Python lists, Sets, Tuples And Dictionaries.

OBJECTIVES.

1) Which of the following is an ordered collection of elements where modifications are allowed.

D

2) Which data structures uses curly braces {} and Consists of Key-value pairs.

D

3) In Python, which data structure is immutable -

C

4) Which data structure is best suited for storing unique elements.

B.

5) Which data structure uses square brackets to store elements.

A

(6) Describe a real world scenario where you would prefer using a tuple over other data structures like lists and dictionaries.

They are used for locating Coordinates on a map. imagine your are ~~making~~ looking for the location of a buried treasure on a map. You would likely to use a tuple to store the X and Y Coordinates Eg (5, 3). These values don't need to be changed or re-ordered later.

(7) Explain the difference between a ~~dict~~ set and a dictionary in python. and provide an example where would be each be most appropriate to use.

Sets:
- Unordered Collection of unique elements.
- Sets store elements without maintaining a specific order.
- Each element must be unique.
- Duplicates are not allowed.
- Uses Cases
- checking membership (if an element exists in the set)
- finding the difference, union or intersection between sets
- storing unique elements like colors.

Example

fruits = {"apple", "banana", "orange", "apple"}

print(fruits) # output: {'apple', 'banana', 'orange'}
↳ Duplicate apple was removed.

Dictionaries

- Collection of Key-value pairs.
- Keys must be unique and immutable (like strings and tuples) while Values can be of any data type.

User Cases:

Looking up information based on a Key (like finding data phone number by name

Associating data with labels (like storing student names and their grades)

Example: Student grades.

```
grades = {"Alice": 90, "Bob": 80, "Charlie": 95}
print(grades["Alice"])
    #output: 90
```

The dictionary allows you to efficiently access student's grade using their name (Key). This wouldn't be possible with a set as sets don't associate values with elements.

Feature	Set	Dictionary
Order	Unordered	Unordered
Duplicates	Not allowed	Not allowed for Keys.
Mutability	Mutable	Mutable
Key data type	None/Any hashable data type	Must be unique and immutable (eg. strings, tuples)
Value data type	Any	Any

Success has a simple formula: do your best, and people may like it.

(8) Discuss the advantages and disadvantages of using a list compared to a set while dealing with large data set.

Lists

Advantages:

• Ordered data → Lists maintain the order in which elements are added. This can be crucial if you need to process the data in a specific sequence.

• Flexibility: lists can hold elements of any data type including duplicates. This provides greater flexibility and handling of data sets.

• Familiar syntax: Lists use familiar square bracket notation [ ] for creation and access. which is easier to understand for beginners.

Disadvantages:

• Search performance: searching for specific elements within a large list can be slow, especially as the list grows. This is because linear search algorithm are typically used. which require each element in the list.

• Memory usage: lists can consume more memory compared to sets, especially if they contain duplicates or a mix of data types.

Modification operations: Inserting or deleting elements in a large list can be expensive as elements might need to be shifted to accomodate o changes

Sets:

Advantages:

- Fast search: searching for specific elements within a large set is very efficient. Sets utilize hash tables for element look up. which offer a constant average flexibility.

Memory Efficiency: Sets generally use less memory than lists, especially when dealing with large datasets containing unique elements.

Unique Elements: Sets inherently ensure that each element is unique, which can be beneficial if you need to avoid duplicates in your data.

Disadvantages:

- Unordered data: Sets do not maintain the order in which elements are added.

- limited data types: Sets typically only allow hashable data types like strings, numbers or tuples as elements. This can restrict their use when dealing with more complex data structures.

less intuitive syntax: working with sets might require understanding functions like add, remove and discard for element manipulation which might be less familiar with indexing.

Summary:

- Choose lists if you need to maintain order, handle diverse data types or prefer ease of use. Lists might be a better choice either for large datasets. However, be aware of potential implications for searching and modification operations.
- Choose sets if fast lookups, ensuring efficiency and ensuring unique elements are better choice for large datasets. However, keep in mind sets do not preserve order and have limitations on data types.