Maps and list in Dart

n Dart, both lists and maps are fundamental data structures used for storing and organizing data. Each serves a different purpose and offers distinct functionalities.

Lists in Dart:

A list in Dart is an ordered collection of elements where each element can be accessed using an index. Lists in Dart are zero-indexed, meaning the first element is at index 0, the second element at index 1, and so on. Lists can hold elements of any data type, including other lists.

```
// Creating a list of integers
List<int> numbers = [1, 2, 3, 4, 5];

// Accessing elements of the list
print(numbers[0]); // Output: 1
print(numbers[2]); // Output: 3

// Adding elements to the list
numbers.add(6);
numbers.addAll([7, 8]);

// Removing elements from the list
numbers.remove(3);
numbers.removeAt(0);

// Length of the list
print(numbers.length); // Output: 7

// Iterating over the list
for (int number in numbers) {
  print(number);
}
```

Output

```
1
3
6
7
8
```

Maps in Dart:

A map in Dart is a collection of key-value pairs where each unique key maps to a value. Unlike lists, maps are not ordered. Each key in a map must be unique, but the values can be duplicated. Maps can hold elements of any data type as keys and values

```dart
// Creating a map of strings to integers
Map<String, int> ages = {
  'Alice': 25,
  'Bob': 30,
  'Charlie': 35,
};

// Accessing values using keys
print(ages['Alice']); // Output: 25
print(ages['Bob']);   // Output: 30

// Adding new key-value pairs
ages['David'] = 40;

// Removing key-value pairs
ages.remove('Charlie');

// Checking if a key exists
print(ages.containsKey('Alice')); // Output: true

// Length of the map
print(ages.length); // Output: 3
```

```
// Iterating over the map
ages.forEach((key, value) {
  print('$key: $value');
});
```

Output

```
25
30
David: 40
Alice: 25
Bob: 30
David: 40
Alice: 25
Bob: 30
```