

Part .1

Objective Type Quiz

- 1.A
- 2. D
- 3. C
- 4.B
- 5.A

Part 2

Non-objective Type Quiz

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

In cases where we are working with geographic coordinates (latitude and longitude) to locate places on a map. Since change of coordinates means change of location, in this case a tuple is the best option since elements in a tuple are unchangeable unlike lists and dictionaries. Once a location is defined, its coordinates don't change.

7. Explain the difference between a set and a dictionary in Python, and provide an example where each would be most appropriate to use.

A set is an unordered collection of unique elements. Thus no duplicates in a set. While a dictionary is an unordered collection of mutable items that have key and value pairs.

A set is used when one needs to store a collection of unique elements and also quickly check if an element is already in the collection. For example; If one is writing a program to keep track of unique user IDs, they use a set to store the IDs and quickly check if a user is already registered.

```
user_ids = {4584651, 'EBU023', 'CM902657HGF12'}
```

A dictionary is used when one needs to store data with associated keys and need to retrieve data by its keys. For example when one is building an address book, they can use a dictionary to store names as the keys and their corresponding phone numbers, age and city.

```
person = {  
    "name" : "Betty",  
    "age" : 30,  
    "gender" : "female",  
    "city" : "Kampala",  
}
```

8. Discuss the advantages and disadvantages of using a list compared to a set when dealing with a large dataset.

Advantages

Lists maintain the order in which elements are added which is crucial when working with large datasets where the order of elements matters while sets do not. If the order of the elements in your dataset is important, a list would be a better choice.

Lists allow duplicate elements which can be beneficial when working with large datasets where duplicates are meaningful and need to be preserved while sets only store unique elements.

Lists provide direct access to elements using indices which can be beneficial when you need to retrieve specific elements quickly. Sets on the other hand do not support direct access to elements.

Disadvantages

Lists can use more memory than sets, especially if the data has many duplicates.

Checking for membership in set can be slower compared to a set. Sets are designed for efficient membership testing, making them faster when you need to check if an element exists in a dataset.

Lists allow duplicate elements which means you may need to implement additional logic to handle and eliminate duplicate manually while on the other hand sets automatically enforce uniqueness, making it easier to work with unique data.

9. Illustrate a situation where you might need to convert a list into a dictionary and vice versa, highlighting the challenges and considerations involved.

If you have a list of items with associated values or attributes, converting it to a dictionary can help you group and aggregate the data based on a certain criteria. For example, you can group a list of students' names and their grades by creating a dictionary where the names are the keys and the grades are the values.

In some scenarios, you might have a dictionary but you need to extract specific information or perform operations that are easier with a list. Converting the dictionary to a list allows you access the data sequentially or perform operations that require an ordered collection.

The following challenges and considerations are kept in mind;

When converting a list to a dictionary, make sure that the keys are unique. Duplicate keys in the list would result in overwriting values in the dictionary.

Lists maintain the order of elements, while dictionaries do not. When converting between the two, be aware that the order of elements may change.

Data integrity: Ensure that the list or dictionary is structured consistently with alternating keys and values in the list or matching key-value pairs in the dictionary. Any deviation can lead to incorrect conversions.

10. Reflect on how the choice of data structure impacts the efficiency and readability of your code, using examples from your own programming experiences.

Efficiency:

Imagine you have a large dataset of student records and you need to frequently search for a student's information based on their ID number. If you store the records in a list, you would need to check through the entire list each time to find the desired student which is time consuming.

However if you use a dictionary with the student ID as the key and the student's information as the value, you can directly access the desired student's information.

Readability:

Let us consider a list of employee names and their corresponding salaries. If you store this data a list of tuples, each containing the name and salary, it might be harder to understand the structure and meaning of the data at a glance. However if you use a dictionary with the employee names as keys and their salaries as values, the code becomes more readable. You can easily access an employee's salary by their name as the key, making the code more self-explanatory and easier to understand for other developers.