Python List, Tuple, Dictionary, and Set Interview Questions with Answers

Q: 1. What is a list in Python?

A: A list is a built-in data structure in Python that can hold an ordered collection of items. Lists are mutable, meaning their contents can be changed. Example:

my_list = [1, 2, 3, 'hello']

Q: 2. How are lists different from arrays?

A: Arrays are provided by external libraries like NumPy or array module and are homogeneous (all elements have the same data type). Lists, however, can contain mixed data types.

Q: 3. How do you add elements to a list?

A: Using append(): Adds a single element.

Using extend(): Adds multiple elements.

Using insert(): Adds an element at a specific position.

Example:

my_list.append(5) # Adds 5

Q: 4. How do you reverse a list?

A: Using reverse() method:

my_list.reverse()

Or using slicing:

reversed_list = my_list[::-1]

Q: 5. What is the difference between sort() and sorted()?

A: sort(): Sorts the list in place (no return value).

sorted(): Returns a new sorted list, leaving the original list unchanged.

Q: 6. What is a tuple, and how is it different from a list?

A: A tuple is an immutable sequence of items. Once created, its contents cannot be changed.

Difference:

- Lists are mutable, tuples are immutable.

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

Q: 7. What happens if you try to modify a tuple?

A: Since tuples are immutable, attempting to modify their elements will raise a TypeError.

Example:

```
my_tuple = (1, 2, 3)
my_tuple[0] = 5  # Raises TypeError
```

Q: 8. What is a dictionary in Python?

A: A dictionary is an unordered collection of key-value pairs. Keys must be unique and immutable, while values can be any data type.

Example:

```
my_dict = {'name': 'John', 'age': 25}
```

Q: 9. How do you add or update elements in a dictionary?

A: To add a key-value pair:

```
my_dict['key'] = 'value'
```

To update:

```
my_dict.update({'key': 'new_value'})
```

Q: 10. How do you remove keys from a dictionary?

A: Using pop(): Removes a key and returns its value.

Using del: Deletes a key-value pair.

Example:

```
del my_dict['key']
```

Q: 11. What is a set, and how is it different from a list?

A: A set is an unordered collection of unique elements. Sets do not allow duplicates and do not maintain order.

Example:

my_set = {1, 2, 3}

Q: 12. How do you remove duplicates from a list using a set?

A: You can convert the list into a set and back into a list:

my_list = [1, 2, 2, 3]

unique_list = list(set(my_list))

Q: 13. How do you perform union, intersection, and difference operations on sets?

A: Union: A | B

Intersection: A & B

Difference: A - B

Example:

A = {1, 2, 3}, B = {3, 4}

A | B => {1, 2, 3, 4}

Q: 14. How can you check if a key exists in a dictionary?

A: Using the 'in' keyword:

if 'key' in my_dict: print('Key exists')

Q: 15. Can a dictionary have duplicate keys?

A: No, dictionary keys must be unique. If a duplicate key is added, the last value will overwrite the previous one.