|  |  |
| --- | --- |
| **code** | # GitHub repository link :- https://github.com/20CS055/CE259\_PYTHON.git  # Dictionary # a. Write a Python script to check whether a given key already exists in a dictionary.  student= {  'Id': '20cs055',  'Name': 'Mihir',  'Age': '19',  'gender': 'male',  'skills': ['Gaming', 'Dancing', 'c', 'c++', 'Python'] }  print('Name' in student) print( 'Address' in student)  # b. Write a Python script to merge two Python dictionaries.  student1= {  'Id': 'ab123',  'Name': 'abc',  'Age': '19'  } student2= {  'Id': '20cs055',  'Name': 'Mihir',  'Age': '19',  'gender': 'male',  'skills': ['Gaming', 'Dancing', 'c', 'c++', 'Python']  } student3=student1.copy() student3.update(student2) print(student3)  # c. Write a Python program to sum all the items in a dictionary.  Dict1= {  'a': 23,  'b': 12,  'c': 19,  'd': 55,  'e': 76  }  sum=0; for i in Dict1:  sum=sum+Dict1[i]  print(sum)  # d. Write a Python script to add a key to a dictionary.  #Sample Dictionary : {0: 10, 1: 20} #ExpectedResult : {0: 10, 1: 20, 2: 30}  dict={  0: 10,  1: 20  } dict.update({2: 30}) print(dict)  # Another example for above question  student\_= {  'Id': 'ab123',  'Name': 'abc',  'Age': '19'  }  print("before updating") print(student\_) student\_.update({'roll\_no':55}) print("after updating") print(student\_)   #e. Write a Python script to concatenate following dictionaries to create a new one. #Sample Dictionary : #dic1={1:10, 2:20} #dic2={3:30, 4:40} #dic3={5:50,6:60} #Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}  dict1={1:10, 2:20} dict2={3:30, 4:40} dict3={5:50,6:60} dict4 = {} for d in (dict1, dict2, dict3): dict4.update(d) print(dict4)  # Tuple  # a. Write a Python program to create a tuple with different data types.  tuple1 = ("Mihir",2.131,False,982) print(tuple1)  # b. Write a Python program to create a tuple with numbers and print one item.  #Creating a tuple with numbers tuple2 = 5, 10, 15, 20, 25 print(tuple2[0])  # c. Write a Python program to add an item in a tuple.  #create a tuple tuple3 = (4, 6, 2, 8, 3, 1) print(tuple3) #tuples are immutable, so you can not add new elements #using merge of tuples with the + operator you can add an element and it will create a new tuple tuple4 = tuple3 + (9,) print(tuple4) #adding items in a specific index tuple5 = tuple3[:5] + (15, 20, 25) print(tuple5) #converting the tuple to list list1 = list(tuple3) #use different ways to add items in list list1.append(30) tuple6 = tuple(list1) print(tuple6)  # d. Write a Python program to convert a tuple to a string.  tuple7 = ('M', 'I', 'H', 'I', 'R') str =''.join(tuple7) print(str)  # e. Write a Python program to find the length of a tuple.  #create a tuple tuple8 =(12,21,32,23,43,34,54,45) print(tuple8) #use the len() function to known the length of tuple print(len(tuple8))  # Set  # a. Write a Python program to add member(s) in a set and clear a set  #A new empty set Fruits\_basket = set() print(Fruits\_basket) print("\nAdding one element:") Fruits\_basket.add("apple") print(Fruits\_basket) print("\nAdding multiple items:") Fruits\_basket.update(["orange","mango","banana","grapes"]) print(Fruits\_basket) print("\nclearing the Fruits\_basket set") Fruits\_basket.clear() print(Fruits\_basket)  # b. Write a Python program to remove an item from a set if it is present in the set.  set1 = {"kishan", "prince", "preet", "jeet", "hit", "mit", "meet"} print(set1) name = input("insert a name to remove from a list") initial\_length=len(set1) for i in set1:  if name == i:  set1.discard(name)  break final\_length=len(set1) if initial\_length == final\_length:  print("item entered by you is not present inside the list") else :  print("item removed successfully from the set") print(set1)  # c. Write a Python program to create an intersection, Union, difference of sets.  # sets are define A = {21, 31, 43, 62, 82, 32, 46}; B = {89, 32, 46, 23, 31, 44, 53}; # intersection print("Intersection :", A & B) # union print("Union :", A | B) # difference print("Difference :", A - B)  # d. Write a Python program to find maximum and the minimum value in a set.  #Create a set set\_ex = {522, 103, 32, 1245, 2, 230} print("Original set elements:") print(set\_ex) print(type(set\_ex)) print("\nMaximum value of the said set:") print(max(set\_ex)) print("\nMinimum value of the said set:") print(min(set\_ex))  # e. Write a Python program to find the most common elements and their counts from list, tuple, dictionary.  from collections import Counter list\_1 = [14, 24, 45, 45, 94, 19] cntr = Counter(list\_1) #most\_common(1) returns top 1 most common element with its frequency. most\_common\_element,frequency = cntr.most\_common(1)[0] # Return the most common element and its frequency with most\_common print("The most common element is {}, and the frequency of that element is {}".format(most\_common\_element,frequency)) |
| **Output** | **Dictionary**  **# a** |
|  | **# b** |
|  | **# c** |
|  | **# d** |
|  | **# e** |
|  | **Tuple**  **# a** |
|  | **# b** |
|  | **# c** |
|  | **# d** |
|  | **# e** |
|  | **Set**  **# a** |
|  | **# b** |
|  | **# c** |
|  | **# d** |
|  | **# e** |