

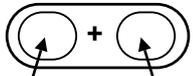
THINKING with the SCRATCH Operators (1)

<https://scratch.mit.edu/projects/477516868>

MATHS with SCRATCH
puts Magic into Maths

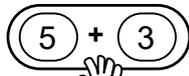
Maths Operators in SCRATCH report the result of a Maths Operation.

This is the green **plus** operator



input windows

The operation is **plus**



8

How to Code addition in SCRATCH.

plus and addition mean the same thing

Open SCRATCH. Open the Operators palette and drag out the *plus* operator. Type a low number in each input window.

Click the block to make it report its value. The block **REPORTS 8**

A. Write the missing value in each *plus* Operator's blank input window.

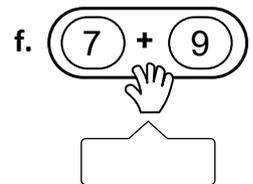
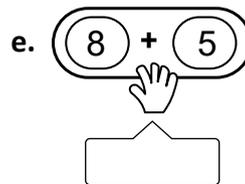
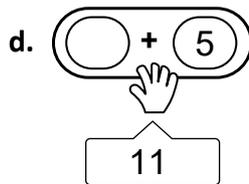
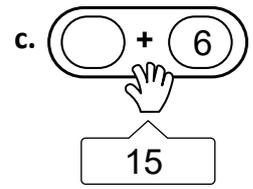
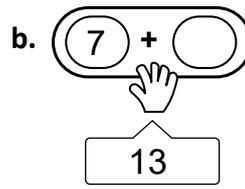
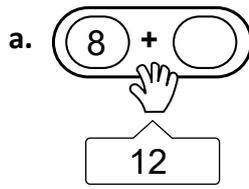
On paper you are used to seeing these like this:

$$\boxed{7} + \boxed{2} = \boxed{9}$$

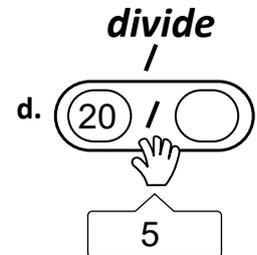
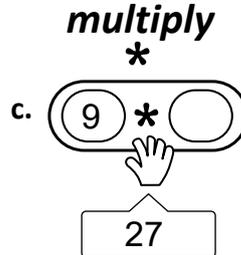
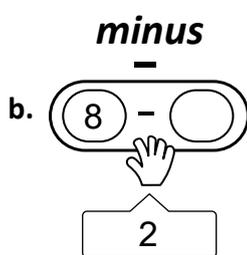
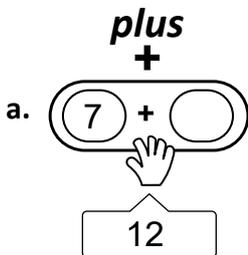
Here is the same thing in Scratch:



9



B. There are four Maths operators. Write the missing value in each SCRATCH input window.



C. Let SCRATCH think of a number between 1 and 10.

Drag out the *pick random* operator and click the block as it is.

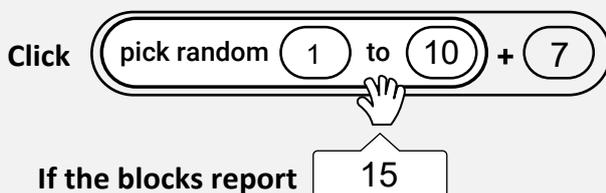


Click *pick random* six times, to get used to what this block does.

Write the six numbers above.

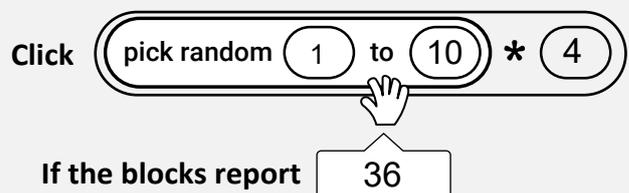
Combine *pick random* with the *plus* operator

PUZZLE 1. Drag *pick random* into the first input window of the *plus* operator. Type 7 in the second input window.



then, which number did SCRATCH pick ?

PUZZLE 2. Use the *multiply* operator.



then which number did SCRATCH pick ?

THINKING with the SCRATCH Operators (continued)

D. Combine an Operator with the say () block.

Click the block to make the sprite **say** its value.
You could even combine it with the Text-to-speech extension (to add the sound of speech).

Experiment in Scratch: What is each missing value ?

a. plus +

Code and Click to make the sprite **say** the value of 14.

Use *text-to-speech* if you like

b. minus -

What missing value will make the sprite say 6 ?

c. multiply *

What missing value will make the sprite say 32 ?

d. divide /

What missing value will make the sprite say 5 ?

e. minus -

What missing value will make the sprite say 8 ?

f. plus +

What missing value will make the sprite say 33 ?

E. SCRATCH adds before it multiplies

Two Operators in a say () block.

What missing number will make the sprite say 40 ?

Example 1
Drag the *multiplication* block into the input window of the *plus* block.

Click to make the sprite say 16.

Example 2
Drag the *plus* block into the input window of the *multiplication* block.

Click to make the sprite say 42.

In written work we have to use brackets to force *addition* before *multiplication*:

$$(2 + 3) \times 8 = 40$$

SCRATCH operators are like brackets. Without brackets, the value of . . .

$$2 + 3 \times 8 = 26$$

... because in the *order of operations*, *multiplication* is done before *addition*. This is how you CODE for 26 in SCRATCH.

You would put the multiply block inside the add block

Try it out and prove it for yourself. Instead of putting *plus* into a *multiply* window you must drag *multiply* into an *plus* window. Click and see the different value.

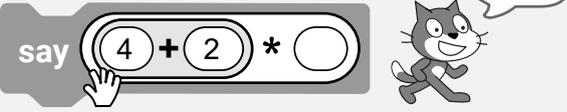
SCRATCH puts Magic into MATHS.

Combine two Operators inside a say () block.

E. continued

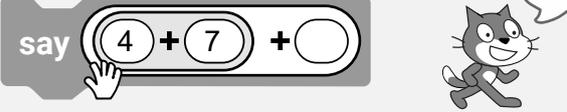
Practice

1. *Scratch adds* then *multiplies* 36



What missing number will make the sprite say 36 ?

2. *Scratch add* then *adds* again 14



What missing number will make the sprite say 14 ?

3. *Scratch adds* then *subtracts* 6



What missing number will make the sprite say 6 ?

4. *Scratch divides* then *multiplies* 21



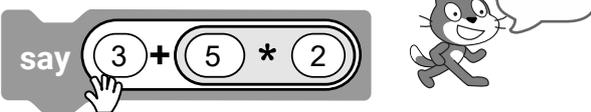
What missing number will make the sprite say 21 ?

Think backwards: Before 8 is subtracted what's the value of the inside block?

F. Code each combination below and click to make the sprite say the correct value.

When there is a combination of two Operators SCRATCH completes the **inside** before the **outside**.

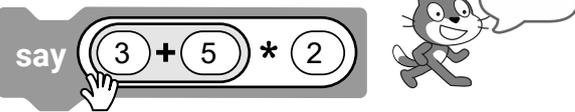
Put the **multiplication** block with 5×2 inside the **plus** block

a. 

SCRATCH **multiplies** before it **adds**
On paper, you would write this expression like this

$$3 + (5 \times 2) = \square$$

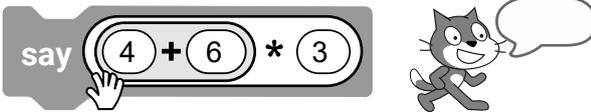
Put the **plus** block with $3 + 5$ inside the **multiplication** block

b. 

SCRATCH **adds** before it **multiplies**
On paper, you would write this expression like this

$$(3 + 5) \times 2 = \square$$

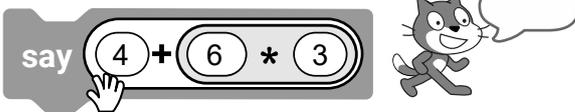
Put the **plus** block with $4 + 6$ inside the **multiplication** block

c. 

SCRATCH **adds** before it **multiplies**
On paper, you would write this expression like this

$$(4 + 6) \times 3 = \square$$

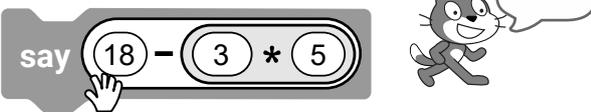
Put the **multiplication** block with 6×3 inside the **plus** block

d. 

SCRATCH **multiplies** before it **adds**
On paper, you would write this expression like this

$$4 + (6 \times 3) = \square$$

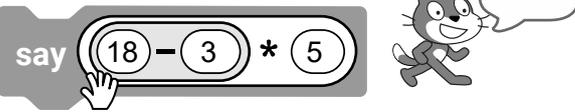
Put the **multiplication** block with 3×5 inside the **subtraction** block

e. 

SCRATCH **multiplies** before it **subtracts**
On paper, you would write this expression like this

$$18 - (3 \times 5) = \square$$

Put the **subtraction** block with $18 - 3$ inside the **multiplication** block

f. 

SCRATCH **subtracts** before it **multiplies**
On paper, you would write this expression like this

$$(18 - 3) \times 5 = \square$$

(All the answers: 3 13

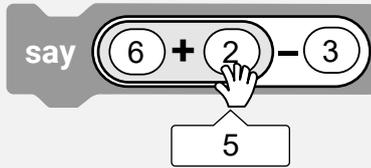
16 22 30 75)

G.

VISUAL THINKING with the SCRATCH

Number Story Problem 1

Six white eggs, 2 brown eggs and 3 of them broken. *Forget about eggs and think only about the numbers.*



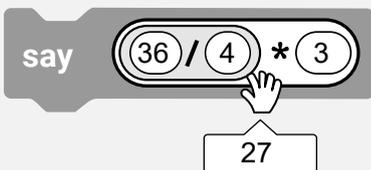
SCRATCH operators make it easier to think through a Number Story like this and **visualise the solution algorithm**. The obvious question is *How many good unbroken eggs are there?*

The **plus** operator tells how many eggs altogether. There are **8 eggs**. Scratch adds them first but we don't need to know. Scratch then *subtracts* the broken eggs and reports 5, which is the number we want. There are 5 good eggs.

On paper, the number story would look like this $(6 + 2) - 3$

Number Story Problem 2

There are 36 flags, and $\frac{3}{4}$ of them are green. How many green flags are there? *Think about the numbers.*



There are a number of ways to solve this problem, but the simplest **solution algorithm** can be **visualised** with Scratch operators. To get a fraction (such as a *quarter of 36*) it's necessary to divide. Then multiply by the number of *quarters* (3).

Scratch calculates a *quarter* of the flags first, using the **divide** operator. Then Scratch *multiplies by 3* to get *three quarters*. We don't need to know a *quarter* so Scratch just reports that there are 27 green flags.

On paper, the number story would look like this $(36 \div 4) \times 3$

Number Story Problem 3

A gardener has 3 rows of cabbages, with 20 in each row. She picks 16 cabbages. How many has she left?

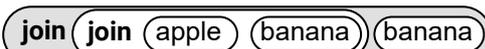
Forget about cabbages and *think only about the numbers*. and code the solution algorithm in Scratch with just two operators.

Sketch the solution algorithm here and code it to make Scratch work it out.

H. Combine Text and Numbers using SCRATCH join operators

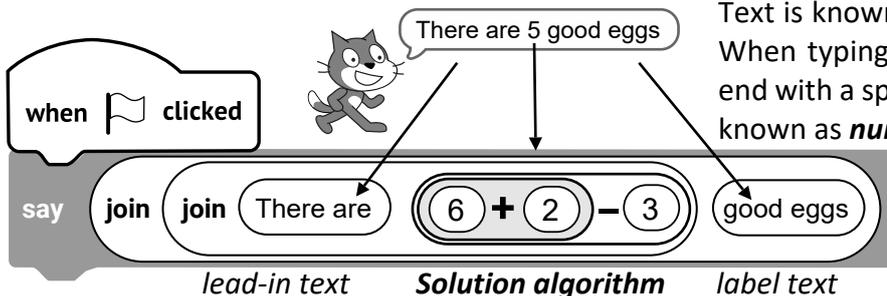
We want to make the sprite say more than the bare answer, e.g. we want the sprite to say "There are 5 good eggs" in Number Story 1. This involves combining *lead-in text*, *the solution* and *label text*. You need two **join** blocks for this. Each **join** has two input windows. When you drag one **join** block into the window of another you have 3 input windows. That's sufficient for this exercise. Sometimes you may need three or more **join** operators.

To do: Drag two **join** blocks from the Operators palette. Insert one of them into the first input window of the the other. The holding text *apple* and *banana* will reveal three windows, *apple banana banana*. Type text into windows 1 and 3.



Text is known as **string data** when in a join block. When typing text in the input window, start and end with a space. Insert the full code combination, known as **numeric data** in the middle window.

Scratch reports this as a number. You can insert the **join** combination in text-to-speech to add sound to the sprite's speech balloon.



In the Operators palette drag out two **join operator**.