Data_Structures_in_python.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

## Topic : Dictionary Exercise

```
[4] #Q1. Create a dictionary with keys 'name', 'age', and 'address' and values 'John', 25, and 'New York' respectively.

    # Creating the dictionary
    person_dict = {'name': 'John', 'age': 25, 'address': 'New York'}
    print("Original dictionary:", person_dict)
```

```
Original dictionary: {'name': 'John', 'age': 25, 'address': 'New York'}
```

```
#Q2. Add a new key-value pair to the dictionary created in Q1 with key 'phone' and value '1234567890'.

# Adding a new key-value pair to the dictionary
person_dict['phone'] = '1234567890'
print("Updated dictionary:", person_dict)
```

```
Updated dictionary: {'name': 'John', 'age': 25, 'address': 'New York', 'phone': '1234567890'}
```

0s completed at 20:14

Data_Structures_in_python.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

## List Exercise

```
[1] #Q1. Create a list of 5 random numbers and print the list.

    import random

    # Creating a list of 5 random numbers
    random_list = [random.randint(1, 100) for i in range(5)]
    print("Original list:", random_list)
```

```
Original list: [54, 4, 41, 63, 71]
```

```
[2] #Q2. Insert 3 new values to the list and print the updated list.

    # Inserting 3 new values to the list
    random_list.extend([10, 20, 30])
    print("Updated list:", random_list)
```

```
Updated list: [54, 4, 41, 63, 71, 10, 20, 30]
```

```
#Q3. Try to use a for loop to print each element in the list.

# Using a for loop to print each element in the list
for element in random_list:
    print(element)
```

```
54
4
41
63
71
10
20
30
```

0s completed at 20:10

Data_Structures_in_python.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Gemini

## *Topic : Set Exercise*

```
[12] #Q1. Create a set with values 1, 2, 3, 4, and 5.

     # Creating the set
     num_set = {1, 2, 3, 4, 5}
     print("Original set:", num_set)
```

```
Original set: {1, 2, 3, 4, 5}
```

```
[13] #Q2. Add the value 6 to the set created in Q1.

     # Adding value 6 to the set
     num_set.add(6)
     print("Set after adding 6:", num_set)
```

```
Set after adding 6: {1, 2, 3, 4, 5, 6}
```

```
#Q3. Remove the value 3 from the set created in Q1.

# Removing value 3 from the set
num_set.discard(3)
print("Set after removing 3:", num_set)
```

```
Set after removing 3: {1, 2, 4, 5, 6}
```

0s completed at 20:27

Data_Structures_in_python.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Gemini

## *Topic : Tuple Exercise*

```
[15] #Q1. Create a tuple with values 1, 2, 3, and 4.

     # Creating the tuple
     num_tuple = (1, 2, 3, 4)
     print("Tuple:", num_tuple)
```

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

```
#Q2. Print the length of the tuple created in Q1.

# Printing the length of the tuple
print("Length of the tuple:", len(num_tuple))
```

```
Length of the tuple: 4
```

0s completed at 20:29