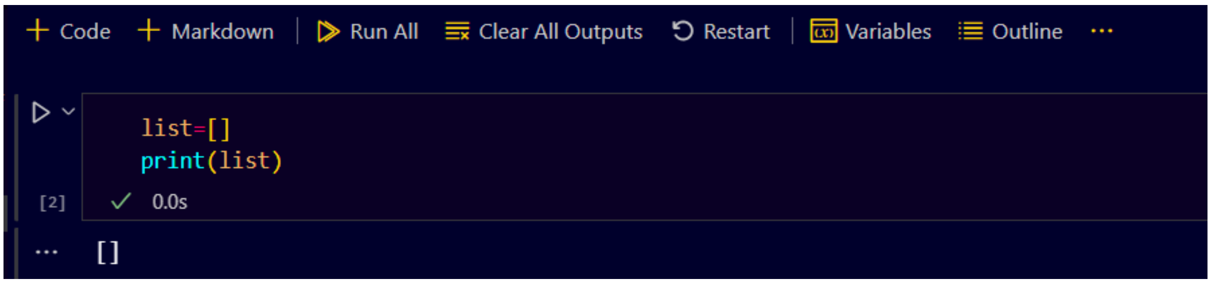
1. What exactly is []?

Solution:- This is an list which is empty



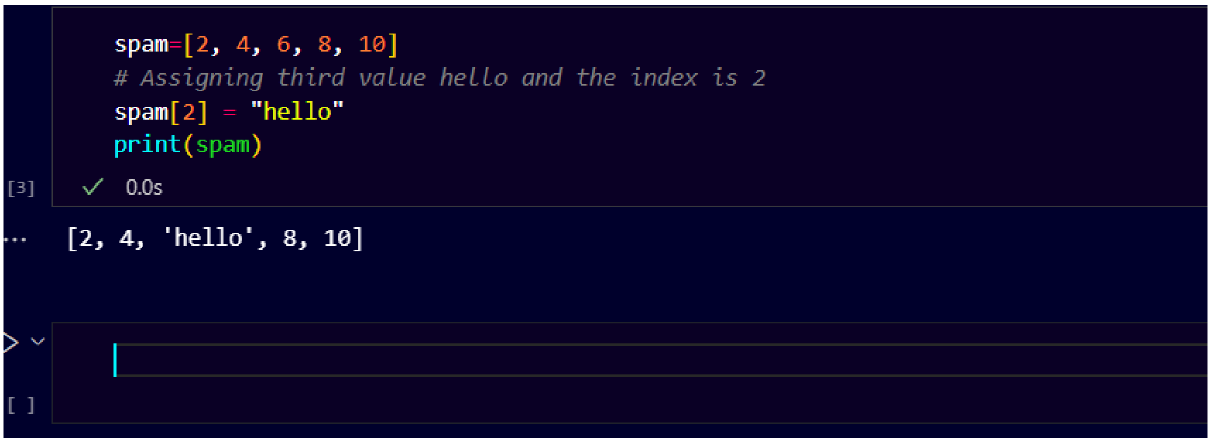
+ Code + Markdown | Run All Clear All Outputs Restart | Variables Outline ...

```
list=[]
print(list)
```

```
[2]  ✓ 0.0s
```

```
[]
```

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.)
   Solution:-



```
spam=[2, 4, 6, 8, 10]
# Assigning third value hello and the index is 2
spam[2] = "hello"
print(spam)
```

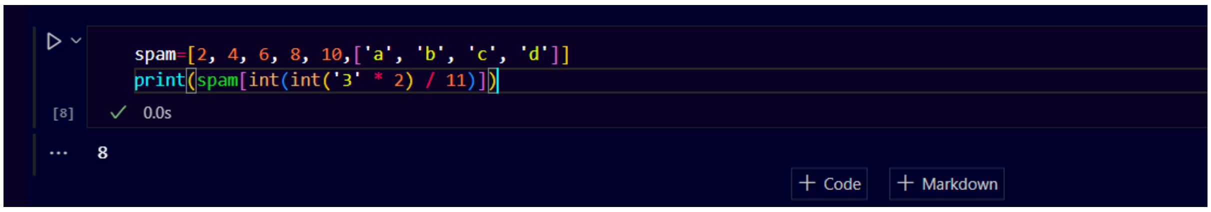
```
[3]  ✓ 0.0s
```

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

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

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

Solution:-



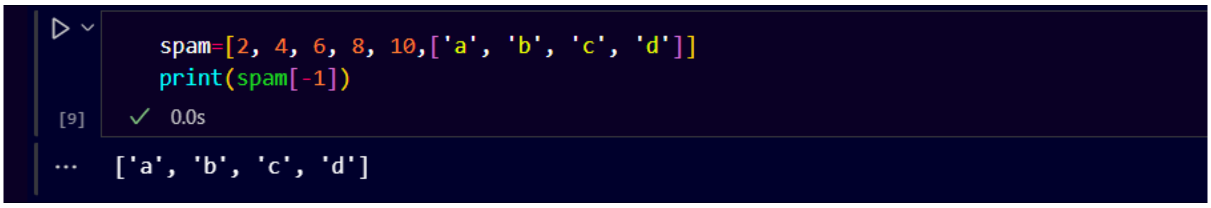
```
spam=[2, 4, 6, 8, 10,['a', 'b', 'c', 'd']]
print(spam[int(int('3' * 2) / 11)])
```

```
[8]  ✓ 0.0s
```

```
8
```

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

Solution:-



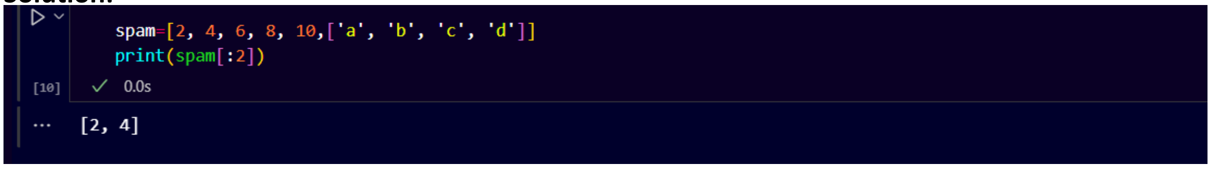
```
spam=[2, 4, 6, 8, 10,['a', 'b', 'c', 'd']]
print(spam[-1])
```

```
[9]  ✓ 0.0s
```

```
['a', 'b', 'c', 'd']
```

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

Solution:-



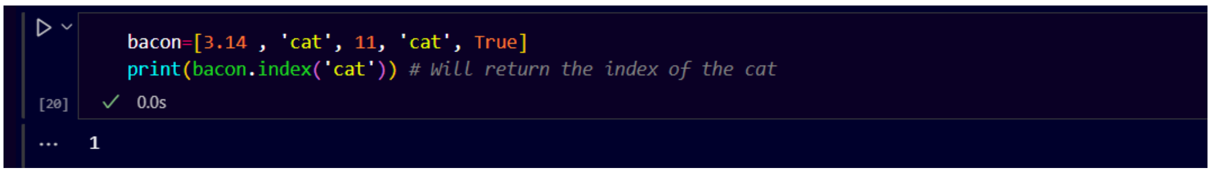
```python
spam=[2, 4, 6, 8, 10,['a', 'b', 'c', 'd']]
print(spam[:2])
```

```
[2, 4]
```

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')?

Solution:-

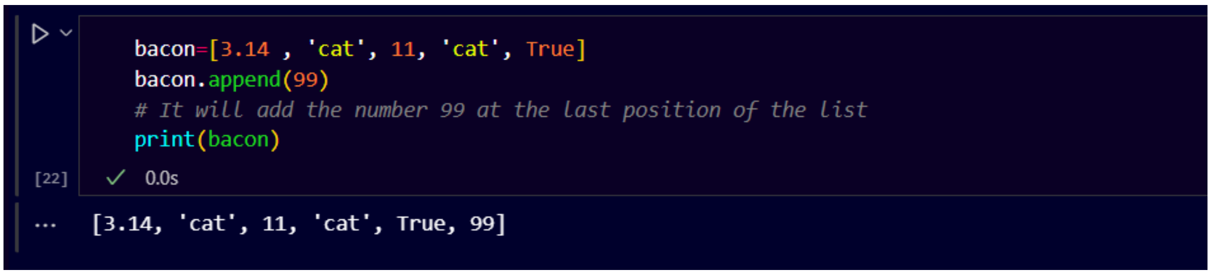


```python
bacon=[3.14 , 'cat', 11, 'cat', True]
print(bacon.index('cat')) # Will return the index of the cat
```

```
1
```

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

Solution:-

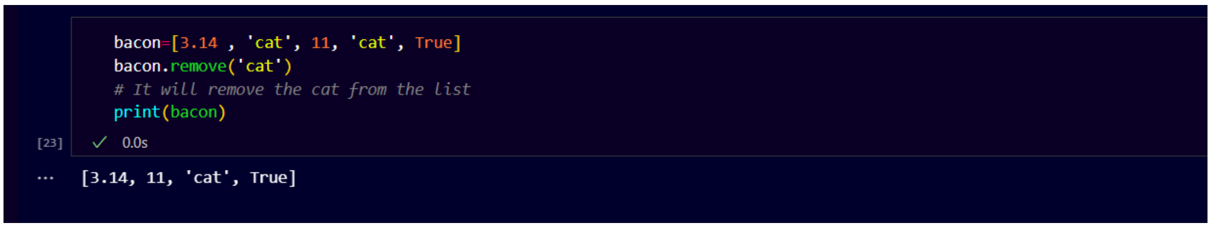


```python
bacon=[3.14 , 'cat', 11, 'cat', True]
bacon.append(99)
# It will add the number 99 at the last position of the list
print(bacon)
```

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

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

Solution:- It will remove the first 'cat' from the list

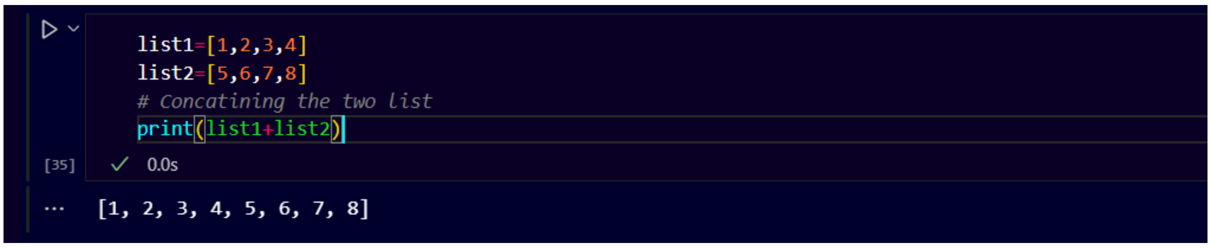


```python
bacon=[3.14 , 'cat', 11, 'cat', True]
bacon.remove('cat')
# It will remove the cat from the list
print(bacon)
```

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

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

Solution:- Merging of the two list is known as the list concatenation

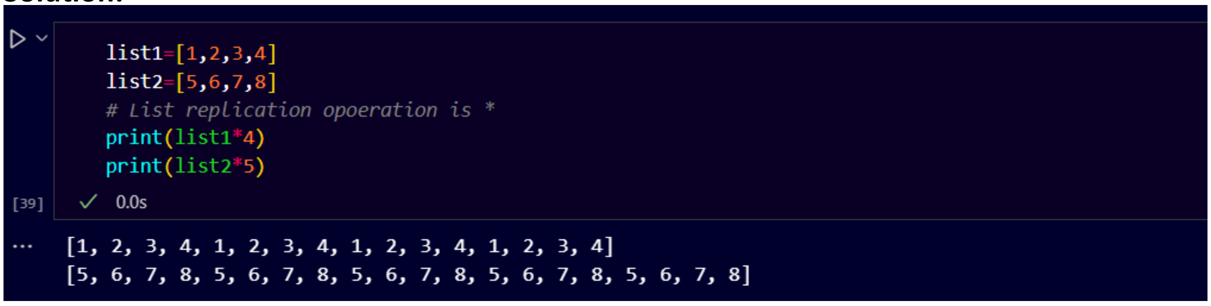


```python
list1=[1,2,3,4]
list2=[5,6,7,8]
# Concating the two list
print(list1+list2)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

List replication operation is *

Solution:-



```
list1=[1,2,3,4]
list2=[5,6,7,8]
# List replication opoeration is *
print(list1*4)
print(list2*5)
```

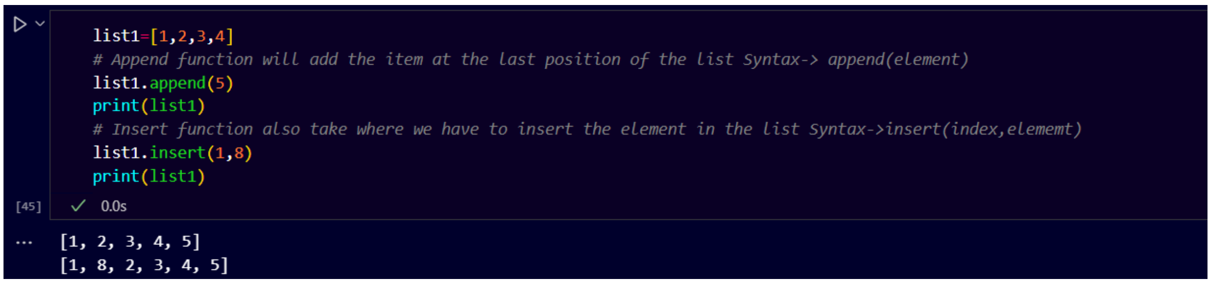
```
[39]  ✓ 0.0s
```

```
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
[5, 6, 7, 8, 5, 6, 7, 8, 5, 6, 7, 8, 5, 6, 7, 8, 5, 6, 7, 8]
```

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

Solution:-

```
# Append function will add the item at the last position of the list Syntax->
append(element)
# Insert function also take where we have to insert the element in the list
Syntax->insert(index,elememt)
```



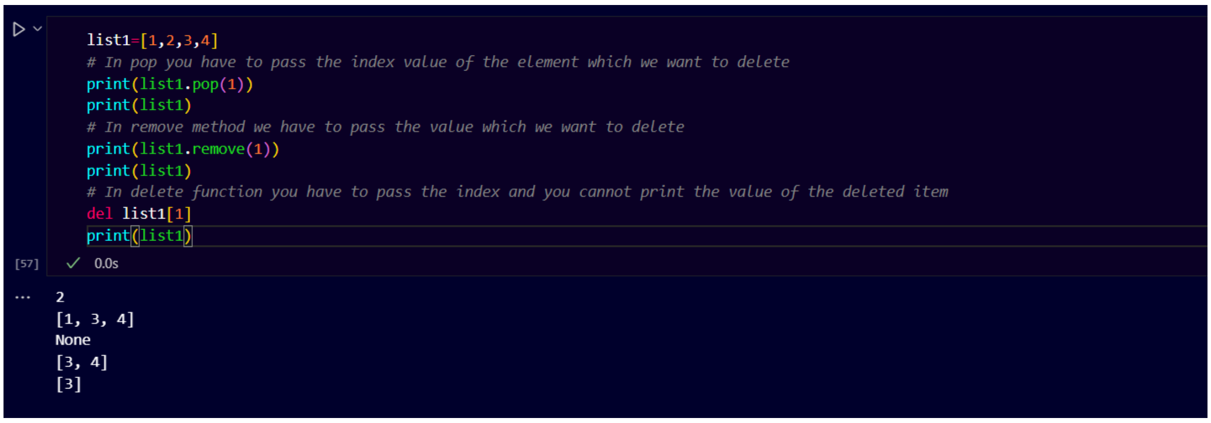
```
list1=[1,2,3,4]
# Append function will add the item at the last position of the list Syntax-> append(element)
list1.append(5)
print(list1)
# Insert function also take where we have to insert the element in the list Syntax->insert(index,elememt)
list1.insert(1,8)
print(list1)
```

```
[45]  ✓ 0.0s
```

```
[1, 2, 3, 4, 5]
[1, 8, 2, 3, 4, 5]
```

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

Solution:-



```
list1=[1,2,3,4]
# In pop you have to pass the index value of the element which we want to delete
print(list1.pop(1))
print(list1)
# In remove method we have to pass the value which we want to delete
print(list1.remove(1))
print(list1)
# In delete function you have to pass the index and you cannot print the value of the deleted item
del list1[1]
print(list1)
```

```
[57]  ✓ 0.0s
```

```
2
[1, 3, 4]
None
[3, 4]
[3]
```

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

Solution:-

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

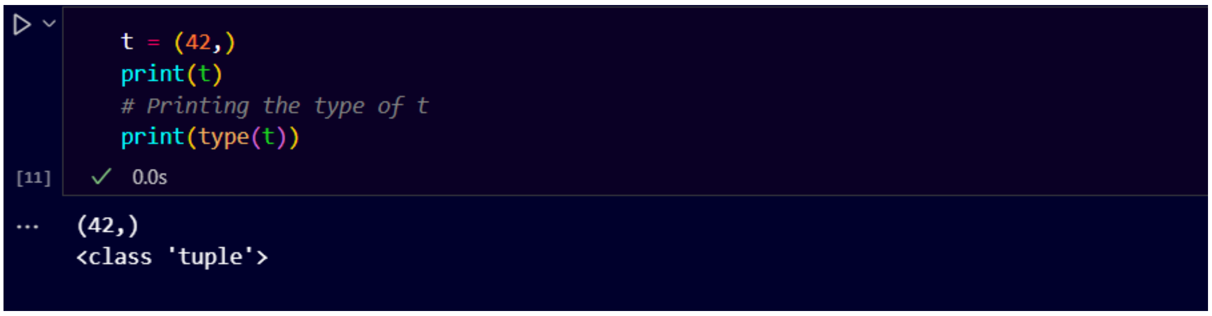
Solution: - 1-> Both are sequential in nature

2-> Both have length characterstics

3-> Both list and string have position

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

Solution:-



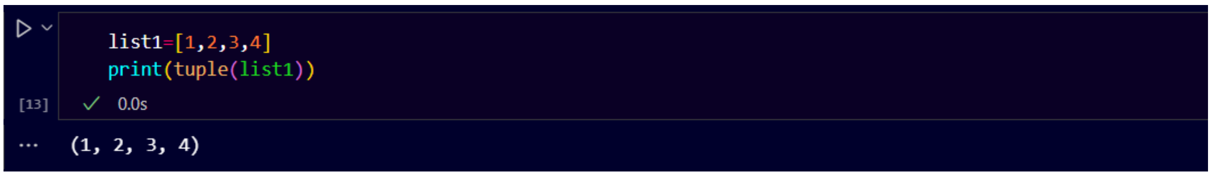
```
t = (42,)
print(t)
# Printing the type of t
print(type(t))
```

```
[11] ✓ 0.0s
```

```
(42,)
<class 'tuple'>
```

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

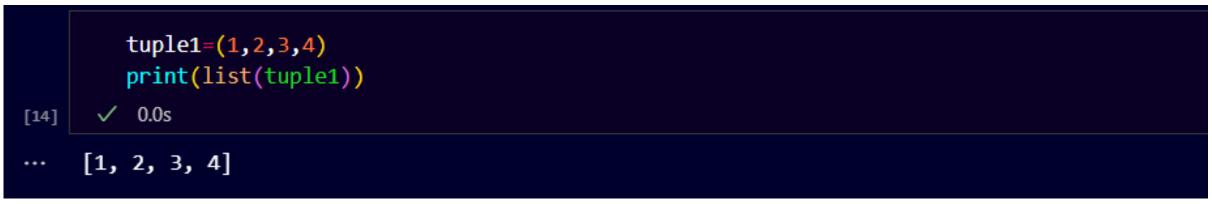
Solution:-



```
list1=[1,2,3,4]
print(tuple(list1))
```

```
[13] ✓ 0.0s
```

```
(1, 2, 3, 4)
```



```
tuple1=(1,2,3,4)
print(list(tuple1))
```

```
[14] ✓ 0.0s
```

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

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

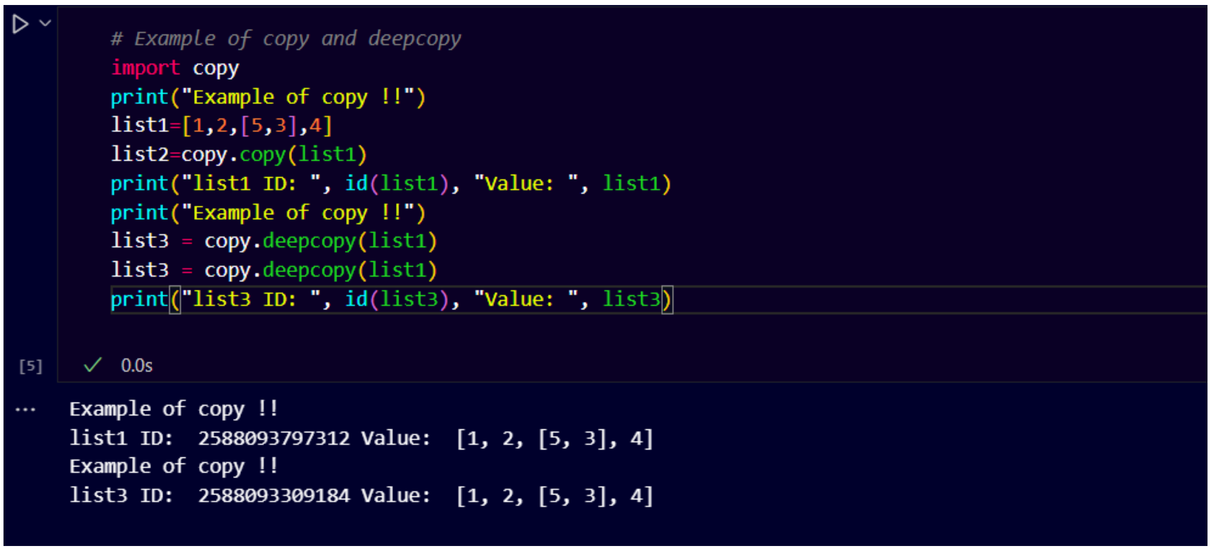
Solution:- Variable will contain the references to the list values rather than the list values

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

Solution:- The result of copy and deepcopy in python is same but there id's are different

DeepCopy:- In the case of deep copy, a copy of the object is copied into another object. It means that any changes made to a copy of the object do not reflect in the original object.

Copy:- In the case of shallow copy, a reference of an object is copied into another object. It means that any changes made to a copy of an object **do** reflect in the original object.



```python
# Example of copy and deepcopy
import copy
print("Example of copy !!")
list1=[1,2,[5,3],4]
list2=copy.copy(list1)
print("list1 ID: ", id(list1), "Value: ", list1)
print("Example of copy !!")
list3 = copy.deepcopy(list1)
list3 = copy.deepcopy(list1)
print("list3 ID: ", id(list3), "Value: ", list3)
```

[5] ✓ 0.0s

```
Example of copy !!
list1 ID:  2588093797312 Value:  [1, 2, [5, 3], 4]
Example of copy !!
list3 ID:  2588093309184 Value:  [1, 2, [5, 3], 4]
```