**Module 6: Discussion Forum**

The list and dictionary object types are two of the most important and often used types in a Python program. What are some ways to insert, update, and remove elements from lists and dictionaries? Why would you choose one data type over another? Provide code examples demonstrating the usages of both data types. Actively participate in this discussion by providing constructive feedback on the criteria, rationales, and examples posted by your peers. Include additional code examples in your responses if applicable.

**List** in python is very powerful data structure. This is used to preserve the insertion order of the items. Unlike array list can store multiple data types of values and can grow dynamically. Important thing is all the list functions update the original list.

Some of the useful functions list support are

append(v): Append value at the end of the list

insert(inx, v): add element to the specific index in the list

remove(v): it removes the first occurrence of the value from the list

extend([]): this function takes a list of elements as an argument and append to the end of the main list

pop(inx): removes the element at the specific and index and reorder the list

clear(): remove all the elements from the list.

Please find the below code snippet to explains the above functions and some special cases of these utility functions.

|  |
| --- |
| l = [1, "John", 100]        print(f"original: {l}")      #append      l.append(10)      print(f"append: {l}")        #insert      l.insert(2, "1111 1st Terr, US xxxx")      print(f"insert: {l}")        #insert at random index      # element will be added to the end of the list, if the index is greate than the size of the list      l.insert(20, "xxx 2st Terr, US xxxx")      print(f"insert at random index: {l}")        #extend      sub\_list = [35, "Male"]      l.extend(sub\_list)      print(f"extend: {l}")        #remove      l.remove(35)      print(f"remove: {l}")      ## when element not found, program will throw an exception      #l.remove(40)      #print(f"remove: {l}")          # pop      l.pop(2)      print(f"pop: {l}")        # clear      l.clear()      print(f"clear: {l}") |

**Output:**

A screen shot of a computer

AI-generated content may be incorrect.

**Dictionary** in python are used to store key value pair. Key and value can be of any type if they are hashable. This data structure is very helpful for faster lookup.

Some of the useful functions available when working with Dictionary in python are

get(key, [, default])

Items(): returns iterable key-value pair tuples.

keys(): returns iterable keys

values(): returns iterable values

setdefault(key, [,default]): if key present, it returns its value, if not present it insert the key and default value.

clear(): clear all the items in dictionary

Please find the below code snippet to explains the above functions and some special cases of these utility functions.

|  |
| --- |
| dic = {} # empty dictionary        # load values into dictionary      dic["key1"] = 10.5      dic[5] = "Val1"      dic["key2"] = "Hello"      print(f"initial dictionary: {dic}")        # get(key, [, default])      result = dic.get("key1")      print(f"get: {result}")        result = dic.get("key5", 20)      print(f"get2: {result}")      #items()      result = dic.items()      print(f"items: {result}")        #keys()      result = dic.keys()      print(f"keys: {result}")        #values()      result = dic.values()      print(f"values: {result}")        #setdefault(key, [,default])      result = dic.setdefault(5, 50)      print(f"setdefault: {result}")        #setdefault(key, [,default])      result = dic.setdefault(15, 50)      print(f"clear: {result}")        #clear()      result = dic.clear()      print(f"setdefault2: {result}") |

**Output:**

**A screen shot of a computer code

AI-generated content may be incorrect.**