  
  
for x, y in zip(list1, list2[::-1]):  
 print(x, y)

### Exercise 6: Remove empty strings from the list of strings

list1 = ["Mike", "", "Emma", "Kelly", "", "Brad"]

**Expected output:**

["Mike", "Emma", "Kelly", "Brad"]

Use a filter() function to remove None type from the list

list1 = ["Mike", "", "Emma", "Kelly", "", "Brad"]  
  
*# remove None from list1 and convert result into list*res = list(filter(None, list1))  
print(res)

### Exercise 7: Add new item to list after a specified item

Write a program to add item 7000 after 6000 in the following Python List

**Given**:

list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]

**Expected output:**

[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]

Use the append() method

list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]  
  
*# understand indexing  
# list1[0] = 10  
# list1[1] = 20  
# list1[2] = [300, 400, [5000, 6000], 500]  
# list1[2][2] = [5000, 6000]  
# list1[2][2][1] = 6000  
  
# solution*list1[2][2].append(7000)  
print(list1)

### Exercise 8: Extend nested list by adding the sublist

You have given a nested list. Write a program to extend it by adding the sublist ["h", "i", "j"] in such a way that it will look like the following list.

**Given List**:

list1 = ["a", "b", ["c", ["d", "e", ["f", "g"], "k"], "l"], "m", "n"]

# sub list to add

sub\_list = ["h", "i", "j"]

**Expected Output:**

['a', 'b', ['c', ['d', 'e', ['f', 'g', 'h', 'i', 'j'], 'k'], 'l'], 'm', 'n']

list1 = ["a", "b", ["c", ["d", "e", ["f", "g"], "k"], "l"], "m", "n"]  
sub\_list = ["h", "i", "j"]  
  
*# understand indexing  
# list1[2] = ['c', ['d', 'e', ['f', 'g'], 'k'], 'l']  
# list1[2][1] = ['d', 'e', ['f', 'g'], 'k']  
# list1[2][1][2] = ['f', 'g']  
  
# solution*list1[2][1][2].extend(sub\_list)  
print(list1)

### Exercise 9: Replace list’s item with new value if found

You have given a Python list. Write a program to find value 20 in the list, and if it is present, replace it with 200. Only update the first occurrence of an item.

**Given**:

list1 = [5, 10, 15, 20, 25, 50, 20]

**Expected output:**

[5, 10, 15, 200, 25, 50, 20]

list1 = [5, 10, 15, 20, 25, 50, 20]  
  
*# get the first occurrence index*index = list1.index(20)  
  
*# update item present at location*list1[index] = 200  
print(list1)

### Exercise 10: Remove all occurrences of a specific item from a list.

Given a Python list, write a program to remove all occurrences of item 20.

**Given**:

list1 = [5, 20, 15, 20, 25, 50, 20]

**Expected output:**

[5, 15, 25, 50]

list1 = [5, 20, 15, 20, 25, 50, 20]  
  
while 20 in list1:  
 list1.remove(20)  
print(list1)