Basics of Dart

(Tiigsi) Mobile AppDev Course

What is Dart

Dart is a new language using C-style which is developed by Google. It appeared first time in 2007 and the first stable release was in Oct 2011. Dart is used for developing web, server, mobile applications and some IoT devices.

It is open-source object-oriented, staticly typed, and easy to learn programming language.

CONTENTS

Variables

A variable is "a named space in the memory" that stores values.

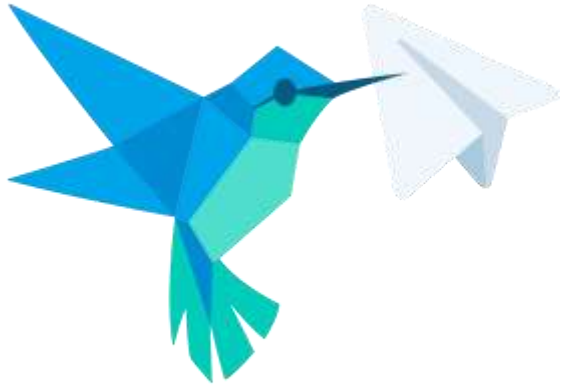
In other words, it acts a container for values in a program.

Variable names are called **identifiers**. These are the naming rules for an identifier:

1. Identifiers cannot be keywords.
2. Identifiers can contain alphabets and numbers.
3. Identifiers cannot contain spaces and special characters, except the underscore (_) and the dollar ($) sign.
4. Variable names cannot begin with a number.

Data Types

Like other languages (C, C++, Java), whenever a variable is created, each variable has an associated data type. These are the data type :



Data Type	Keyword	Description
Numbers	int, double	Represents numeric literals
Strings	String	Represents a sequence of characters
Booleans	bool	Represents Boolean values true and false
Lists	List	An ordered group of objects
Maps	Map	Represents a set of values as key-value pairs

Comments

In every programming language comments play an important role for a better understanding of the code. Comments are a set of statements that are not meant to be executed by the compiler.

Types of Dart Comments:

1. Dart Single line Comment.

```
// This is a single line comment.
```

2. Dart Multiline Comment.

```
/*
These are multiple line
of comments
*/
```

Operators

The operators are special symbols that are used to carry out certain operations on the operands.

These are the main dart operators:

1. Arithmetic Operators
2. Relational Operators
3. Assignment Operators
4. Logical Operators
5. Conditional Operator

1. Arithmetic Operators

This class of operators contain those operators which are used to perform arithmetic operation on the operands. They are binary operators i.e they act on two operands. They go like this:

Symbol	Operator Name	Description
+	Addition	Use to add two operands
–	Subtraction	Use to subtract two operands
*	Multiplication	It is Use to reverse the sign of the expression
/	Divition	An ordered group of objects
%	Modulus	Use to give remainder of two operands

2. Relational Operators

This class of operators contain those operators which are used to perform arithmetic operation on the operands. They are binary operators i.e they act on two operands. They go like this:

Symbol	Operator Name	Description
>	Greater than	Check which operand is bigger and give result as boolean.
<	Less than	Check which operand is smaller and give result as boolean.
>=	Greater than/Equal to	Check which operand is greater or equal to each other.
<=	Less than/Equal to	Check which operand is less than or equal to each other.
==	Equal to	Check whether the operand are equal to each other or not.
!=	Not Equal to	Check whether the operand are not equal to each other or not.

3. Assignment Operators

This class of operators contain those operators which are used to assign value to the operands. It goes like this:

Symbol	Operator Name	Description
=	Equal to	Check which operand is bigger and give result as boolean.
??=	Assignment Operator	Assign the value only if it is null

4. Logical Operators

This class of operators contain those operators which are used to assign value to the operands. It goes like this:

Symbol	Operator Name	Description
&&	And Operator	Use to add two conditions and if both are true than it will return true.
||	Or Operator	Use to add two conditions and if even one of them is true than it will return true.
!	Not Operator	It is use to reverse the result.

5. Conditional Operators

This class of operators contain those operators which are used to assign value to the operands. It goes like this:

Symbol	Operator Name	Description
condition ? expersion1 : expersion2	Ternary Operator	It is a simple version of if-else statement. If the condition is true than expersion1 is executed else expersion2 is executed.
expersion1 ?? expersion2	Null Check Operator	If expersion1 is non-null returns its value else returns expression2 value.

Control Flow

The control statements or flow of control statements are used to control the flow of Dart program. These statements are very important in any programming languages to decide whether other statement will be executed or not

These are the main dart Control Flow statements:

1. If -Else statement
2. Switch-case statement
3. Loops

If-else Statement

This type of statements simply checks the condition and if it is true the statements within it is executed but if it in is not then the statements are simply ignored in the code.

Syntax:

```
if ( condition ) {
    // body of if
}
```

Example:

```
int age = 10;
if (age > 3)  {
        print("Condition is true");
 }
```

Switch Statement

In Dart, switch-case statements are a simplified version of the nested if-else statements. Its approach is the same as that in Java.

Example:

```
String daytime = “Morning”;
switch (daytime) {
    case “Morning” : {
        print("It is morning");
    } break;
    case “Evening”: {
        print("It is evening");
    }
}
```

Loops

A looping statement in Dart or any other programming language is used to repeat a particular set of commands until certain conditions are not completed. There are different ways to do so.

Loops types are as below:

1. for loop
2. for... in loop
3. for each loop
4. while loop
5. do-while loop

Functions

Function is a set of statements that **take inputs, do some specific computation and produces output**. Functions are created when certain statements are repeatedly occurring in the program and a function is created to replace them. Functions make it easy to divide the complex program into smaller sub-groups and increase the code reusability of the program.

Function Syntax

```
return_type function_name ( parameters ) {
        // Body of function
        return value;
}
```

Classes & Objects

Dart is an object-oriented programming language, so it supports the concept of class, object ... etc. In Dart, we can define classes and objects of our own. We use the class keyword to do so.

Sytanx of Class Declaration

```
class class_name {
  // Body of class
}
```

Sytanx of Object Declaration

```
var object_name = class_name();
```

THANKS

Cabdifataax Cabdilaahi