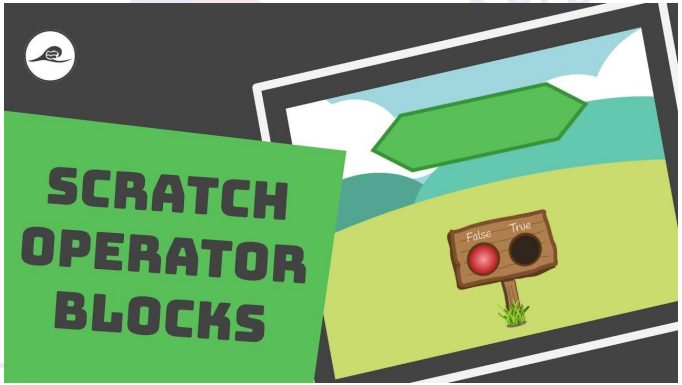
Lesson 16 – Operator Blocks



What we will Learn

We have seen the scoreboard in an IPL match. Some have great features, and some are controlled manually.

Have you wondered how they are controlled automatically?

This is done by code.

At its very simple format, the code will control the numbers & alphabets in the scoreboard that need to be added and updated regularly.

In this lesson, we shall learn about Operator blocks that can handle data in numeric & alphabetic form to enable this.

What are Operator Blocks

Operators blocks allow us to compare variables and values, do calculations with numbers, and work with strings (text).

This enables them to:

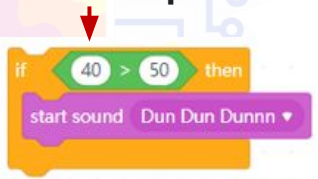
- Handle nums for mathematical & other operations.
- Handle text (strings).
- Mix of numbers & strings.
- Specify conditions for evaluation.
- & more...

They are unlike other blocks. They **Piggy Back** on them.

In so doing:

- They 

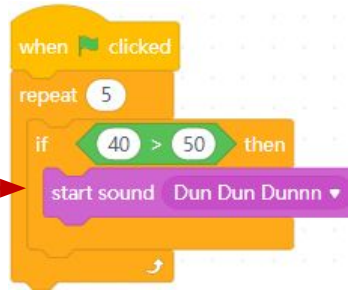
- Are filled with a value & placed inside

other blocks. 

- Thus creating the required block statement
- The number of such statements we can create is infinite.

The statement thus created is then used like other block statements.

It is stacked as a code line, in a code requiring them.



None of them can be stacked on their own.

We have 18 of them. They are colour coded light green. 

Categories of Operator Blocks

Operators blocks can be divided into two categories:

- Reporter blocks. 

- Boolean blocks. 

Reporter Blocks

Think of reporter blocks as a container holder that holds:

- Numeric values (integers & floats or decimals). 

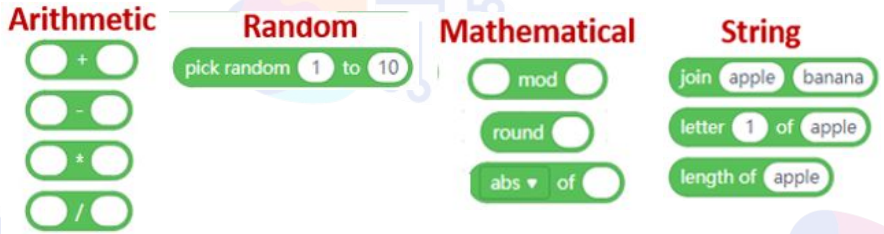
- String (text) values. 

As and when required, these stored values are taken out for use in the code.

They are oval in shape. 

Types of Reporter Blocks

Blocks are divided into four functional categories.



Boolean Blocks

The hexagonal 'Boolean' blocks are used for making conditions, that return true or false. These blocks can be used in Control blocks with a hexagonal input. Think of Boolean blocks as a container holder that holds:

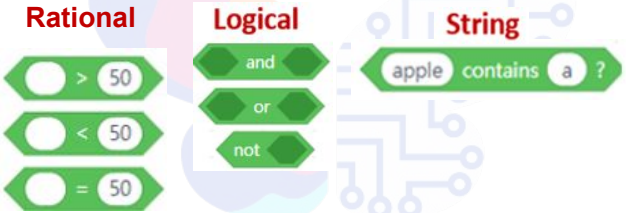
- Conditions for evaluation. 

- Output of such evaluation can either be True or False. Nothing else.
- Used with control blocks, they enable code to take choice based decisions.

They are hexagonal in shape. 

Types of Boolean Blocks

Boolean blocks are divided into three functional categories.



In this lesson, we shall now see details of Arithmetic operators.

The rest we shall see in level 2.

Arithmetic Operators

They carry out simple addition, subtraction, multiplication & division.

They need to be made. Let us now see a few examples of:

- How they are made.
- How they are used in a code.

Project 39. Simple Addition of two Numbers

Story: Add 84 & 37219.

Procedure:

- In operator block select statement: 

- In the roundels enter the nums to be added. 

Note the size of the roundel keeps increasing as per the number.

- Place this reporter block in a stack block by moving it from left towards right.

- Block statement is ready.

Add trigger & run the code to get the answer

Practice a few additions using integer & decimal nums.
In same way practice:

- Substation of two nums.
- Multiplication of two nums.
- Division of two nums.

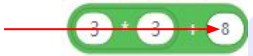
Project 40. Calculations under BODMAS rule

Story: Calculate (3 x 3) + 8.

Doing so in Scratch has three steps:

- **Take this block into script area.** 

  Place  **in the first roundel.** 

  Enter 8 in the second roundel. 

In other words (3 * 3) acts as bracket.

This then enables calculation under BODMAS rule.

Now try the same using other arithmetic operators.

Project 41. Addition of Multiple Numbers

Say we want to add five numbers - 10, 20, 30, 40 & 50.

Drag  **into script area.**

Entre 10 in 1st roundel. 

Now in 2 steps, create an operator block to enter the other four numbers.

- **Step 1.**

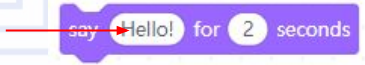
- **Step 2.**

This way the calculations could go into any number of additions

Now place this big block 

in second roundel of 

to get the final block. 

This is then put in first roundel of 

To make the final statement. 

Add a trigger & run it to get 150. 

Now you can use this block as:

- **Part of a any project code that requires it.**
- **To do big calculations, simply by punching in the new numbers.**
- **There is no limit to the number of additions you can do.**

A typical daily life use of this is, the Sum utility, we use so very often.

Project 42. Consolidation of Procedure for use of Arithmetic Operators

The story line of this project is: “Maths teacher calls one student forward. She asks him to tell the table of 18. The boy thinks for a while. He recites the table correctly. The entire class claps”.

In this project, besides doing maths we have added an important function called Broadcast with the aim of teaching how to make projects interactive.

Analysis of the Project

The project requires:

- **Background of a Class.**
- **Two sprites – Teacher & a Boy.**

Both need to take some actions, thus, both will need to be coded.

When coding more than one sprite at a time, both must be aware of what the other is saying or doing, & at what time, or in what sequence.

- This is done through **Broadcast Messages**.
- For this, all must remain on a **listening mode.**
- As & when they receive a broadcast message, this broadcast message **Acts as a Trigger** to take the next action.

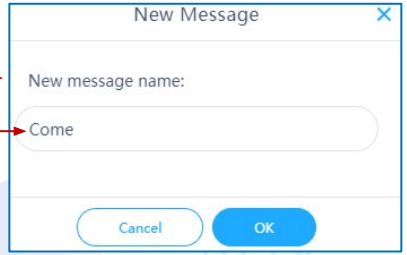
For this we have three broadcast statements in Event block.

Procedure to make the Code for the Teacher

Step 1: Make the statement of the teache

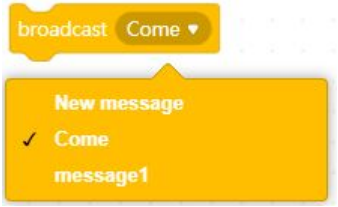
Step 2: Create a trigger for the boy.
To do this, take the event block to script area.

In its dropdown select New massage.
New message entry window opens:



Enter any one word from step 1 statement (ex come).

It gets added to the dropdown.



As the teacher will be having two statements
we must add the wait block
This will avoid voice bubble problem.
Add broadcast block & wait.
Add second Statement of teacher.
Create & add its broadcast block.
Note here we have used just one alphabet (T).

Code of the teacher is now complete.

Making the code for the Boy.

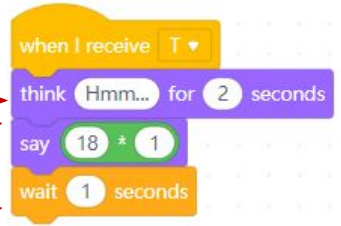
This has two parts:
In part 1 the boy is initially hidden. He comes forward and appears on the stage when called by the teacher. Thus, for this part we need:

- **A code to remain hidden. The trigger for this is the trigger for the project, because at the start of the project he is hidden.**

- **A code to appear. The trigger for this is the broadcast message from teacher to come forward.**

Now comes the second part of the boys code – His reciting the table. It consists of three parts:

1. The trigger which is the broadcast message from the teacher to tell us the table.
1. The second is to show the boy thinking (optional).
2. The third part is his reciting the first line 18x1. This must be followed by wait to avoid bubble problem.

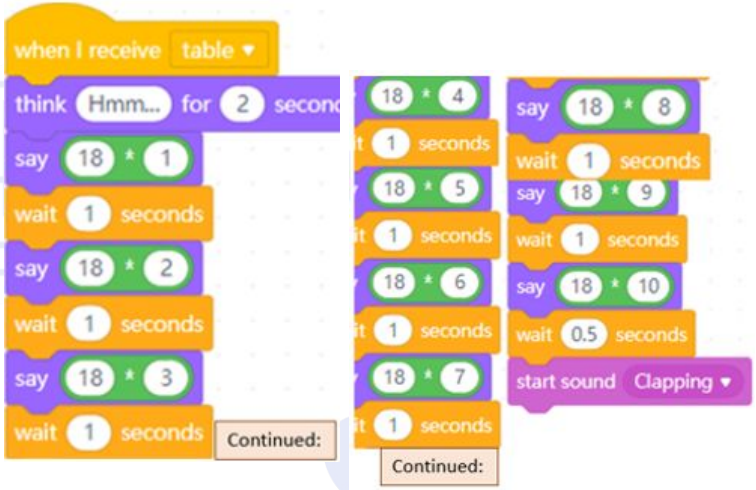


Thereafter, it is a repeat of these two lines till the end of the table.
You can use the Duplicate utility for doing this.

The code is as shown in these three columns.

Continued: has been used purely for depiction purposes, to guide you to put this part of code stack, over the next while making your own.

Code & see the result yourself.



You can now make, a table of any number including a decimal number, or up to any number, or from any number to any other number.

This is an important capability for later use.

Try doing some yourself.

Use of Scratch Based Coding to Learn Maths

For most children maths is a difficult subject to learn. Coding offers an excellent tool to schools to teach maths, as well as to students to learn more, the fun way.

Normally children start learning maths in grades three upwards. By this time they can easily learn Scratch.

Having done so, now learning maths will not only become easy, but they will also learn the real concepts & imaginative tricks around it, which otherwise they will never learn.

Take Aways

- Operators are the most versatile blocks of Scratch.
-
- They piggyback on other blocks.
-
- Their key use is for:
  - Doing arithmetic & mathematical calculations.
  - Carrying out logical & relational comparisons.
  - Evaluating conditions as true/false
  - Introducing children to concept of Strings in coding.

End of Lesson 16

Code Karega India Badhega