1) Discuss string slicing and provide examples.

Answer:
String slicing in Python is used to extract a part (substring) of a string using index positions. The slicing syntax is:

```
string[start : end : step]
```

- start → starting index (inclusive)
- end → ending index (exclusive)
- step → increment value

```
text = "Python Programming"


print(text[0:6])     # Output: Python

print(text[7:18])    # Output: Programming

print(text[:6])      # Output: Python

print(text[7:])      # Output: Programming

print(text[::2])     # Output: Pto rgamn

print(text[::-1])    # Output: gnimmargorP nohtyP
```

2) Explain the key features of lists in Python.

Answer:
A list is a collection data type used to store multiple items in a single variable.

Key features of lists:

- Lists are ordered.
- Lists are mutable (can be changed).
- Lists allow duplicate values.
- Lists can store different data types.
- Lists are defined using square brackets [ ].

```
my_list = [10, "Harsha", 3.5, True]
```

```
my_list = [10, "Harsha", 3.5, True]
```

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

Answer:

Accessing elements:

```
a = [10, 20, 30, 40]
```

```
print(a[0])     # 10

print(a[-1])    # 40
```

modifying

```
a[1] = 25

print(a)       # [10, 25, 30, 40]
```

deleting

```
a.remove(30)    # removes value 30

del a[0]       # removes element at index 0

a.pop()        # removes last element

print(a)
```

4) Compare and contrast tuples and lists with examples.

Answer:
Both tuples and lists are used to store multiple values. However, they have key differences.

Lists are mutable, meaning their elements can be changed. Tuples are immutable, meaning their elements cannot be changed after creation.

Lists use square brackets [ ], while tuples use parentheses ( ).

Example of list:

```
a = [1, 2, 3]

a[0] = 10


b = (1, 2, 3)

# b[0] = 10   # This will give an error
```

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

Answer:
A set is an unordered collection of unique elements.

Key features of sets:

- Sets do not allow duplicate values.
- Sets are unordered (no indexing).
- Sets are mutable.
- Defined using curly braces { }.

Example:

```
s = {1, 2, 3, 3, 4}
print(s)    # Output: {1, 2, 3, 4}
```

set operations:

```
a = {1, 2, 3}
b = {3, 4, 5}


print(a.union(b))       # {1, 2, 3, 4, 5}
print(a.intersection(b)) # {3}
```

Sets are useful for removing duplicates and performing mathematical set operations.

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

Answer:

Use cases of tuples:

- Storing fixed data that should not change.
- Representing coordinates (x, y).
- Returning multiple values from a function.

Example:

```
def get_values():
   return (10, 20)
```

Use cases of sets:

- Removing duplicate values from a list.
- Performing union, intersection, and difference operations.
- Membership testing.

```
numbers = [1, 2, 2, 3]
unique_numbers = set(numbers)
print(unique_numbers)
```

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

Answer:
A dictionary stores data in key-value pairs and is defined using curly braces { }.

Adding items:

```
student = {"name": "Harsha", "age": 23}
```

student["city"] = "Pune"

modifying
student["age"] = 24

deleting :

del student["city"]

student.pop("age")

print(student)

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

Answer:
Dictionary keys must be immutable because they are used to uniquely identify values. Immutable objects ensure that the key does not change after creation, which maintains data integrity.

Valid dictionary keys:

- int
- float
- string
- tuple (if it contains immutable elements)

Example:

d = {1: "One", "name": "Harsha", (1, 2): "Tuple Key"}

Invalid key example (list is mutable):

# d = {[1, 2]: "Invalid"}   # This will give an error