ASSIGNMENT - 04

QUESTION : 01
What exactly is []?

ANSWER :
This is a empty list [] .

QUESTION -02 :
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.)

ANSWER :

In [4]:
```
spam = [2,4,6,8,10]
spam[2] = 'hello'    # here I have added the hello as our third value in the l
print (spam)
```

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

QUESTION - 03 :
spam = ['a','b','c','d']

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

ANSWER :

In [6]:
```
spam = ['a', 'b','c','d']
spam[int(int('3' * 2) / 11)] # spam[int(33/11)] = spam[3]
```

Out[6]:
```
'd'
```

HERE THE STRING 3 IS MULTIPLY BY 2 THERE RESULT IS '3'*'3' i.e. 33/11 the result is 3 so d is present in 3rd position thatswhy the output is d

QUESTION - 04 :

---> What is the value of spam[-1] ?

ANSWER :

In [9]:
```
spam = ['a', 'b','c','d']
spam[-1]
```

Out[9]:
```
'd'
```

ANSWER :
'd' is stored in spam [-1]

In [11]:
```
spam = ['a', 'b','c','d']
spam[:2]
```

QUESTION -05 :
What is the value of spam[:2]?

Out[11]:
```
['a', 'b']
```

here the range starts with 0 and stops at 2

QUESTION - 06 :

ANSWER :
'd' is stored in spam [-1]

In [11]:
```
spam = ['a', 'b','c','d']
spam[ :2]
```

QUESTION -05 :
What is the value of spam[:2]?

Out[11]: ['a', 'b']

here the range starts with 0 and stops at 2

QUESTION -06 :
bacon = [3.14, 'cat,' 11, 'cat,' True]

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

ANSWER :

In [13]:
```
bacon = [3.14, 'cat',11, 'cat',True]
bacon.index('cat')
```

Out[13]: 1

here it takes first occurence of cat i.e. 1

QUESTION -07 :

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

ANSWER :

In [14]:
```
bacon = [3.14, 'cat',11, 'cat',True]
bacon.append(99)
```

In [15]:
```
bacon
```

Out[15]: [3.14, 'cat', 11, 'cat', True, 99]

as append function adds the element at last in the list the 99 is the last element of the list

QUESTION -08 :

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

ANSWER :

In [16]:
```
bacon = [3.14, 'cat',11, 'cat',True]
bacon.remove('cat')
What are the list concatenation and list replication operators?
print(bacon)
```

ANSWER :
[3.14, 11, 'cat', True]

the cat is removed from the list as remove is the function of the list

In [18]:
```
# THE LIST CONCATENATION MEANS ADDING TWO LIST
#HERE IS THE EXAMPLE OF THE CONCATENATION OF LIST
```

QUESTION -09 :

```
list_1 = [ 1,3,5,6]
list_2 = [3,7,5,3]
```

```
What are the list concatenation and list replication operators?
print(bacon)
```

```
ANSWER :
[3.14, 11, 'cat', True]
```

```
the cat is removed from the list as remove is the function of the list
```

In [18]:
```
# THE LIST CONCATENATION MEANS ADDING TWO LIST
#HERE IS THE EXAMPLE OF THE CONCATENATION OF LIST
QUESTION -09 :
list_1 = [ 1,3,5,6]
list_2 = [3,7,5,3]

result = list_1 + list_2

print("THE CONCATENATION OF TWO LIST IS ----> ", result)
```

```
THE CONCATENATION OF TWO LIST IS ---->  [1, 3, 5, 6, 3, 7, 5, 3]
```

In [20]:
```
# THE REPLICATION OF LIST MEANS REPEATING LIST AT THE GIVEN NUMBER OF TIMES
# HERE IS THE EXAMPLE OF THE REPLICATION OF LIST

list_3 = [4,8,9,5]

list_3 * 4           # ----> repeating list 4 times
```

Out[20]:
```
[4, 8, 9, 5, 4, 8, 9, 5, 4, 8, 9, 5, 4, 8, 9, 5]
```

```
QUESTION -10 :
What is difference between the list methods append() and insert()?
```

```
ANSWER :
1. The append() function adds element at the end of the list
2. The insert() function inserts the element at particular index we define
while inserting the element


---------->  EXAMPLE :
```

In [23]:
```
#THE APPEND FUNCTION

list = [4,7,4,2,1]
list.append(10)
print(list)  # here it adds the element at the end of the list
```

```
[4, 7, 4, 2, 1, 10]
```

In [26]:
```
list = [4,7,4,2,1]
list.insert(4,'true')
print(list)  # here the true is entered in the 4th position of our list
```

```
[4, 7, 4, 2, 'true', 1]
```

```
QUESTION -11 :
What are the two methods for removing items from a list?
```

In [40]:
```
# pop() removes an element from the given position in the list and return it
We can remove item from a list using
1. pop() function
2. remove() function
l1 = [2,4,7,5,3,2]
list = l1.pop(2) # here the 2nd index position element i.e. 7 will be removed
print(list)
print("original list --> ",l1)
```

```
ANSWER :
7
original list -->  [2, 4, 5, 3, 2]
```

In [41]:
```
# remove() function remove the first occurence of the given item from the list
```

In [40]:
```
ANSWER :
# pop() :removes an element from the given position in the list and return it
We can remove item from a list using
1 pop() function
2 remove() function
l1 = [2,4,7,5,3,2]
list = l1.pop(2) # here the 2nd index position element i.e. 7 will be removed
print(list)
print("> EXAMPLE
print("original list --> ",l1)
```

```
7
original list -->  [2, 4, 5, 3, 2]
```

In [41]:
```
# remove() function remove the first occurence of the given item from the list

l2 = [2,5,7,8,3,2,3]
l3 = l2.remove(3)
print(l3)
print("original list --> ",l2)
```

```
None
original list -->  [2, 5, 7, 8, 2, 3]
```

In [42]:
```
 # here the first occurence of the 3 is removed from the list ie in the 5th pc
```

```
QUESTION -12 :
Describe how list values and string values are identical.
```

```
ANSWER :
1. Both lists and strings have indexes and slices.
2. Both lists and strings can be concatenated and replicated.
3. We can use for loop in both lists and strings.
4. in and not in operator are used in both lists and strings.
```

```
QUESTION -13 :
What's the difference between tuples and lists?
```

```
ANSWER :
     TUPLES                              LISTS
--> Tuples are immutable.          --> Lists are mutable.
--> Tuples are represented with    --> Lists are represented with the
    the round bracket or paren-        square brackets [].
    thisis ().
--> we cannot modify the tuple.    --> we can modify the list.
--> tuples have the operations     --> list also have the operations like
    like concatenation, repl-          concatenation, replication, slicing.
    ication,slicing.
```

```
QUESTION -14 :
How do you type a tuple value that only contains the integer 42?
```

```
ANSWER :
```

In [43]:
```
tuple = (42, )
```

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

In [44]:
```
tuple
```

Out[44]:
```
(42,)
```

```
ANSWER :
```

In [3]:
```
QUESTION -15 :
#conversion from list to tuple
list4 = [4,6,5]
l = tuple(list4)
print(l)
```

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

In [44]:
```
tuple
```

Out[44]:
```
(42,5)
```

ANSWER :

QUESTION -15 :

In [3]:
```
#conversion from list to tuple
list4 = [4,6,5]
l = tuple(list4)
print(l)
```

```
(4, 6, 5)
```

In [5]:
```
#conversion from tuple to list
tuple2 = (6,7,6,4,3)
t1 = list(tuple2)
print(t1)
```

```
[6, 7, 6, 4, 3]
```

```
QUESTION -16 :

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

```
ANSWER :
They contain references to list values
```

```
QUESTION -17 :
How do you distinguish between copy.copy() and copy.deepcopy()?
```

```
ANSWER :
The copy.copy() function will do a shallow copy of a list,
The copy.deepcopy() function will do a deep copy of a list. only
copy.deepcopy() will duplicate any lists inside the list
```

In [9]:
```
list6 = [3,5,3,2,6]
list = list6.copy()
list
```

Out[9]:
```
[3, 5, 3, 2, 6]
```

In [ ]:
```

```