Storage

In order to operate with the data, we must previously reserve memory spaces in which to store it. These are what we call **data storage structures**.

These structures must be defined by:

- **Identifier**: name that we give to the data during the program to refer to it.
- **Type**: nature and range of values that can be stored (number, text, boolean...)
- **Value**: specific content of the data at that moment.

In this course we are going to learn about only two structures, variables and vectors (also called arrays or lists in Scratch), although we are only going to focus in depth on the first one.

Variables and constants

They are spaces that store **a single DATA** of a different nature, and to which a name that identifies it is associated. The type of data they can store are:

- **Numeric**: they can be integers (int) or real (float), depending on whether or not they accept decimals, generally with Anglo-Saxon notation where the decimal is marked with a dot.
- **Characters** (char): letters or typographic signs. They are defined by enclosing the character with a single quote.
- **Character strings** (string): words or phrases. They are defined by enclosing the text with double quotes.
- **Logical** (Boolean): Supports values of TRUE or FALSE.

Variables, as their name indicates, are data structures that will take **different values** throughout the execution of the program, while **constants** adopt the **same value** throughout the program (For example, the PI number, or the percentage of the VAT).

It is very important to properly select the name of the variables and constants so that they help understand the stored data. Generally, **constants** are defined with names in **uppercase** and **variables** in **lowercase**.

To learn more about how it works, we are going to create the following program. We will call it SALES CALCULATOR.

Program: Sales calculator

Statement:

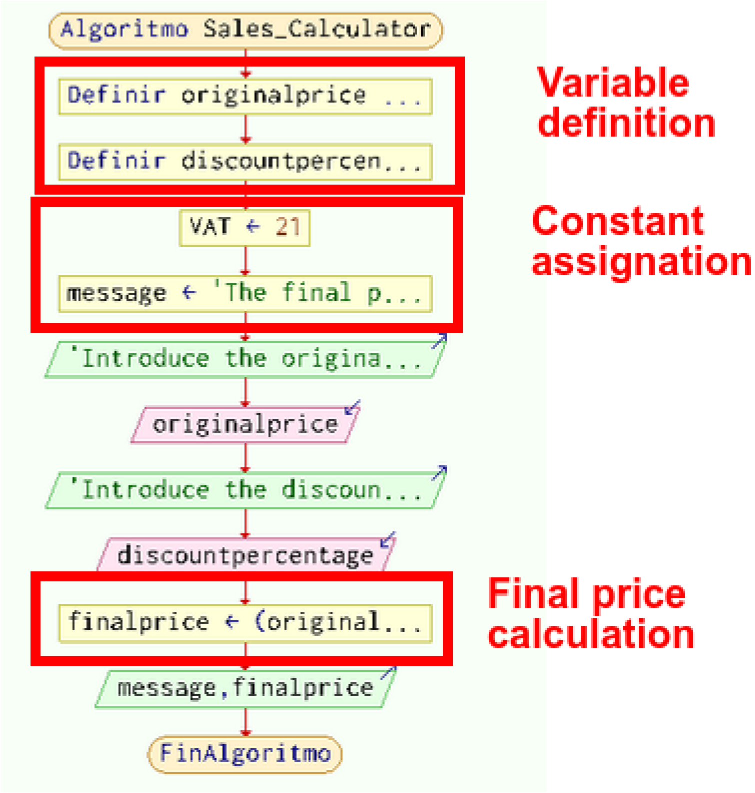
Create a program that requests from the user a price and an offer percentage. Later, based on a VAT percentage (Constant), the program shows a message with the total price to pay, VAT included.

Steps 1 and 2: Analysis and flow chart of the *Sales calculator* program

The elements involved will be:

- **Start** and **end** of program.
- **Outputs**: Request original price and discount percentage, and show the final price.
- **Inputs**: Original price, discount percentage
- **Storage**: original price (real number because it can be decimal), discount percentage (integer number because we will use the value as a percentage), final price (real number), VAT percentage (constant), and optionally the message to display (constant)
- **Processing**: addition, multiplication and division.

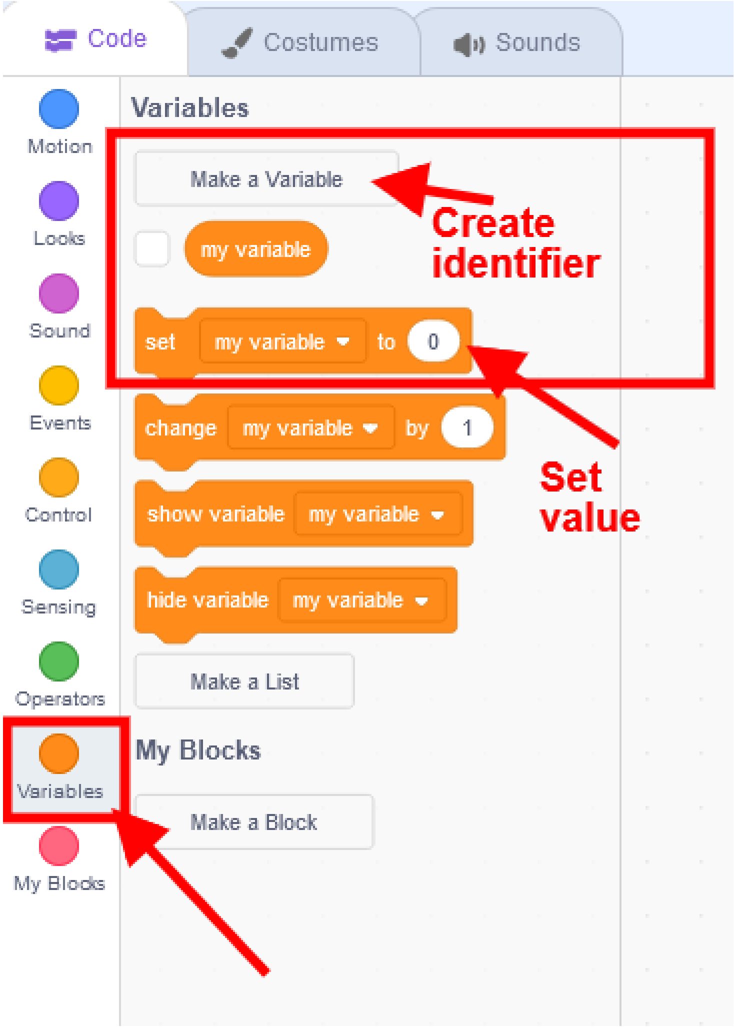
The flowchart is:


Algoritmo Sales_Calculator
Definir originalprice ...
Definir discountpercen...
VAT ← 21
message ← 'The final p...
'Introduce the origina...
originalprice
'Introduce the discoun...
discountpercentage
finalprice ← (original...
message,finalprice
FinAlgoritmo
Variable definition
Constant assignation
Final price calculation

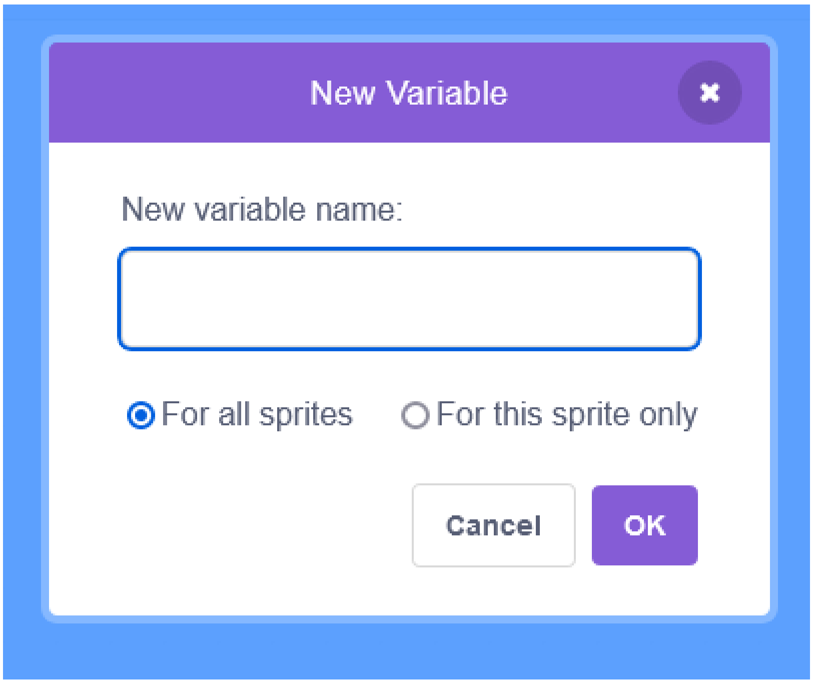
Variables could also be assigned initial values, although they will be modified throughout the program. If this is not done, 0 or empty is assumed as the initial value.

Steps 3, 4 y 5: Coding, compilation and checking of the *Sales Calculator* program with Scratch

In Scratch, data storage structures are located in the **Variables** block. From there we can create as many variables as we need, and also assign them the desired value using the corresponding block.



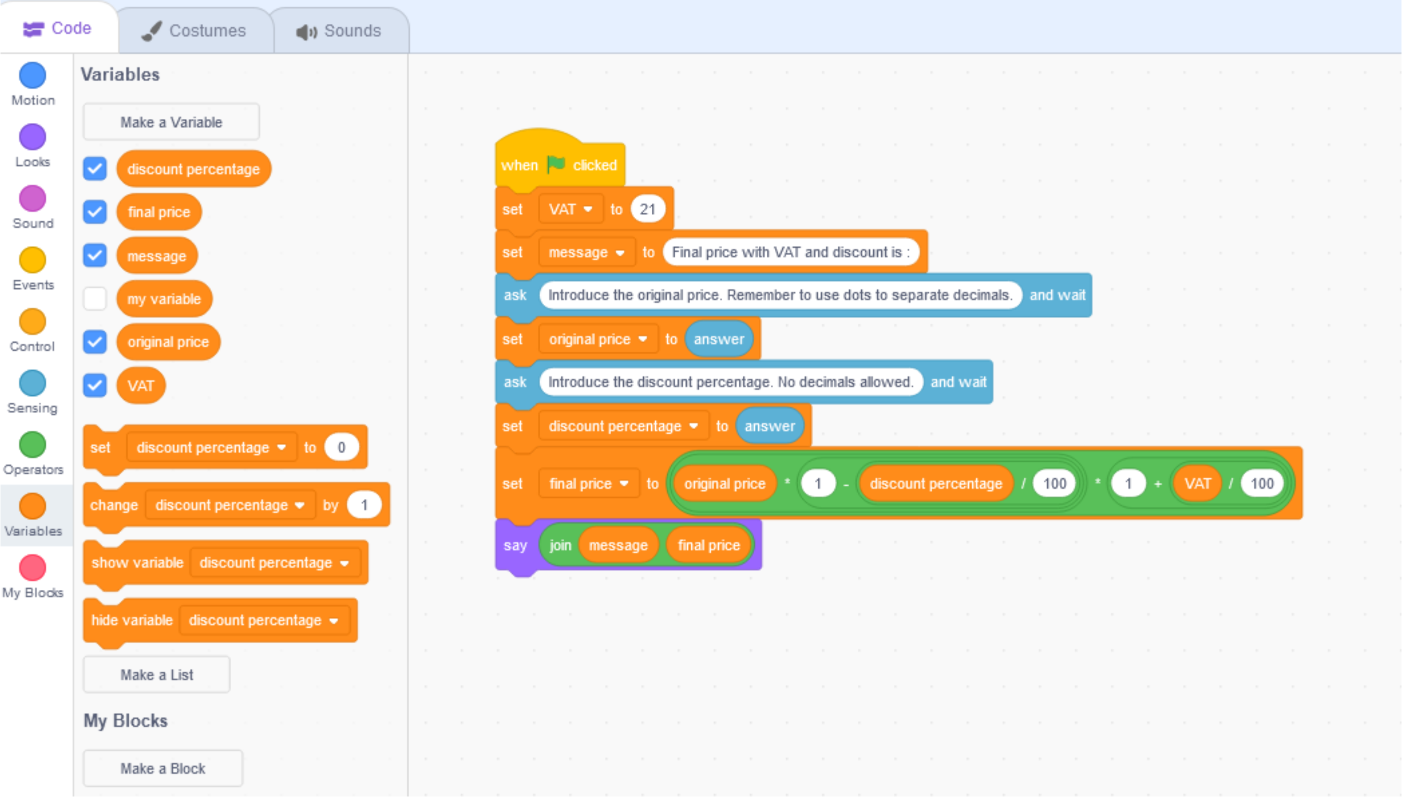
When creating a variable, the first thing it asks us for, apart from its identifier, is to know if it is defined as **local** (only for this sprite) or as **global** (common for all sprites). In this course we only develop programs for the same object, so it doesn´t matter.



New variable name:

● For all sprites ○ For this sprite only

Once the necessary variables have been defined, we will go on to create the code in the object using **sensing, looks** and the arithmetic and concatenation **operators** that we need, and which we will see in more detail in the next section. The code in blocks would look like this:



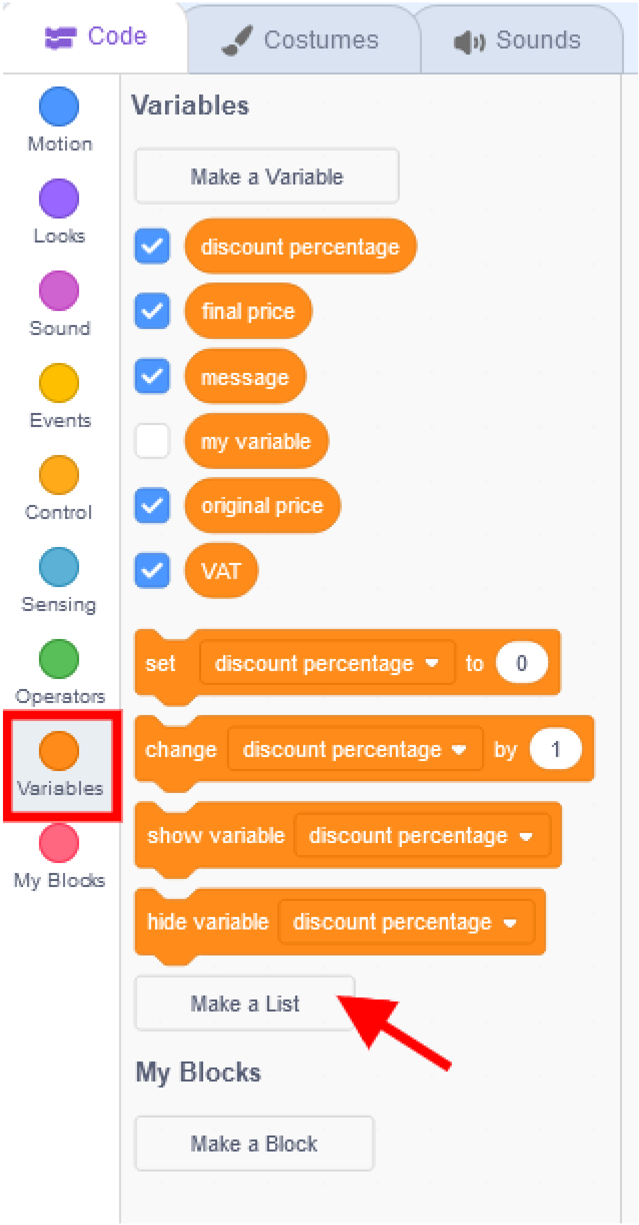
```
when 🏁 clicked
set VAT ▾ to 21
set message ▾ to Final price with VAT and discount is :
ask Introduce the original price. Remember to use dots to separate decimals. and wait
set original price ▾ to answer
ask Introduce the discount percentage. No decimals allowed. and wait
set discount percentage ▾ to answer
set final price ▾ to ((original price * (1 - (discount percentage / 100))) * (1 + (VAT / 100)))
say join message final price
```

*Save the file with the name **salescalculator.sb3.***

Complex data storage structures

Although it is not the subject of this course, in programming **data sets** can be stored as long as they are of the **same type** in other types of complex structures, the so-called **arrays**, which in Spanish are translated as vectors or matrices, and in PSeInt they are called **dimensions**.

Scratch only contemplates the use of **lists** that would be ordered series of data of the same type, and that are available within the same **Variables** block.



Revision #9
Created 15 October 2023 18:58:12 by Ana López Floría
Updated 25 January 2025 13:36:47 by Ana López Floría