1.Dart Programming Language - Lecture [1]

[In depth look at variables and the concepts regarding them]

1. Introduction

Variables are the corner stone of any program no matter how simple it is, and they can vary in complexity from describing simple things such as numbers and letters to very complicated objects like humans or any real life object. Mastering the concepts behind variables can help you understand the most fundamental skills of problem solving.

In this lecture we will go through different key points such as what it is a variable, what are the details behind creating a variable, scope of variables, life cycle of variables, naming standards, types of variables, variable usage and bad practices.

Learning Objectives

1. Understand the concept of Variables.
2. Be able to apply all variable standards.
3. Analyze variable definition.
4. Applying good practices.
5. Avoid bad practices.

2. Context

In programming, the ability to store, manipulate, and retrieve data is fundamental. Variables are the building blocks that allow programmers to perform these tasks efficiently.

They play a crucial role in controlling the flow of a program, such as through loops and conditionals.

Key Terms and Definitions

1. Variable: A name that references a value, and it contains a group of attributes and behaviors
2. Data type: The way you intend to use the variable.
3. Modifiers: Key words the modify the behavior of a variable.
4. Declaration: The process of defining the variable Data type and name
5. Initialization: The process of giving a variable an initial value
6. Assignment: The process of giving a variable a value
7. Scope: The area of code that the variable is accessible

8. Life Cycle: The value of the variable since its creation until the end of the program

3. Main Content

1. Variable Creation (Declaration + initialization):

To understand how to create a variable we need to understand 4 things first:
1. Data types 2. Modifiers 3. Naming of a variable 4. Value of a variable

[Data types]:

Data types are the way we limit the values we can assign to a variable and the can be extremely important, they ensure:

Data integrity, *Type Safety, easier debugging, code clarity, data validation, data exchange. (1)*

there are a lot of already defined data types, int for defining whole numbers, String for defining text values, List for saving multiple values. Defining a variable from a data type makes that variable accepts values from that type, not only that, it also gives the variable new attributes and behaviors that well be studied in next lectures.
We must know that we can define our own data types.

In dart we can use the var keyword to make the variable dynamic which means its type will be determined based on the value we assign to it. Using var often ***is not recommended***.

[Modifiers]:

Modifiers are a group of built-in Keywords that can be used when defining a variable to change its behavior, here are some of these key words (final, const, static, private, …). The purpose of these modifiers is:

Controlling accessibility, defining constants, overriding, … .(2)

modifiers are optional and are not required to create a variable but can be used when needed.

[Naming a variable]:

Naming a variable is a simple task yet many fail to do it properly, the name of the variable should always point to what this variable does, for example if you create a variable that describes a person called Ahmed the name of the variable should be Ahmed, so we should avoid calling variables x and y.

Another important point in the topic of naming a variable is to focus on the naming conventions of variables, in almost all programming languages variables are named following the camel case convention, we write the first word all small letters then the first letter of each word will be uppercase as follows, firstName, totalAmount.

We should avoid bad habits when naming a variable such as using abbreviations such as fN (firstName), or using a keyword (calling a variable myInt or myString).

There are names that cannot be allowed for a variable such as starting with a number, having a space, using a key word, including special characters, ….

Examples: 1Name, first Name, int, total+amount, -name, ….

[value of a variable]:

Assigning a value to a variable is done by using the (=) simple, and this value ***MUST*** match the data type of the variable, for example, if the variable was of the Boolean type we can only assign two values to it, True or False. If the variable was a string we can only assign a text to it like "Hi my name is Ali", this text ***MUST*** be inside a quotation marks.

After we understood all of the previous points now we can fully understand any code the creates a variable. Let's take a look at a few examples:

```
int age = 25; // Integer type variable
double height = 5.9; // Double (floating-point) type variable
String name = "Alice"; // String type variable
bool isStudent = true; // Boolean type variable
```

```
var age = 25; // Dart makes it 'int'
var height = 5.9; // Dart makes it 'double'
var name = "Alice"; // Dart makes it 'String'
var isStudent = true; // Dart makes it 'bool'
```

```
const double pi = 3.14159; // value cannot be changed
const String appName = "MyApp"; // value cannot be changed
```

2. Null Safety:

Null safety is a very important topic in dart, and it is not used in other programming languages, let's start by understanding what is null.

Null represents the absence of a value, which means nulls can lead to errors. This type of errors can be very common when using dart, we will learn more about these errors in the next lecture.

In dart we can use the (?) symbol to indicate if a variable is allowed to be null or not. If this symbol ***is not used a variable cannot be*** ***null,*** if it is used then the variable ***can be null.*** For example:

```
int age = null // this will cause an error because the variable cannot be null
```

```
int? age = null // this is accepted because we used (?)
int? age = 10 // this is normal
```

in conclusion when we use (?) after the data type of the variable then we can assign null to that variable, if we did not use (?) then it is not allowed for the variable to be null

3. Variable scope

The scope of a variable is the place where I can use the variable. There are two kinds of scopes:

- The global scope, here the variable is defined ***everywhere*** in the program and can be accessed anywhere
- The block scope, here the variable is only defined inside a block and can only be used in that block only, and if we tried to use this variable outside of its block, an error will occur.

[variable shadowing]:

Variable shadowing is a bad practice where we use the same name of a variable twice in different blocks. Even if this does not cause an error, because the variables are in different blocks but this creates a lot of confusion and should be avoided.

4. Variable life cycle

As a programmer this is one of the most important concepts we should follow, it makes debugging, problem solving, and analyzing the code so much easier. It is the simple process of following the changes that happens to a variable at each step since it is created until the program is over. ***This is one of the most useful practices that will use until the end of this course.***

4. Practical Examples

Practice 1: [assigning null]

Create a variable the is not followed by (?) and assign null to it, what is the error that shows?

Practice 2: [changing a constant]

Create a variable using the modifier const then change its value. What is the error that shows?

Practice 3: [applying naming conventions and standards]

Create 5 variables with the correct naming conventions and standards.

5. References

Further Reading:

(1): https://en.wikipedia.org/wiki/Data_integrity

(2): https://dart.dev/language