🧱 Interview Questions : Set in Python

🎯 **Objective**:

Understand the concept of sets in Python and practice basic operations such as union, intersection, difference, and set methods.

🟢 Beginner Level (Basic Syntax & Concepts)

1. What is a set in Python? How is it different from a list or tuple?
2. How do you create a set in Python? Provide an example.
3. What happens when you add a duplicate element to a set?
4. How do you add and remove elements from a set? Name the methods.
5. What is the difference between `remove()` and `discard()` in sets?
6. How do you check if an element exists in a set?
7. What will be the output?

```
s = {1, 2, 3}
    s.add(2)
    print(s)
```

🟡 Intermediate Level (Operations & Use Cases)

8. What are the time complexities for common set operations (add, remove, lookup)?
9. Explain the difference between union, intersection, and difference with examples.
10. How can sets be used to remove duplicates from a list? Provide a code snippet.
11. What is a `frozenset` in Python? How is it different from a regular set?
12. Write a function that takes two lists and returns the common elements using sets.
13. Given two sets, find the symmetric difference.
14. What does `set.update()` do? How is it different from `union()`?
15. What is the result of this code?

```
a = {1, 2, 3}
b = {2, 3, 4}
print(a & b, a | b, a - b)
```

🔵 Advanced Level (Practical Scenarios & Algorithmic Thinking)

16. Write a function to find if two lists have any common elements using sets. Optimize for time.
17. Write a function to count the number of unique words in a paragraph using sets.
18. You have a list of 1 million integers. How would you efficiently find all duplicates?
19. Explain how sets are implemented internally in Python (hashing, hash tables).
20. Write a function that finds all unique pair sums in a list (e.g., `pair_sums([1, 2, 3]) => {3, 4, 5}`).
21. Given a log of user IDs, how would you use sets to find how many unique users accessed a service?
22. Can a set contain another set as an element? Why or why not? How can this limitation be overcome?
23. What happens if you try to add a list to a set? Why?

🔴 Expert Level (Design, Optimization & Edge Cases)

24. Design a function that finds all common elements between multiple sets (not just two).
25. Write a function that takes a list of strings and returns a list of strings that appear in more than one list (multi-set intersection).
26. Create a system to track duplicate files by their content using sets. What data structure would you combine with sets?
27. You are given a stream of integers. How would you use a set to detect the first duplicate efficiently?
28. Explain set performance considerations in memory-heavy applications (e.g., large datasets).
29. What is the difference between `set comprehension` and `for-loop + add()`? Give an example.
30. Write a function that uses set operations to validate if a Sudoku row/column contains unique numbers (1–9).

Interview Questions : Set in Python