Name: Sai kesarkar

Email: saiskesarkargoa@gmail.com

Course name: Data analytics July 2024 batch

Assignment name: Data Structure

Submission date: 15th august 2024

Git Link:https:https://github.com/Saisk16/saiskass215082024

Drive link:https://docs.google.com/document/d/1PLA_5oBhNSb68L4j1LF9RLubLLY73gaksYEmo6Ys1a8/edit

1)Discuss string slicing and provide examples1)

Ans. String slicing refers to extracting a fraction of a string with the help of indexing. Slicing allows capturing substrings and characters.

Example

H="temporary"

print(H[3:5])

Output: po

2) Explain the key features of lists in Python.

Ans. List are mutable, easy to use and flexible data structure. The elements of a list can be accessed by an index. List are ordered and allow duplicate values, and different types of values. List modification includes functions like append(), insert(), extend().

3) Describe how to access, modify, and delete elements in a list with examples

#ans.Accessing an element in a list includes mentioning its indexes in square brackets, a range of indexes

Modify:includes operations like Adding an element in a list using the append operation.

Delete: includes removal of an element from the list. This can be done using the .remove command.

```
list1=[2,3,6,"india",92]

print("original list is")

print(list1)

print("accessing an element from a list:")

print(list1[3])

print("modifying a list(adding something)")

list1.append(8)

print(list1)

print("delete an element")

list1.remove(3)

print(list1)
```

output

```
original list is

[2, 3, 6, 'india', 92]

accessing an element from a list:

india

modifying a list(adding something)

[2, 3, 6, 'india', 92, 8]

delete an element
```

```
[2, 6, 'india', 92, 8]
```

Q4. Compare and contrast tuples and lists with examples

ans. list are mutable but tuples are not mutable, list are represented using square brackets and tuples are represented using round brackets.

```
Ds1=(5,6,8,4,"red")

Ds2=[5,6,8,4,"red"]

print("datastructure2 can be modified but datastructure1 cannot be modified")

Ds2.append(99)

Ds1.append(99)
```

Output

```
datastructure2 can be modified but datastructure1 cannot be modified

---------------------------------------------------------------------------

AttributeError                    Traceback (most recent call last)

Cell In[5], line 8

      6 print("datastructure2 can be modified but datastructure1 cannot be modified")

      7 Ds2.append(99)

----> 8 Ds1.append(99)


AttributeError: 'tuple' object has no attribute 'append'
```

Q5.Describe the key features of sets and provide examples of their use

#ans sets are immutable,unordered data structure that allows all types of data but never duplicate data.

```
set1={34,67,88}

type(set1)

output. Set

set2={6,8,90}

print(set1.union(set2))

output

{34, 67, 6, 88, 8, 90}
```

Q6. Discuss the use cases of tuples and sets in Python programming

Ans. Tuples are used to insert records into database, sets are used for finding unique elements in a database.

Q7.Describe how to add, modify, and delete items in a dictionary with examples

```
dict1={"class":"a","rn":"56","name":"sai"} # original dict

dict1["S"]="M"

print(dict1) # modified dict

output.


 {'class': 'a', 'rn': '56', 'name': 'sai', 'S': 'M'}




print(dict1)#original dict

dict1.pop("name")# removing an attribute using pop

print(dict1)
```

output

```
{'class': 'a', 'rn': '56', 'name': 'sai', 'S': 'M'}

{'class': 'a', 'rn': '56', 'S': 'M'}
```

```
print(dict1["rn"])# accessing an attribute
```

output

```
56
```

Q8)Discuss the importance of dictionary keys being immutable and provide examples

Ans. The hash table implementation of dictionaries uses a hash value calculated from the key value to find the key. If the key were a mutable object, its value could change, and thus its hash could also change. But since whoever changes the key object can't tell that it was being used as a dictionary key, it can't move the entry around in the dictionary. Then, when you try to look up the same object in the dictionary it won't be found because its hash value is different. If you tried to look up the old value it wouldn't be found either, because the value of the object found in that hash bin would be different.