Untitled

July 20, 2024

[ ]:
```
#Read the answers of given questions.
```

[1]:
```
#Q1 Discuss string slicing and provide examples.
#Ans. String slicing is a technique in which we are extract the substring from␣
 ↪a main string.
#Example:
name="Rounak"
print(name[0:7])
```

```
Rounak
```

[ ]:
```
#Q2 Explain the key features of lists in Python
#Ans. Key features of list is:-
# 1. it is muttable.
# 2. it is support duplicate itmes.
# 3. it contains diffrent types of items
# 4. it is arrange in ordered
# 5. it is support indexing
```

[2]:
```
#Q3 Describe how to access, modify and delete elements in a list with example
# Ans. 1. Access:- in python we can acess elements by their index.
# Example:-
num=[50, 60, 70, 80]
f1= num[0]
f2= num[1]
f3= num[2]
print(f1, f2, f3)
```

```
50 60 70
```

[3]:
```
# Modifying Elements:-
# We can modify a list by assign new value in a specific index.
# Example:-
num=[50, 60, 70, 80]
num[0]=80
num[1]=90
print(num)
```

```
[80, 90, 70, 80]
```

[7]:
```
# Delete Elements:-
# we can delete elements by using remove(), pop(), del functions.
# Example:-
num=[50, 60, 70, 80]
num.remove(60)
num.pop()
print(num)
```

```
[50, 70]
```

[8]:
```
#compare and contrast tupple and list with example.
# Ans. Feature        List        Tupple
# Mutability        Mutable        Immutable
# Syntax        []        ()
# Performance        Slower        Faster
# Use Cases        Collections that may change        Collections that should␣
 ↪remain constant
# Methods        Many built-in methods        Fewer built-in methods
# Example of lis:-
# Creating a list
my_list = [10, 20, 30, 40, 50]

# Accessing elements
print(my_list[1])  # Output: 20

# Modifying elements
my_list[1] = 25
print(my_list)  # Output: [10, 25, 30, 40, 50]

# Adding elements
my_list.append(60)
print(my_list)  # Output: [10, 25, 30, 40, 50, 60]

# Removing elements
my_list.remove(25)
print(my_list)  # Output: [10, 30, 40, 50, 60]
```

```
20
[10, 25, 30, 40, 50]
[10, 25, 30, 40, 50, 60]
[10, 30, 40, 50, 60]
```

[9]:
```
# Example of tupple:-
# Creating a tuple
my_tuple = (10, 20, 30, 40, 50)
```

```python
# Accessing elements
print(my_tuple[1])  # Output: 20

# Attempting to modify elements (this will raise an error)
try:
    my_tuple[1] = 25
except TypeError as e:
    print(e)  # Output: 'tuple' object does not support item assignment

# Tuples do not have methods like append() or remove()
# We can, however, concatenate tuples to create a new one
new_tuple = my_tuple + (60,)
print(new_tuple)  # Output: (10, 20, 30, 40, 50, 60)

# We can also use tuple unpacking
a, b, c, d, e = my_tuple[:5]
print(a, b, c, d, e)  # Output: 10 20 30 40 50
```

```
20
'tuple' object does not support item assignment
(10, 20, 30, 40, 50, 60)
10 20 30 40 50
```

[ ]: