Unit 2

1-Mark Questions

Strings:

1. **Q:** How do you create a string in Python?
   A: By enclosing characters in single (`' '`), double (`" "`), or triple quotes (`''' '''` / `""" """`).
2. **Q:** What is string indexing?
   A: Accessing individual characters of a string using their position (index).
3. **Q:** What is the index of the first character in a Python string?
   A: `0` (zero-based indexing).
4. **Q:** Write the output of `"Hello"[1]`.
   A: `e`.
5. **Q:** What does slicing do in strings?
   A: It extracts a portion (substring) of a string using `[start:end]`.
6. **Q:** Write the output of `"Python"[::-1]`.
   A: `"nohtyP"` (reversed string).
7. **Q:** Name one string method used to convert string to uppercase.
   A: `upper()`.
8. **Q:** Which method is used to remove whitespace from both ends of a string?
   A: `strip()`.
9. **Q:** Is a Python string mutable or immutable?
   A: Immutable.
10. **Q:** Write the output of `"apple".find("p")`.
    A: `1`.

Lists:

1. **Q:** How do you create an empty list in Python?
   A: `my_list = []` or `my_list = list()`

2. **Q:** What is the index of the first element in a Python list?
   A: `0`
3. **Q:** Write the slicing expression to get the first 3 elements of a list `nums`.
   A: `nums[:3]`
4. **Q:** Which keyword is used in **list comprehension**?
   A: `for`
5. **Q:** Name one method used to add an element to a list.
   A: `append()`

Tuples:

6. **Q:** Can tuples be modified after creation?
   A: No, tuples are immutable.
7. **Q:** How do you access the second element of a tuple `t`?
   A: `t[1]`
8. **Q:** Name a method used with tuples.
   A: `count()` or `index()`

Sets:

9. **Q:** How do you create an empty set in Python?
   A: `set()` (not `{}` because that creates a dictionary)
10. **Q:** Which operation finds common elements between sets?
    A: Intersection (`&` or `.intersection()`)
11. **Q:** Name one method used with sets.
    A: `add()`, `remove()`, `union()`

Dictionaries:

12. **Q:** How do you create an empty dictionary in Python?
    A: `{}` or `dict()`
13. **Q:** How do you access the value of key `"name"` in dictionary `d`?
    A: `d["name"]`
14. **Q:** Name one method used with dictionaries to get all keys.
    A: `keys()`

15. **Q:** Write an example of dictionary comprehension to create a square dictionary for numbers 1 to 3.
    A: `{x: x**2 for x in range(1, 4)}`
16. **Q:** Which function is commonly used to iterate over key-value pairs in a dictionary?
    A: `.items()`

2-Mark Questions

Strings:

1. **Q:** Explain string immutability in Python with an example.
   A: Strings cannot be changed after creation.

2. Q: Differentiate between `isalpha()` and `isdigit()` methods.
A: `isalpha()` → Returns `True` if all characters are alphabets.

- `isdigit()` → Returns `True` if all characters are digits.

3. **Q:** What is string formatting? Give an example.
   A: Inserting variables inside a string using `format()` or f-strings.

Lists:

1. **Q:** Differentiate between `append()` and `extend()` methods in lists.
   A:
   - `append()` adds a single element at the end of the list.
   - `extend()` adds multiple elements from another iterable to the list.
2. **Q:** Write a Python program to create a list of squares of numbers from 1 to 5 using list comprehension.
   A:

```
squares = [x**2 for x in range(1, 6)]
print(squares)   # [1, 4, 9, 16, 25]
```

Tuples:

3. **Q:** Why are tuples faster than lists in Python?
   A: Tuples are immutable and stored in a fixed way, so Python optimizes memory usage and lookup, making them faster than lists.

4. **Q:** Write a Python statement to find the index of element `20` in tuple `t = (10, 20, 30)`.
   A:

```
t = (10, 20, 30)
print(t.index(20))
```

Sets:

5. **Q:** Write the difference between `remove()` and `discard()` in sets.
   A:
   - `remove()` raises an error if the element is not present.
   - `discard()` does not raise an error if the element is missing.
6. **Q:** Write a Python program to find the union of two sets.
   A:

```
A = {1, 2, 3}
B = {3, 4, 5}
print(A | B)
```

Dictionaries:

7. **Q:** How is a dictionary different from a list in Python?
   A:
   - Dictionary stores data as **key-value pairs**, while a list stores elements sequentially by index.
   - Dictionaries are unordered (before Python 3.7) but lists are ordered.
8. **Q:** Write a dictionary comprehension to create a dictionary of numbers 1 to 5 with their cubes.
   A:

```
cubes = {x: x**3 for x in range(1, 6)}
print(cubes)
```

5-Mark Questions and Answers

Q1. Explain string slicing with an example. Write a program to reverse a string using slicing.

Answer:

* **String slicing** allows extracting substrings using `[start:end:step]`.
* If step = `-1`, it reverses the string.

```python
```

```
text = "PythonProgramming"

# Slicing examples
print(text[0:6])    # Output: Python
print(text[-11:-6]) # Output: Progr

# Reverse string using slicing
reversed_text = text[::-1]
print("Reversed:", reversed_text)
```

Q2. What are common string methods? Demonstrate at least three with examples.

Answer:
Common string methods:

1. `upper()` → converts to uppercase.
2. `replace()` → replaces substrings.
3. `split()` → splits a string into a list.

```python
s = "hello world"
print(s.upper())          # HELLO WORLD
print(s.replace("world","Python"))  # hello Python
print(s.split())          # ['hello', 'world']
```

Q3. Explain list comprehension with an example. Write a program to generate a list of squares of numbers from 1 to 10.

Answer:

* **List comprehension** provides a concise way to create lists.
* Syntax: `[expression for item in iterable if condition]`

```python
# List comprehension example
squares = [x**2 for x in range(1, 11)]
print("Squares:", squares)
```
Output: `[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]`

Q4. Differentiate between lists and tuples. Write a program to access elements using indexing.

Answer:

* **Lists**: Mutable (can be changed).
* **Tuples**: Immutable (cannot be changed).

```python
# List and Tuple example
my_list = [10, 20, 30]
my_tuple = (100, 200, 300)

print("List element at index 1:", my_list[1])   # 20
print("Tuple element at index 2:", my_tuple[2]) # 300
```

Q5. Explain set operations with an example. Write a program to find common elements between two sets.

Answer:

* Sets support operations like **union, intersection, difference**.

```python
A = {1, 2, 3, 4}
B = {3, 4, 5, 6}

print("Union:", A | B)         # {1,2,3,4,5,6}
print("Intersection:", A & B)   # {3,4}
print("Difference:", A - B)     # {1,2}
```

Q6. What is dictionary comprehension? Write a program to create a dictionary of numbers and their squares.

Answer:

* **Dictionary comprehension** is a compact way of creating dictionaries.
* Syntax: `{key: value for item in iterable}`

```python
squares_dict = {x: x**2 for x in range(1, 6)}
print("Squares Dictionary:", squares_dict)
```

Output: `{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}`

Q7. Write a program to iterate over a dictionary and print keys and values. Explain two dictionary methods.

Answer:

```python
student = {"name": "Ravi", "age": 20, "marks": 88}

# Iterating
for key, value in student.items():
```

```
    print(key, ":", value)

# Methods
print(student.keys())   # dict_keys(['name','age','marks'])
print(student.values()) # dict_values(['Ravi',20,88])
```

* `keys()` → returns all keys.
* `values()` → returns all values.

More 5-Mark Questions:

Strings:

1. Write a Python program to check whether a given string is a palindrome.
2. Write a Python program to check whether a string contains only digits.
3. Write a Python program to count the number of vowels and consonants in a string.
4. Write a Python program to find the longest word in a given sentence.
5. Write a Python program to replace all spaces in a string with `-`.
6. Write a Python program to reverse each word in a given sentence.
7. Write a Python program to find the number of words in a string without using the `split()` function.

Lists:

1. Write a Python program to read 5 numbers into a list and then print the largest and smallest numbers.
2. Explain list slicing with suitable examples. Write a program to reverse a list using slicing.
3. Write a Python program using list comprehension to generate a list of even numbers between 1 and 20.

Tuples:

4. Write a Python program to create a tuple with numbers and print all the elements at odd indices.
5. Explain any four properties of tuples with examples.
6. Write a program to convert a list into a tuple and print its length.

Sets:

7. Write a Python program to find common elements between two sets and also find elements present only in the first set.
8. Explain any four methods of sets with examples.
9. Write a program to remove duplicate elements from a list using a set.

Dictionaries:

10. Write a Python program to store names and marks of 5 students in a dictionary and print the student with the highest marks.
11. Explain dictionary comprehension with an example. Write a program to create a dictionary of numbers and their squares from 1 to 10.
12. Write a Python program to count the frequency of each character in a given string using a dictionary.