|  |  |
| --- | --- |
| Q: | 1. What is []?   This bracket indicate the list in Python |
| Q: | 1. How would you assign the value 'hello' as the third value in a list stored in a variable named spam? (Assume spam contains [2, 4, 6, 8, 10].)   Spam[2] = ‘hello’  For the following three questions, let’s say spam contains the list ['a', 'b', 'c', 'd']. |
| Q: | 1. What does spam[int(int('3' \* 2) // 11)] evaluate to?   Int(int(‘3’\*2) // 11) = 3 => spam[3] = d |
| Q: | 1. What does spam[-1] evaluate to?   d |
| Q: | 1. What does spam[:2] evaluate to?   [‘a’,’b’]  For the following three questions, let’s say bacon contains the list [3.14, 'cat', 11, 'cat', True]. |
| Q: | 1. What does bacon.index('cat') evaluate to?   1 |
| Q: | 1. What does bacon.append(99) make the list value in bacon look like?   [3.14, 'cat', 11, 'cat', True,99]. |
| Q: | 1. What does bacon.remove('cat') make the list value in bacon look like?   [3.14,11, 'cat', True,99] |
| Q: | 1. What are the operators for list concatenation and list replication?   + operator for list concatenation  \* operator for list replication |
| Q: | 1. What is the difference between the append() and insert() list methods?   append() method: add the new element to the end of the list  insert() method: insert the new element to specific index of the list |
| Q: | 1. What are two ways to remove values from a list?   del : remove the specific index of the list  remove(‘’value’): remove the value of the list if the list have many same value -> will remove the first one |
| Q: | 1. Name a few ways that list values are similar to string values.   Indexing, slicing , len or loop |
| Q: | 1. What is the difference between lists and tuples?   Lists are mutable that mean you could add, remove or modify the lists  Tuples are immutable that mean you could read only |
| Q: | 1. How do you type the tuple value that has just the integer value 42 in it?   tuple((42,)) |
| Q: | 1. How can you get the tuple form of a list value? How can you get the list form of a tuple value?   tuple([‘a’,’b’,’c’])  list ((‘a’,’b’,’c’)) |
| Q: | 1. Variables that “contain” list values don’t actually contain lists directly. What do they contain instead?   The reference of the list |
| Q: | 1. What is the difference between copy.copy() and copy.deepcopy()?   Copy.copy() will duplicate the value of the list  Copy.deepcopy() will duplicate not only the value but the container as well |
|  | Example:  Copy.copy()  import copy  old\_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  new\_list = copy.copy(old\_list)  old\_list[1][1] = 'ZZ'  print("Old list:", old\_list)  print("New list:", new\_list)  Out put  Old list: [[1, 2, 3], [4, 'ZZ', 9], [7, 8, 9]]  New list: [[1, 2, 3], [4, 'ZZ', 9], [7, 8, 9]]  copy.deepcopy()  import copy  old\_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  new\_list = copy.deepcopy(old\_list)  old\_list[1][0] = 'ZZ'  print("Old list:", old\_list)  print("New list:", new\_list)  Output  Old list: [[1, 2, 3], ['ZZ', 5, 6], [7, 8, 9]]  New list: [[1, 2, 3], [4, 5, 6], [7, 8, 9]] |