[Dart Programming Language] - Lecture [7]

[Exercises on Functions]

1. Introduction

This lecture will be full of exercises that explains the concept of functions and how to use them to solve a problem.

Learning Objectives

1. Understand how to use functions.
2. Getting used to functions.

2. Main Content

Q1. Write a function to check if a number is even or odd?

Before we start writing code immediately, we must first consider and review the theoretical guidelines before using functions.

1. we need first to determine what does this function need as parameters.
   a. How many parameters
   b. What type of parameters do we need?
2. Then we determine what the result of the function is, what will this function give us?
   a. What is the return value?
   b. What is the type of the return value?
3. Then we write the body of the function that achieves the result we want.

So if we apply these steps on the question we get the following:

1. This function requires a number to check.
   a. Only one parameter is needed
   b. Since this parameter is a number, we will make it integer
2. This function checks if a number is even or odd, so the result of this function will be either even or odd. We can describe this result in so many ways, for example, we can choose to return the string odd or even. Or we can return true or false, or we can return 1 or 0. And so on.
   a. In this case we will return 0 or 1. 0 if even, 0 if odd.
   b. Since we are returning 0 or 1 the type of the function will be integer.
3. Finally, we start writing the body of the function.

The function will be as follows:

```
int checkOddOrEven(int number)
  {
    if (number.isEven)
      {
        return 0;       }
    else {
        return 1;
  } }
```

Q2. Write a code to find the maximum value between three values.

Applying the same rules as before:

- We have three parameters. All of them are integers
- We have an output that is a number and its type will be integer

```
int maxOfThree(int a, int b, int c)
     {
        if (a >= b && a >= c)
          {
            return a;
          } else if (b >= a && b >= c){
            return b;
            } else {
           return c;
     } }
```

If we follow this code we can clearly see that it achieves the result we want, however, if the question was changed to 4 values or 5 or 6 the code will get messy very fast. This means we should rewrite this function in a more dynamic way.

```
int maxOfValues(List numbers){

  numbers.sort();

  return numbers.last;

}
```

This code is simple and more dynamic than the previous one.

Q3. You are given a map in Dart where the keys are student names (strings) and the values are lists of integers representing the marks obtained by each student in various subjects. Write a function calculateAverageMarks that takes this map as input and returns a new map where the keys are the student names, and the values are the average marks of each student.

```
Map<String, double> calculateAverageMarks(Map<String, List<int>> marks) {
  Map<String, double> averageMarks = {};
  for (var entry in marks.entries) {
    int total = 0;
    for (var mark in entry.value) {
      total += mark;
    }
    double average = total / entry.value.length;
    averageMarks[entry.key] = double.parse(average.toStringAsFixed(2));
  }
  return averageMarks;
}
void main() {
  Map<String, List<int>> studentMarks = {
    "Ali": [85, 92, 78],
    "Ahmed": [79, 88, 91],
    "Omar": [90, 85, 87]
  };
  Map<String, double> result = calculateAverageMarks(studentMarks);
  print(result);
}
```

Q4. You are given a map in Dart where the keys are product names (strings) and the values are lists of maps. Each map in the list represents a sale, with keys for "quantity" (an integer) and "pricePerUnit" (a double). Write a function calculateTotalRevenue that takes this map as input and returns a new map where the keys are the product names, and the values are the total revenue generated for each product.

```
Map<String, double> calculateTotalRevenue(Map<String, List<Map<String, dynamic>>>
sales) {

  Map<String, double> totalRevenue = {};

  for (var entry in sales.entries) {

    double total = 0;

    for (var sale in entry.value) {

      total += sale["quantity"] * sale["pricePerUnit"];

    }

    totalRevenue[entry.key] = double.parse(total.toStringAsFixed(2));

  }

  return totalRevenue;

}

void main() {

  Map<String, List<Map<String, dynamic>>> productSales = {

  "Apples":
 [{"quantity": 150, "pricePerUnit": 2.0},  {"quantity": 200, "pricePerUnit": 2.5}, {"quantity":
180, "pricePerUnit": 2.2}],
"Oranges":
[{"quantity": 120, "pricePerUnit": 1.5},{"quantity": 160, "pricePerUnit": 1.6},{"quantity":
140, "pricePerUnit": 1.4} ],
"Bananas":

[{"quantity": 100, "pricePerUnit": 1.2},{"quantity": 150, "pricePerUnit": 1.3},{"quantity":
130, "pricePerUnit": 1.1} ]};

  Map<String, double> result = calculateTotalRevenue(productSales);

  print(result);  }
```