Data Structure

October 21, 2024

[29]:
```
# List

# Q1

import random
random_numbers = [random.randint(1, 100) for _ in range(5)]
print("Random Numbers:", random_numbers)
```

```
Random Numbers: [66, 59, 33, 44, 50]
```

[31]:
```
# Q2

random_numbers.insert(0, 101)
random_numbers.append(102)
random_numbers.insert(2, 103)
print("Updated List:", random_numbers)
```

```
Updated List: [101, 66, 103, 59, 33, 44, 50, 102]
```

[33]:
```
# Q3

third_element = random_numbers[2]
print("Third Element:", third_element)
```

```
Third Element: 103
```

[35]:
```
# Q4

random_strings = ['apple', 'banana', 'cherry']
combined_list = random_numbers + random_strings
print("Combined List:", combined_list)
```

```
Combined List: [101, 66, 103, 59, 33, 44, 50, 102, 'apple', 'banana', 'cherry']
```

[37]:
```
# Q5

print("Elements in Combined List:")
for element in combined_list:
```

```
    print(element)
```

```
Elements in Combined List:
101
66
103
59
33
44
50
102
apple
banana
cherry
```

[49]:
```
# Dictionary

# Q1

person = {'name': 'John', 'age': 25, 'address': 'New York'}
print(person)
```

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

[51]:
```
# Q2

person['phone'] = '1234567890'
print(person['phone'])
```

```
1234567890
```

[43]:
```
# Q3

person.pop('address')
```

[43]:
```
'New York'
```

[45]:
```
# Q4

print(person['age'])
```

```
25
```

[47]:
```
# Q5

print('phone' in person)
```

```
True
```

[53]:
```
# Set

# Q1

set1 = {1, 2, 3, 4, 5}
print(set1)
```

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

[55]:
```
# Q2

set1.add(6)
print(set1)
```

```
{1, 2, 3, 4, 5, 6}
```

[57]:
```
# Q3

set1.remove(3)
print(set1)
```

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

[59]:
```
# Q4

print(len(set1))
```

```
5
```

[61]:
```
# Q5

set2 = {6, 7, 8}
new_set = set1.union(set2)
print(new_set)
```

```
{1, 2, 4, 5, 6, 7, 8}
```

[63]:
```
# Tuple

# Q1

tuple1 = (1, 2, 3, 4)
print(tuple1)
```

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

[65]:
```
# Q2
```

```
print(len(tuple1))
```

```
4
```

[67]:
```
# Q3

tuple2 = (5, 6)
new_tuple = tuple1 + tuple2
print(new_tuple)
```

```
(1, 2, 3, 4, 5, 6)
```

[69]:
```
# Q4

print(new_tuple[:2])
```

```
(1, 2)
```

[71]:
```
# Q5

print(4 in tuple1)
```

```
True
```

[73]:
```
# Tuple

# Exercise 1

first_name = input("Please enter your first name: ")
last_name = input("Please enter your last name: ")

full_name = f"{first_name.upper()} {last_name.upper()}"
initials = f"{first_name[0].upper()} {last_name[0].upper()}"
first_name_len = len(first_name)
last_name_len = len(last_name)
full_name_len = first_name_len + last_name_len

print(f"Your full name is {full_name}")
print(f"Your initials are {initials}")
print(f"First name length is {first_name_len} letters")
print(f"Last name length is {last_name_len} letters")
print(f"Full name length is {full_name_len} letters")

print(f"First name starts with {first_name[0].upper()}")
print(f"First name ends with {first_name[-1].upper()}")
print(f"Last name starts with {last_name[0].upper()}")
print(f"Last name ends with {last_name[-1].upper()}")
```

```
print(f"First name indexes are 0 âĂŞ {first_name_len - 1}")
print(f"Last name indexes are 0 âĂŞ {last_name_len - 1}")

print(f"First name trims 1 {first_name[:3]}")
print(f"First name trims 2 {first_name[1:]}")
print(f"Last name trims 1 {last_name[:3]}")
print(f"Last name trims 2 {last_name[3:]}")
```

```
Please enter your first name:  peter
Please enter your last name:  Cambridge

Your full name is PETER CAMBRIDGE
Your initials are P C
First name length is 5 letters
Last name length is 9 letters
Full name length is 14 letters
First name starts with P
First name ends with R
Last name starts with C
Last name ends with E
First name indexes are 0 âĂŞ 4
Last name indexes are 0 âĂŞ 8
First name trims 1 pet
First name trims 2 eter
Last name trims 1 Cam
Last name trims 2 bridge
```

[75]:
```
# Exercise 2

name = input("Please enter your name: ")

if len(name) > 2:
    encrypted_name = name[0] + '*' * (len(name) - 2) + name[-1]
else:
    encrypted_name = name

print(f"Encrypted name: {encrypted_name}")
```

```
Please enter your name:  John

Encrypted name: J**n
```

[79]:
```
# Exercise 3

sample_list = ['abc', 'xyz', 'aba', '1221']
count = 0
for string in sample_list:
    if len(string) >= 2 and string[0] == string[-1]:
```

```
        count += 1

print(f"Expected Result: {count}")
```

```
Expected Result: 2
```

[81]:
```
# Exercise 4

divisible_by_7 = [num for num in range(1, 1001) if num % 7 == 0]
print(divisible_by_7)
```

```
[7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112, 119, 126, 133,
140, 147, 154, 161, 168, 175, 182, 189, 196, 203, 210, 217, 224, 231, 238, 245,
252, 259, 266, 273, 280, 287, 294, 301, 308, 315, 322, 329, 336, 343, 350, 357,
364, 371, 378, 385, 392, 399, 406, 413, 420, 427, 434, 441, 448, 455, 462, 469,
476, 483, 490, 497, 504, 511, 518, 525, 532, 539, 546, 553, 560, 567, 574, 581,
588, 595, 602, 609, 616, 623, 630, 637, 644, 651, 658, 665, 672, 679, 686, 693,
700, 707, 714, 721, 728, 735, 742, 749, 756, 763, 770, 777, 784, 791, 798, 805,
812, 819, 826, 833, 840, 847, 854, 861, 868, 875, 882, 889, 896, 903, 910, 917,
924, 931, 938, 945, 952, 959, 966, 973, 980, 987, 994]
```

[83]:
```
# Exercise 5

my_list = [9, 12, 15, 18, 21]
divided_by_3_dict = {element: element / 3 for element in my_list}
print(divided_by_3_dict)
```

```
{9: 3.0, 12: 4.0, 15: 5.0, 18: 6.0, 21: 7.0}
```

[ ]:
```

```