PYTHON ASSIGNMENT 4

Q1.What exactly is []?
Ans. [] is an empty list

Q2. 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 [5]:
```
spam = [2, 4, 6, 8, 10]
spam[2] = 'Hello'
spam
```

Out[5]:
```
[2, 4, 'Hello', 8, 10]
```

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

Q3. What is the value of spam[int(int('3' * 2) / 11)]?

In [7]:
```
spam = ['a', 'b', 'c', 'd']
spam[int(int('3' *2)/11)]  #spam[33/11]= spam[3]
```

Out[7]:
```
'd'
```

Q4. What is the value of spam[-1]?

In [8]:
```
spam[-1]
```

Out[8]:
```
'd'
```

Q5. What is the value of spam[:2]?

In [9]:
```
spam[:2]
```

Out[9]:
```
['a', 'b']
```

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

In [17]:
```
bacon = [3.14, 'cat', 11, 'cat', True]
```

Q6. What is the value of bacon.index('cat')?

In [11]:
```
bacon.index('cat')
```

Out[11]:
```
1
```

Q7. How does bacon.append(99) change the look of the list value in bacon?

In [18]:
```
bacon.append(99)
bacon  #append adds the value to the end of the list
```

Out[18]:
```
[3.14, 'cat', 11, 'cat', True, 99]
```

Q8. How does bacon.remove('cat') change the look of the list in bacon?

In [19]:
```
bacon.remove('cat')
bacon   #removes the first occurance of the item
```

Out[19]:
```
[3.14, 11, 'cat', True, 99]
```

Q9. What are the list concatenation and list replication operators?

The operator for list concatenation is +, while the operator for replication is *.

In [20]:
```
#CONTATENATION

a =[1,2,3]
b =[5,6,7]
a+b
```

Out[20]:
```
[1, 2, 3, 5, 6, 7]
```

In [22]:
```
#REPLICATION

a*2
```

Out[22]:
```
[1, 2, 3, 1, 2, 3]
```

Q10. What is difference between the list methods append() and insert()?

The difference between the two methods is that . append() adds an item to the end of a list, whereas . insert() inserts and item in a specified position in the list.

In [27]:
```
#APPEND

x = [1,2,3,4,5]
x.append(10)
x
```

Out[27]:
```
[1, 2, 3, 4, 5, 10]
```

In [28]:
```
#INSERT

x.insert(2, 'Greetings')
x
```

Out[28]:
```
[1, 2, 'Greetings', 3, 4, 5, 10]
```

Q11. What are the two methods for removing items from a list?

Ans. The two methods of removing items from a list are remove and delete

In [37]:
```
#REMOVE (Removes the first occurance of the item)

lst = ['Rose', 'Lily', 'Tulip', 'Orchid', 'sunflower', 'Rose']
lst.remove('Rose')
lst
```

Out[37]:
```
['Lily', 'Tulip', 'Orchid', 'sunflower', 'Rose']
```

In [38]:
```
#DELETE

lst = ['Rose', 'Lily', 'Tulip', 'Orchid', 'sunflower', 'Rose']
del lst[4]
lst
```

Out[38]:
```
['Rose', 'Lily', 'Tulip', 'Orchid', 'Rose']
```

Q12. Describe how list values and string values are identical.

Ans.
1.Both lists and strings can be passed to len()
2.Have indexes and slices
3.Can be used in for loops
4.Can be concatenated or replicated
5.Can be used with the in and not in operators

Q13.What's the difference between tuples and lists?
Ans.
Lists : are mutable - they can have values added, removed, or changed. lists use the square brackets, [and]
Tuples : are immutable; they cannot be changed at all. Tuples are written using parentheses, (and)

Q14. How do you type a tuple value that only contains the integer 42?

In [39]:
```
tuple =(42)
tuple
```

Out[39]:
```
42
```

Q15. How do you get a list value's tuple form? How do you get a tuple value's list form?

In [51]:
```
#TUPLE VALUES'S IN LIST FORM

i = (1,2,3,4,5)
i1 = list(i)
i1
```

Out[51]:
```
[1, 2, 3, 4, 5]
```

In [56]:
```
#LIST VALUES TO TUPLE FORM
```

```
j =[1,2,3,4,5]
j1 = tuple(j)
print(j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[56], line 4
      1 #LIST VALUES TO TUPLE FORM
      3 j =[1,2,3,4,5]
----> 4 j1 = tuple(j)
      5 print(j)

TypeError: 'int' object is not callable
```

Q16. Variables that 'contain' list values are not necessarily lists themselves. Instead, what do they contain?
Ans. They contain references to list values

Q17. How do you distinguish between copy.copy() and copy.deepcopy()?
Ans. 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