1. what are data structure, and why are they important?

Ans1.Data structure are organized ways to store and manage data, enabling efficient access, modification, and munipulation.

2. Explain the differece between mutable and immtuble data types with example.

Ans2. Mutable data types can be modified after creation, while immutable data types cannot.

Example;

Mutable:list,dictionaries, sets

```
 my list = [1, 2, 3];

 my_list[0] = 10 # [10, 2, 3]

immutable: Tuples, strings, integers

my_tuples= (1, 2, 3);

my_tuples[0] = 10 # Error
```

3. what are the main difference between list and tuples in python?

Ans3. The main difference between lists and tuples is that lists are mutable,while tuples are immutable.

4. Describe how dictionaries store data.

Ans4. Discribe store data as key-value pairs.

5. Why might you use a set instead of a list in python?

Ans5. You might use a set instead of a list when you need to store unique elements and perform fast membership testing.

6. What is a string in python, and how is it different from a list?

Ans6. A string is an immutable sequence of characters, whereas a list is a mutable collection of iteams.

7. How do tuples ensure data integrity in python?

Ans7. Tuples ensure data integrity in python by being immutable, meaning their contents cannot be modified after creation, thus preventing unintended changes.

8. What is a hash table, and how does it related to dictionary in python?

Ans8. A hash table is a data structure that maps keys to values, and python dictionaries are implemented as hash tables.

9. Can list contain different data types in python?

Ans9. Yes, Python list can contain different data types.

10.Explain why string are immutable in python.

Ans10. Strings are immutable in python to ensure data integrity, security, and efficient memory management.

11. What advantages do dictionaries offer over lists for certain tasks?

Ans11. Dictionaries offer fast lookups, efficient data retrieval, and meaningful key-based access, making them ideal for tasks that require associating keys with values.

12. Describe a scenario where using a tuple would be preferable over a list?

Ans12. When data shouldn't be changed, like days of the week oor months of the year, tuples are preferable over lists due to their immutability, ensuring data integrity.

13. How do sets handle duplicate values in Python?

Ans13. Sets in python automatically eliminate duplicate values, storing only unique elements.

14. How does the “in” keyword work differently for lists and dictionaries?

Ans14. For lists, 'in' checks for values, while for dictionaries, in checks for keys.

15. Can you modify the elements of a tuple? Explain why or why not.

Ans15. No, tuples are immutable in python, meaning their elements cannot be modified after creation.

16. What is a nested dictionary, and give an example of its use case?

Ans16. A nested dictionary is a dictionary containing another dictionary as its value, useful for representing complex data like student records.

Examples:

student ={

```
 'name':'john',
  'grades':{
       'math': 85,
        'science' 90
  }

}
```

17. Describe the time complexity of accessing elements in a dictionary.

Ans17. The time complexity of accessing elements in a dictionary is 0(1), making dictionaries efficient for fast lookups.

18. In what situations are lists preferred over dictionaries?

Ans 18. List are preferred over dictionaries when you need to store ordered collections of items or when you need to access elements by their index position.

19. Why are dictionaries considered unordered, and how does that affect data retrieval ?

Ans19. Dictionaries in order python verisons are unordered, affecting data retrieval, but in python 3.7+, they maintain insertion order.

20. Explain the difference between a list and a dictionary in terms of data retrieval.

Ans20. Lists retrieve data by index position, while dictionaries retrieve data by key.

practical

1. write a code to create a string with your name and print it.

```
str = "himani"
print("my name is", str)
```

```
my name is himani
```

2. Write a code to find the length of the string "Hello World".

```
str ="hello world"
l=len(str)
print("lengh=",l)
```

```
lengh= 11
```

3. Write a code to slice the first 3 characters from the string "Python Programming".

```
str="python programming"
nstr=str[::-1]
print(nstr)
```

```
gnimmargorp nohtyp
```

4. Write a code to convert the string "hello" to uppercase.

```
str= "hello"
print(str.upper())
```

```
HELLO
```

5.Write a code to replace the word "apple" with "orange" in the string "I like apple".

```
str = "I like apple"
new_str= str.replace("apple", "orange")
print(new_str)
```

```
I like orange
```

6. Write a code to create a list with numbers 1 to 5 and print it.

```
l=[1, 2, 3, 4, 5]
print(l)
```

```
[1, 2, 3, 4, 5]
```

7. Write a code to append the number 10 to the list [1, 2, 3, 4].

```
l = [1, 2, 3,4]
l.append(10)
print(l)
```

```
[1, 2, 3, 4, 10]
```

8. Write a code to remove the number 3 from the list [1, 2, 3, 4, 5].

```
l = [1, 2, 3, 4, 5]
l.remove(3)
print(l)
```

```
[1, 2, 4, 5]
```

9. Write a code to access the second element in the list ['a', 'b', 'c', 'd'].

```
l = ['a','b', 'c','d',]
print(l[2])
```

```
c
```

10. Write a code to reverse the list [10, 20, 30, 40, 50].

```
l =[10, 20, 30, 40 ,50]
nl = l [::-1]
print(nl)
```

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

11. Write a code to create a tuple with the elements 100, 200, 300 and print it.

```
t = (100, 200, 300)
print(t)
```

```
(100, 200, 300)
```

12. Write a code to access the second-to-last element of the tuple ('red', 'green', 'blue', 'yellow').

```
t =('red', 'green', 'blue','yellow')
print (t[1:])
```

```
('green', 'blue', 'yellow')
```

13. Write a code to find the minimum number in the tuple (10, 20, 5, 15).

```
t = (10, 20, 5, 15)
min_value = min(t)
print("minimum number:", min_value)
```

```
minimum number: 5
```

14. Write a code to find the index of the element "cat" in the tuple ('dog', 'cat', 'rabbit').

```
animals = ('dog', 'cat', 'rabbit')
index = animals.index('cat')
print("Index of 'cat':", index)
```

```
Index of 'cat': 1
```

15. Write a code to create a tuple containing three different fruits and check if "kiwi" is in it.

```
fruit = ("apple", "banana","orange")
if "kiwi" in fruit:
    print("kiwi is in the tuples.")
else:
   print("kiwi is not in the tuple.")
```

```
kiwi is not in the tuple.
```

16. Write a code to create a set with the elements 'a', 'b', 'c' and print it.

```
my_set = {'a', 'b', 'c'}
print(my_set)
```

```
{'c', 'a', 'b'}
```

17. Write a code to clear all elements from the set {1, 2, 3, 4, 5}.

```
my_set = {1, 2, 3, 4, 5}
my_set.clear()
print(my_set)
```

```
set()
```

18. Write a code to remove the element 4 from the set {1, 2, 3, 4}.

```
my_set = {1, 2, 3, 4}
my_set.remove(4)
print(my_set)
```

```
{1, 2, 3}
```

19. Write a code to find the union of two sets {1, 2, 3} and {3, 4, 5}.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
union_set = set1.union(set2)
print(union_set)
```

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

20. Write a code to find the intersection of two sets {1, 2, 3} and {2, 3, 4}.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
intersection_set = set1.intersection(set2)
print(intersection_set)
```

```
{2, 3}
```

21. Write a code to create a dictionary with the keys "name", "age", and "city", and print it.

```
#create the dictionary
my_dict = {
    "name" : "Himani",
    "age" : 20,
    "city" : "Alwar"
}
#print the dictionary
print(my_dict)
```

```
{'name': 'Himani', 'age': 20, 'city': 'Alwar'}
```

22. Write a code to add a new key-value pair "country": "USA" to the dictionary {'name': 'John', 'age': 25}.

```
my_dict = {'name': 'john', 'age': 25}
my_dict['country'] = 'USA'

print(my_dict)
```

```
{'name': 'john', 'age': 25, 'country': 'USA'}
```

23. Write a code to access the value associated with the key "name" in the dictionary {'name': 'Alice', 'age': 30}.

```
my_dict = {'name': 'Alice', 'age' : 30}
name_value = my_dict['name']
print(name_value)
```

```
Alice
```

24. Write a code to remove the key "age" from the dictionary {'name': 'Bob', 'age': 22, 'city': 'New York'}.

```
my_dict = {'name': 'Bob', 'age' :22, 'city': 'New york'}
del my_dict['age']
print(my_dict)
```

```
{'name': 'Bob', 'city': 'New york'}
```

25. Write a code to check if the key "city" exists in the dictionary {'name': 'Alice', 'city': 'Paris'}.

```
my_dict = {'name': 'Alice', 'city': 'Paris'}

if 'city' in my_dict:
    print("key 'city' exists in the dictionary")
else:
   print("key 'city' does not exist.")
```

```
key 'city' exists in the dictionary
```

26. Write a code to create a list, a tuple, and a dictionary, and print them all.

```
# create a lists
my_list = [1, 2, 3, 4]

#create a tuples
my_tuples = ('apple', 'banana', 'cherry')
# create a lists
my_list = [1, 2, 3, 4]

#create a tuples
my_tuples = ('apple', 'banana', 'cherry')

#create a dictionary
my_dict = {'name': 'Alice', 'age': 30, 'city':'New York'}
```

```
#print them all
print("List:", my_list)
print("Tuple:", my_tuples)
print("Dictionary:", my_dict)
```

```
List: [1, 2, 3, 4]
Tuple: ('apple', 'banana', 'cherry')
Dictionary: {'name': 'Alice', 'age': 30, 'city': 'New York'}
```

27. Write a code to create a list of 5 random numbers between 1 and 100, sort it in ascending order, and print the result.(replaced)

```
import random
import random

# Create a list of 5 random numbers between 1 and 100
random_list = [random.randint(1, 100) for _ in range(5)]

# Sort the list in ascending order random_list.sort()
random_list.sort()

# Print the result
print("sorted list:", random_list)
```

```
sorted list: [9, 11, 16, 56, 74]
```

28. Write a code to create a list with strings and print the element at the third index.

```
# Create a list with strings
my_list = ["apple", "banana", "cherry", "date", "elderberry"]

# Print the elements at the third
index = 3 # index starts from 0
print("Element at index 3:", my_list[3])
```

```
Element at index 3: date
```

29. Write a code to combine two dictionaries into one and print the result.

```
# Define two dictionaries
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd':4}

# Combine the dictionaries
combined_dict = {**dict1, **dict2}

# Print the result
print("Combined Dictionary:", combined_dict)
```

```
Combined Dictionary: {'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

30. Write a code to convert a list of strings into a set.

```
# Define a list of strings
my_list = ["apple", "banana", "cherry", "apple", "banana"]

# Convert the list into a set
my_set = set(my_list)

# Print the result
print("set:", my_set)
```

```
set: {'apple', 'banana', 'cherry'}
```