

DATA STRUCTURES IN PYTHON

TOPIC :LIST

Q1.

CREATE A LIST OF 5 RANDOM NUMBERS AND PRINT THE LIST.

```
# Create a list of 5 random numbers and print the list.  
Rand_Numbers=[1,5,7,9,11]  
print(Rand_Numbers)
```

```
[1, 5, 7, 9, 11]
```

Q2.

INSERT 3 NEW VALUES TO THE LIST AND PRINT THE UPDATED LIST.

```
# new 3 random numbers  
Rand_New_Numbers=[13,15,17]  
print(Rand_New_Numbers)
```

```
[13, 15, 17]
```

```
# add the elements of the Rand_New_Number to Rand_Numbers  
Rand_Numbers.extend(Rand_New_Numbers)  
print("updated List:",Rand_Numbers)
```

```
updated List: [1, 5, 7, 9, 11, 13, 15, 17]
```

```
# insert the new element to the specific position based on the index  
Rand_Numbers.insert(__index: 1, __object: 3)  
print("updated List:",Rand_Numbers)
```

```
updated List: [1, 3, 5, 7, 9, 11, 13, 15, 17]
```

Q3.

TRY TO USE A FOR LOOP TO PRINT EACH ELEMENT IN THE LIST.

```
# Generate a list of 5 random numbers using for loop
import random
random_numbers = [random.randint(a: 1, b: 100) for _ in range(5)]
print(random_numbers)
# Insert 3 new values using the for loop to the list and print the updated list.
new_values = [random.randint(a: 1, b: 100) for _ in range(3)]
print(new_values)
# add element from the new_values to the random_numbers
random_numbers.extend(new_values)
print("Updated List with extend:", random_numbers)
# insert new_values elements to the specific position of the random_numbers
random_numbers.insert(__index: 1, new_values)
print("Updated List with insert:", random_numbers)
print ( type(random_numbers))
```

```
[33, 9, 10, 6, 6]
[93, 44, 74]
Updated List with extend: [33, 9, 10, 6, 6, 93, 44, 74]
Updated List with insert: [33, [93, 44, 74], 9, 10, 6, 6, 93, 44, 74]
<class 'list'>
```

TOPIC: DICTIONARY

Q1.

CREATE A DICTIONARY WITH KEYS 'NAME', 'AGE', AND 'ADDRESS' AND VALUES 'JOHN', 25, AND 'NEW YORK' RESPECTIVELY.

```
# Create a dictionary with keys 'name', 'age', and 'address'
# and values 'John', 25, and 'New York' respectively.
student={"name":"John","age":"25","address":"New York"}
print(student)
```

```
{'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'.

```
student.update({"Phone NO":"1234567890"})
print(student)
```

```
{'name': 'John', 'age': '25', 'address': 'New York', 'Phone NO': '1234567890'}
```

TOPIC: SET

Q1.

CREATE A SET WITH VALUES 1, 2, 3, 4, AND 5.

```
# # Creating a set with values 1, 2, 3, 4, and 5
new_set = {1, 2, 3, 4, 5}
print("Set:", new_set)
```

```
Set: {1, 2, 3, 4, 5}
```

Q2.

ADD THE VALUE 6 TO THE SET CREATED IN Q1.

```
# Add the value 6 to the set created "new_set"
new_set.add(6)
print("New Set:", new_set)
```

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

Q3.

REMOVE THE VALUE 3 FROM THE SET CREATED IN Q1.

```
# Remove the value 3 from the set created "new_set"
new_set.remove(3)
print("updated set", new_set)
```

```
updated set {1, 2, 4, 5, 6}
```

TOPIC:TUPLE

Q1.

CREATE A TUPLE WITH VALUES 1, 2, 3, AND 4

```
# Create a tuple with values 1, 2, 3, and 4
tuple = (1, 2, 3, 4)
print("tuple",tuple)
```

```
tuple (1, 2, 3, 4)
```

Q2.

PRINT THE LENGTH OF THE TUPLE CREATED IN Q1.

```
# print the length of the tuple created
print(len(tuple))
```

```
4
```