```
1. What exactly is []?
In pyhton [] this is called as a list container, if we try to print type of [] we will
get type of list.
```

```

```

2. In a list of values stored in a variable called spam, how would you assign the value 'hello' as the third value? (Assume [2, 4, 6, 8, 10] are in spam.)

In [4]:

```
spam = [2,4,6,8,10]
# store "hello" at third position
spam.insert(2, "hello") # As in python the starting position (indexing) started from 0
print(spam)
```

```
[2, 4, 'hello', 6, 8, 10]
```

In [ ]:

```

```

Let's pretend the spam includes the list ['a','b','c'] for the next three queries.

3. What is the value of spam[int(int('39'* 2) / 11)]?

In [13]:

```
spam = ['a','b','c','d']
spam[int(int('39'*2)/11)]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-13-2129536216f5> in <module>
      1 spam = ['a','b','c','d']
----> 2 spam[int(int('39'*2)/11)]

IndexError: list index out of range
```

This gives error because we try to extract element whose index is not present.

4. What is the value of spam[-1]?

In [14]:

```
spam[-1]
#This will return last element from list means rightmost element in list, because if we tra
#the index will start from -1
```

Out[14]:

```
'd'
```

5. What is the value of spam[:2]?

In [15]:

```
spam[:2]
#This will give two elmemts from the list starting from 0 to 1 as second parameter/index is
```

Out[15]:

```
['a', 'b']
```

Let's pretend bacon has the list [3.14,'cat', '11','cat', True] for the next three questions.

6. What is the value of bacon.index('cat')?

In [19]:

```
bacon = [3.14,'cat','11','cat',True]
bacon.index('cat') # here we got output as 1 because we are checking that the "cat" element
                # can as we can see in bacon we have two 'cat' elements but this index func
```

Out[19]:

```
1
```

7. How does bacon.append(99) change the look of the list value in bacon?

In [20]:

```
bacon.append(99)
bacon  # Append operation will add element in the list at the last index
```

Out[20]:

```
[3.14, 'cat', '11', 'cat', True, 99]
```

8. How does bacon.remove('cat') change the look of the list in bacon?

In [21]:

```
bacon.remove("cat")
bacon                  # The remove operation will remove/ delete first occurrence of eleme
```

Out[21]:

```
[3.14, '11', 'cat', True, 99]
```

```

```

9. What are the list concatenation and list replication operators?

In [25]:

```
# List concationation means we can add one list to another list by usin + operator, or we c
# example
list1 = [1,2,3,4]
list2 = [4,5,6,6]
list3 = list1 + list2 # adding a list1 and list2 using + operator
list4 = list1*3 #Multiplying list1 and list2
print(list1)
print(list2)
print(list3)
print(list4)
```

```
[1, 2, 3, 4]
[4, 5, 6, 6]
[1, 2, 3, 4, 4, 5, 6, 6]
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
```

10. What is difference between the list methods append() and insert()?

```
the very basic difference between append() operation and insert() operations is the
append() operation will add/append the element at last index of list everytime
while by using insert() operation we can add elemnet at specific index of the list
```

11. What are the two methods for removing items from a list?

```
In python we have pop() method and remove() method to delete element from the list
pop() operation will remove the last element of list
whie remove() operation will remove the element at specific index of list
```

12. Describe how list values and string values are identical.

```
String and List are both iterabale objects i.e. we can iterate both objects on loops.
```

In [3]:

```
#Example 12
print("list Object")
fruits = ["mango","banana","strawberry","apple"]
for fruit in fruits:
    print(fruit)
print()

print("String Object")
name = "Yakub"
for character in name:
    print(character)
```

```
list Object
mango
banana
strawberry
apple

String Object
Y
a
k
u
b
```

13. What's the difference between tuples and lists?

```
The very basic difference between tupelse and list is that
list is mutable object, while the tuples object is immutable.

Let's try understand with the example.
```

In [8]:

```
#Example 13 List
# We are abale to manipulate the elements in the list, we can abale to add, delete
#Elements from the list

vehicles = ["ferrari","BMW","Mercedz-Benz", "TATA","Hyundai"]
vehicles.append("Maruti")
print("List after adding new element")
print(vehicles)
print()
vehicles.pop()
print("List after pop operation")
print(vehicles)
```

```
List after adding new element
['ferrari', 'BMW', 'Mercedz-Benz', 'TATA', 'Hyundai', 'Maruti']

List after pop operation
['ferrari', 'BMW', 'Mercedz-Benz', 'TATA', 'Hyundai']
```

In [10]:

```
#Example 13 Tuple
# As we know once tuple is defines we will not abale to perform operation to modify the tup

trees = ("Banyan","Coconut-Tree","Mango-Tree","Neem")
print("tuples before append operation")
print(trees)
print()
print("tuple after append operation")
trees.append("Tree") # We are getting error because tuple is immutable object
```

```
tuples before append operation
('Banyan', 'Coconut-Tree', 'Mango-Tree', 'Neem')

tuple after append operation


---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_16008/2630068309.py in <module>
      7 print()
      8 print("tuple after append operation")
----> 9 trees.append("Tree")

AttributeError: 'tuple' object has no attribute 'append'
```

14. How do you type a tuple value that only contains the integer 42?

In [11]:

```
num = (42)
```

15. How do you get a list value's tuple form? How do you get a tuple value's list form?

In [14]:

```
#Solution 15 A
nums = tuple([1,2,34,567,7])
type(nums)
```

Out[14]:

```
tuple
```

In [15]:

```
#Solution 15 B
nums = list((3,4,5,6,7))
type(nums)
```

Out[15]:

```
list
```

16. Variables that "contain" list values are not necessarily lists themselves. Instead, what do they contain?

```
List can contain not only list objects it may contain, list in list, list in tuple, tuple in list, list with dictionary in it.
list1 = [1,24,4,[4,4,6,6],"name","world",("a","b",3,4,5),{"name" : "john", 1 : 549}]
```

17. How do you distinguish between copy.copy() and copy.deepcopy()?

```
copy.copy() method will affect the original list or objects if we perform any operation on it
copy.deepcopy() method will assign new object which reference to the original object so it will change original object.
```

In [ ]:

```
```