

# Programming with B4X

## Lesson 4 – Variables and Range

Version 1.0, February 2021



Anywhere Software

## Lesson 4 - Variables and Range

 **3h**

### What students should know

- Explain how a computer stores data in ram.
- Explain what a variable is.
- How to name a variable
- Assign a value to a variable.
- Use Mathematical Operators
- Use log command to display a variable

Imagine that you live on a street with a few million houses all in a row; each house has as you all know its address which starts at number 1 and ends at the last house; in order to be able to locate a friend who lives on that street you need to know the number of the house; so we have on the one hand a house number and on the other the friend who lives in that house.



Figure 1 Computer Memory (<https://www.freepik.com>)

The computer's central memory works in much the same way. There are many houses each with its address and a "resident" within each house. This address is called memory address and the "resident" content. On the computer often the "resident" who from now on we will call variable needs more houses to fit.

For a programmer to use memory, he must know the data he needs and the type of data. These can be integers or real numbers, words or letters, some reasonable values (truth, false). He also needs a "home" in the computer memory to store them represented by address.

In B4X the data can be stored in different types as:

B4X	Type
Boolean	boolean
Byte	integer 8 bits
Short	integer 16 bits
Int	integer 32 bits
Long	long integer 64 bits
Float	floating point number 32 bits
Double	double precision number 64 bits



Char	character
String	array of characters

Table 1- Simple types of variables

Every type needs different space in memory to store values.

Because it is difficult for the developer to remember all the addresses of his data, each address corresponds to a name. Fortunately, in fact this is done by the programming language itself and all it takes is to think of a good name for his data. For example, a data that is an integer for age could be called "age". Now, there is a "home" called age in computer memory.



### Remember

Variables are used to store information to be referenced and manipulated in a computer program. They also provide a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves. It is helpful to think of variables as containers that hold information.

### How to find how many variables you need

In any programming problem that a developer encounters, they should be able to locate the data and the information's of the problem.

In programming we name all those elements that we need to know to move forward in solving a problem. Usually in a programming problem we find them in the pronunciation of the exercise with the help of keywords such as:

- Reads
- Registers
- Ask
- Accept
- Type

*Example 1: Write a program that converts the euros we type into dollars.*

*Example 2: Make a program that accepts a positive integer and calculates its square, cube, and square root.*

### Information's

Information's in programming are all the elements that we need to calculate after processing our data. We usually find them in pronunciation using keywords such as:

- Calculates
- Displays
- Writes
- Counts
- Convert



In the previous examples what are the requested?

## Naming Variables

The names of the variables in B4X must follow specific rules.

- They must start with a capital or small character.
- They can then have digits or the character underscore.
- B4X does not distinguish capital and small letters.

Also, it's a good practice to name variables beginning with 3 small letters indicating the kind of a variable and continue with 1 uppercase letter and a meaningful word. For example:

- Dim **intAge** as Int
- Dim **fltAmount** as Float
- Dim **strName** as String

This practice helps a lot when you find variables in the code to recognize the type and the value it stores.

## Declaring Variables

### My First Variable



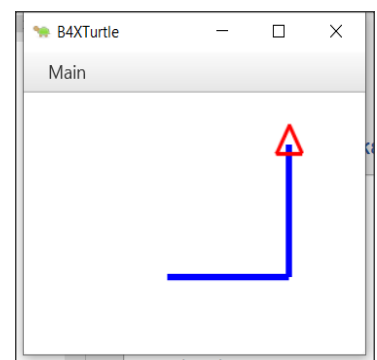
Make a program that assigns a value into integer and then draws with the help of the turtle a line of length as long as the value in the variable.

In B4X to use a variable we must first inform the language of its existence in order to commit space in the computer's memory to store its value.

For example, in the following code, the statement is as follows:

```
'Program: My First Variable
Sub Turtle_Start
  Private intDistance As Int
  Public intTurn As Int
  intDistance = 100
  intTurn = 90

  Turtle.SetPenColor(xui.Color_Blue).SetPenSize(5)
  Turtle.MoveForward(intDistance)
  Turtle.TurnLeft(intTurn)
  Turtle.MoveForward(intDistance)
End Sub
```



The declaration of variables begins with the keyword Private or Public.

Private means that the variable is known only in the specific space declared and no other program or subprogram does not know its existence and thus the value it contains.

Instead, a variable statement that starts with the keyword Public can be known to other programs or subprograms or classes etc.

After the keyword Private or Public follows the name of the variable we chose to give. This is where the rules discussed above apply. Finally, the type of the variable follows. For simple variables these are all those described in the *Table 1- Simple types of variables*.



#### Teachers tip

You don't have to explain all the variables already as well as their use. For your students to start programming, the basics of integer, float, string are enough. As you progress through the courses you can include other types according to your needs.

## Comments

In computer programming, a comment is a programmer-readable explanation or annotation in the source code of a computer program. They are added with the purpose of making the source code easier for humans to understand and are generally ignored by compilers and interpreters. The syntax of comments in various programming languages varies considerably. (Wikipedia, 2021)

In B4X comments are inserted by writing the character ' as their first letter. From this point on it is not recognized by the translator of the language. Generally, in B4X comments you should put anywhere it is important to remember what you are doing as well as before the subprograms to explain what their job is. Comments are easily distinguished in code from the green color given to them by the programming environment (IDE).

### Example

```
'Program: My First Variable
'This program draws a right angle, with sides as much as the
'value of the intDistance variable
Sub Turtle_Start
  Private intDistance As Int
  Public intTurn As Int
  intDistance = 100      'The sides of the right angle
  intTurn = 90           '90o angle

  Turtle.SetPenColor(xui.Color_Blue).SetPenSize(5)
  Turtle.MoveForward(intDistance)
  Turtle.TurnLeft(intTurn)
  Turtle.MoveForward(intDistance)
End Sub
```

## The log area and the log function.

During programming various errors occur. Generally, errors in programming are divided into two categories syntax and logical. For now, we will deal with the syntax errors that are recognized by the programming language and indicate them on the logs screen. In order to access the logs screen we need to click on the relevant logs tab at the bottom right. The Logs screen itself is divided into two frames, the first of

which displays errors and the bottom screen displays language messages or that information we want to display using the log() function. Using the Log() function helps the developer display messages while running a program as well as variable values to help control the program's proper operation.

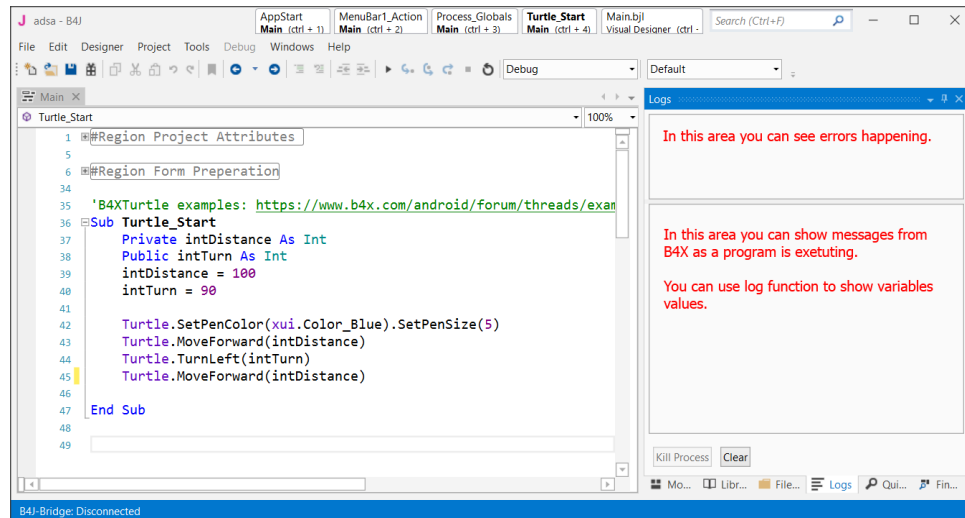


Figure 2 Logs Screen

To display any information on the screen it is sufficient to use the log() function as the example of the following image.

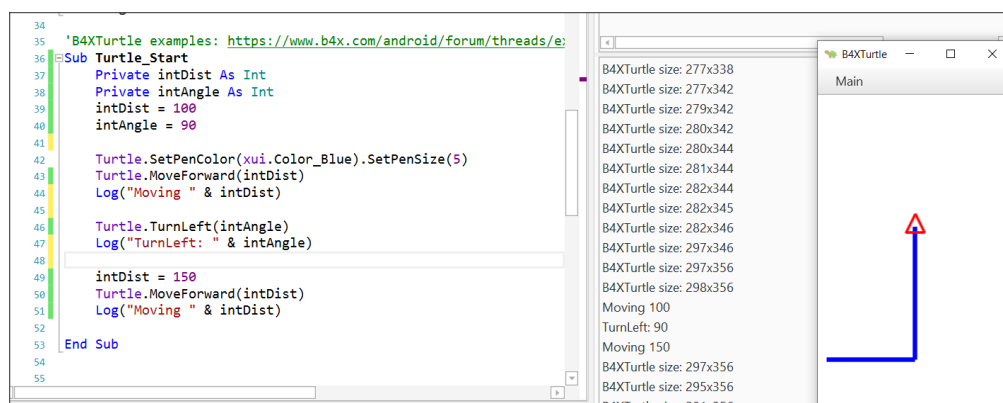


Figure 3 Using log function

## Mathematical Operators

B4X supports all known mathematical operations:

Operator	Example	Operation
+	$x + y$	Addition
-	$x - y$	Subtraction
*	$x * y$	Multiplication
/	$x / y$	Division
Mod	$x \text{ Mod } y$	Modulo
Power	$\text{Power}(x, y) \quad x^y$	Power of

## Examples:

```
Private intA, intB, intC, intS As Int
Private fltD, fltM As Float
intA = 40
intB = 20
intC = 30

intS = intA + intB + intC
Log(intS)                \Shows 90

fltD = intS / 3
Log(fltD)                \Shows 30

intA = intA + 1           \Increase intA by 1
Log(intA)                \Shows 41

intS = Power(intA - 11, 2) \ 302
Log(intS)                \Shows 900

fltM = intA mod 2         \41 modulo 2
Log(fltM)                \Shows 1
```

## Strings

In computer programming, a string is traditionally a sequence of characters, either as a literal constant or as variable. The latter may allow its elements to be mutated and the length changed, or it may be fixed (after creation) (Wikipedia, Wikipedia - Strings, 2021).

A string is declared like the other variables using the String statement

```
Private strName as String
```

Assigning value to a string can be done with the = symbol or by reading a value from the user (something we'll see later).

```
Private strName, strSurName as String
strName = "George"
strSurName = "Smith"
```

Also we can string together using the character &.

```
Private strName, strSurName as String
strName = "George"
strSurName = "Smith"
Private strPerson as String

strPerson = strName & " " & strSurName
log(strPerson)          \ shows George Smith in log screen

Private strName2 as String
strName2 = "John"
strName2 = strName2 & " Smith"
```

There are also a lot of functions regarding strings that makes our life easier when we are dealing with them:

<b>CharAt</b> (Index)	Returns the character at the given index.
<b>CompareTo</b> (Other)	Lexicographically compares the string with the Other string.
<b>Contains</b> (SearchFor)	Tests whether the string contains the given SearchFor string.
<b>EndsWith</b> (Suffix)	Returns True if the string ends with the given Suffix substring.
<b>EqualsIgnoreCase</b> (Other)	Returns True if both strings are equal ignoring their case.
<b>Length</b>	Returns the length, number of characters, of the string.
<b>Replace</b> (Target, Replacement)	Returns a new string resulting from the replacement of all the occurrences of Target with Replacement.
<b>StartsWith</b> (Prefix)	Returns True if this string starts with the given Prefix.
<b>ToLowerCase</b>	Returns a new string which is the result of lower casing this string.
<b>ToUpperCase</b>	Returns a new string which is the result of upper casing this string.
<b>Trim</b>	Returns a copy of the original string without any leading or trailing white spaces.

Table 2 String Functions (<https://www.b4x.com/android/documentation.html>)



#### Teachers tip

You can find more information about string manipulation in language booklets at (<https://www.b4x.com/android/documentation.html>)

## Exercises

1. In the following exercises, identify the variables you need to declare. For each of them, write the relevant statement and give it an appropriate name.
  - Calculate the volume of a cylinder with a radius of one metre and a height of two metres.
  - Make a program that accepts a positive integer and calculates its square, cube, and square root.
  - Make a program that reads a sum of money in € and calculates and displays the corresponding amount in \$.
  - Write a program that reads the length of the sides of a rectangle from the keyboard and calculates and displays its area.
  - The total resistance  $R$  of two resistances  $R_1$  and  $R_2$  connected in series is  $R_1 + R_2$  and parallel  $(R_1 * R_2) / (R_1 + R_2)$






respectively. Make a program that it reads two values of resistant R1 and R2 and calculates the total resistance in series and parallel.

2. In the following variable names, select which are correct and which are not:  
intAge

Name	True	False
int Age	<input type="checkbox"/>	<input type="checkbox"/>
_fltAmount	<input type="checkbox"/>	<input type="checkbox"/>
strName	<input type="checkbox"/>	<input type="checkbox"/>
1myAge	<input type="checkbox"/>	<input type="checkbox"/>
int_value	<input type="checkbox"/>	<input type="checkbox"/>

3. It's the end of the semester and you got your grades from three classes: Geometry, Algebra, and Physics. Create a program that: gives in 3 variables the grades of these 3 classes (Grades range from 0 - 10) Calculate the average of your grades.
4. You have bought a Bitcoin and now it's on the rise!!! Create a program that:
- Assign the value of the bitcoin at the time of purchase.
  - Assign the percentage of increase (or decrease)
  - Logs the total value of your bitcoin.
  - Logs the increase or decrease value.
5. You now own some property, and you want to calculate the total area of the property. Create a program that:
- Assign the width and height in two variables.
  - Calculate and log the area.
6. You are interested in buying a new laptop. You check the price and you see that the price is 300\$ without the 10% tax. Create a program that:
- Assign the the price of the laptop in a variable.
  - Assign the tax percentage in a second variable.
  - Calculate and logs the total amount.
7. In a company the monthly salary of an employee is calculated by the minimum wage 400\$ per month, plus 20\$ multiplied by the number of years employed, plus 30\$ for each child they have. Create a program that:
- Assign the number of years employed in a variable
  - Assign the number of children the employee has in second variable.



- 
- Calculate and logs the total amount of salary the employee makes.
8. Create a program that log the last digit of a given integer.
  9. Create two variables a and b, and initially set them each to a different number. Write a program that swaps both values.

Example: a = 10, b = 20

Output: a = 20, b = 10

10. Create two variables 'a' and 'b', and initially set them each to a different number. Write a program that double the Value of 'a' variable and increase the value of 'b' by 1.

Example: a = 10, b = 20

Output: a = 20, b = 21