**Programando con B4X**

Tema 4 – Variables y Rango

Version 1.0, marzo 2021

Autor original: [Prokopis Pliroforikos](https://github.com/pliroforikos)

Traducido al español por [LaMashino](https://github.com/Lamashino)

[](https://creativecommons.org/licenses/by/4.0/)

# Tema 4 - Variables y Rango

* Explicar cómo un ordenador almacena datos en la RAM
* Explicar qué es una variable
* Cómo nombrar una variable
* Asignar un valor a una variable
* Usar operadores matemáticos
* Usar el comando Log para mostrar una variable

Lo que los estudiantes aprenderán

**3h**

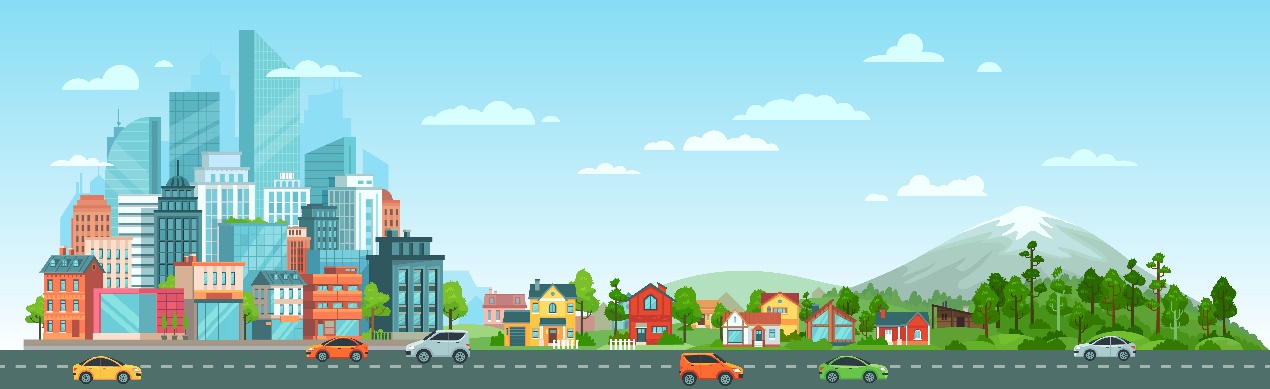
Imagínese que vive en una calle con varios millones de casas seguidas; cada casa tiene, como todos saben, su dirección que comienza en el número 1 y termina con el número de la última casa; para poder localizar a un amigo que vive en esa calle es necesario conocer el número de la casa; así que tenemos por un lado un número de casa y por otro el amigo que vive en esa casa. 

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

La memoria central de la computadora funciona de la misma manera. Hay muchas casas cada una con su dirección y un "residente" dentro de cada casa. Esta dirección se denomina **dirección de memoria** y al “residente” se le llama contenido. En el ordenador muchas veces el "residente" (que llamaremos **variable**) necesita varis casas para poder tener hueco.

Para que un programador utilice la memoria, debe conocer los datos que necesita y el tipo de datos. Estos pueden ser números enteros o reales, palabras o letras o valores lógicos (verdadero o falso). También necesita un "hogar" en la memoria de la computadora para almacenarlos representados por la dirección.

En B4X los datos se pueden almacenar en diferentes tipos como:

|  |  |
| --- | --- |
| **B4X** | **Tipo de dato** |
| Boolean | Booleano o lógico (verdadero/falso) |
| Byte | Entero de 8 bits (de 0 a 255) |
| Short | Entero de 16 bits (0 a 65535) |
| Int | Entero de 32 bits (0 a 4.294.967.296) |
| Long | Entero largo de 64 bits (0 a 264) |
| Float | Número decimal de 32 bits |
| Double | Número decimal de doble precisión de 64 bits |
| Char | Carácter (‘a’, ‘1’, ‘%’, etc.) |
| String | Cadena de caracteres (“hola”, “adiós”, “juan”,…) |

Tabla 1- Tipos básicos de las variables

Cada tipo necesita un espacio diferente en la memoria para almacenar su contenido.

Debido a que es difícil para el desarrollador recordar todas las direcciones de sus datos, a cada dirección se le pone un nombre. Por suerte, esto lo hace el propio lenguaje de programación y todo lo que se necesita es pensar en un buen nombre para tus datos. Por ejemplo, un dato que sea un número entero para la edad podría llamarse "edad". Ahora, hay un "hogar" llamado edad en la memoria de la computadora.

**Recuerda**

Las variables se utilizan para almacenar información para ser referenciada y manipulada en un programa. También proporcionan una forma de etiquetar los datos con un nombre descriptivo, para que el lector y nosotros mismos podamos entender nuestros programas con mayor claridad. Es útil pensar en las variables como contenedores que contienen información.



## Cómo averiguar cuántas variables necesitas

En cualquier problema de programación que encuentre un desarrollador, debería poder ubicar los datos y la información del problema.

En programación debemos dar un nombre a todos aquellos elementos que necesitamos conocer para resolver un problema. Normalmente en un problema de programación los podemos extraer del enunciado del ejercicio con la ayuda de verbos clave como:

* Leer
* Registrar
* Preguntar
* Aceptar
* Teclear

*Ejemplo 1: Escribe un programa que convierta los euros que escribimos en dólares.*

*Ejemplo 2: Haga un programa que acepte un número entero positivo y calcule su cuadrado, cubo y raíz cuadrada.*

**Información para mostrar**

La información que queremos mostrar está formada por todos los elementos que necesitamos calcular después de procesar nuestros datos. Normalmente los encontramos en el enunciado del problema con verbos clave como:

* Calcular
* Mostrar
* Escribir
* Contar
* Convertir

¿Qué debemos mostrar en los ejemplos anteriores?

## Cómo dar nombre a las Variables

Los nombres de las variables en B4X deben seguir estas reglas:

* Deben comenzar con mayúscula o minúscula.
* Después, pueden tener dígitos o el carácter de subrayado (\_).
* B4X no distingue entre mayúsculas y minúsculas.

Además, es una buena práctica poner 3 letras delante del nombre de la variable indicando el tipo de variable que es y continuar con 1 letra mayúscula y una palabra significativa. Por ejemplo:

* Dim **intEdad** as Int
* Dim **fltCantidad** as Float
* Dim **strNombre** as String

Esto te ayudará mucho cuando encuentre una variable en el código pudiendo así reconocer el tipo y el valor que almacena.

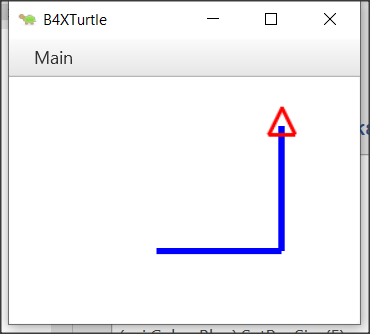
## Declaración de Variables

### Mi Primera Variable

Escribe un programa que asigne un valor a un número entero y luego dibuje con la ayuda de la tortuga una línea de longitud tan larga como el valor de la variable.

En B4X, para usar una variable, primero debemos informar al lenguaje de su existencia para que le asigne espacio en la memoria del ordenador para almacenar su valor.

Por ejemplo, en el siguiente código, la declaración es la siguiente:

‘Programa: Mi Primera Variable

Sub Turtle\_Start

Private intDistancia As Int

Public intGiro As Int

intDistancia = 100

intGiro = 90

Turtle.SetPenColor(xui.Color\_Blue).SetPenSize(5)

Turtle.MoveForward(intDistancia)

Turtle.TurnLeft(intGiro)

Turtle.MoveForward(intDistancia)

End Sub

La declaración de variables comienza con la palabra clave **Private** o **Public**.

**Private** significa que la variable se conoce solo en el espacio específico declarado y ningún otro programa o subprograma conoce su existencia y, por lo tanto, el valor que contiene.

En cambio, una declaración de variable que comienza con la palabra clave **Public** puede ser conocida por otros programas, subprogramas o clases, etc.

Después de la palabra clave **Private** o **Public** sigue el nombre de la variable. Aquí es donde se aplican las reglas discutidas anteriormente. Finalmente, sigue el tipo de variable. Para variables simples, los tipos son todos los descritos en el *Tabla 1-* Tipos básicos de las variables.

**Consejo para el profesor**

No es necesario que explique todas las variables ni su uso. Para que sus estudiantes comiencen a programar, los conceptos básicos de integer, float, string son suficientes. A medida que avanza en los cursos, puede incluir otros tipos según sus necesidades.



## Comentarios

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.

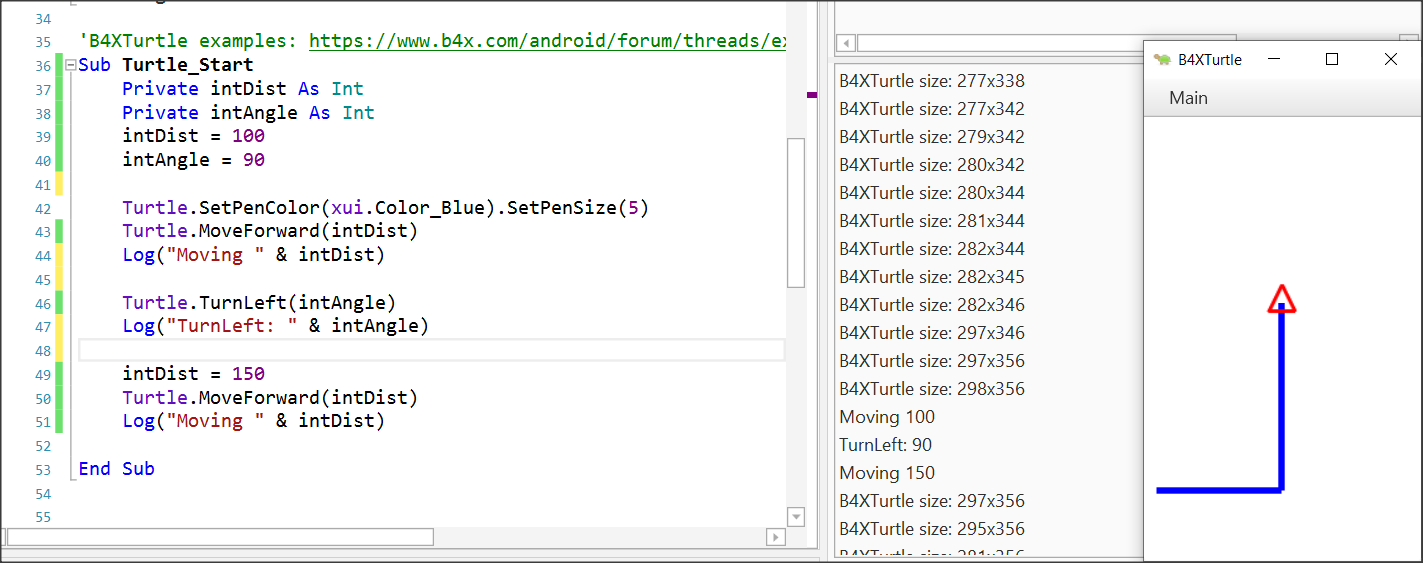


Figure 3 Using log function

## 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 Mod y | Modulo |
| Power | Power(x,y) xy | 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 = intAa + 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”

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

|  |  |
| --- | --- |
| **CharAt**(Index) | Returns the character at the given index. |
| **CompareTo**(Other) | Lexicographically compares the string with the Other string. |
| **Contains**(SearchFor) | Tests whether the string contains the given SearchFor string. |
| **EndsWith**(Suffix) | Returns True if the string ends with the given Suffix substring. |
| **EqualsIgnoreCase**(Other) | Returns True if both strings are equal ignoring their case. |
| **Length** | Returns the length, number of characters, of the string. |
| **Replace**(Target, Replacement) | Returns a new string resulting from the replacement of all the occurrences of Target with Replacement. |
| **StartsWith**(Prefix) | Returns True if this string starts with the given Prefix. |
| **ToLowerCase** | Returns a new string which is the result of lower casing this string. |
| **ToUpperCase** | Returns a new string which is the result of upper casing this string. |
| **Trim** | 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 R1 and R2 connected in series is R1 + R2 and parallel (R1\*R2)/(R1+R2) respectively. Male 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 | □ | □ |
| \_fltAmount | □ | □ |
| strName | □ | □ |
| 1myAge | □ | □ |
| int\_value | □ | □ |

1. 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.
2. 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.

1. 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.
2. 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.
3. 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.
4. Create a program that log the last digit of a given integer.
5. 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

1. 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