

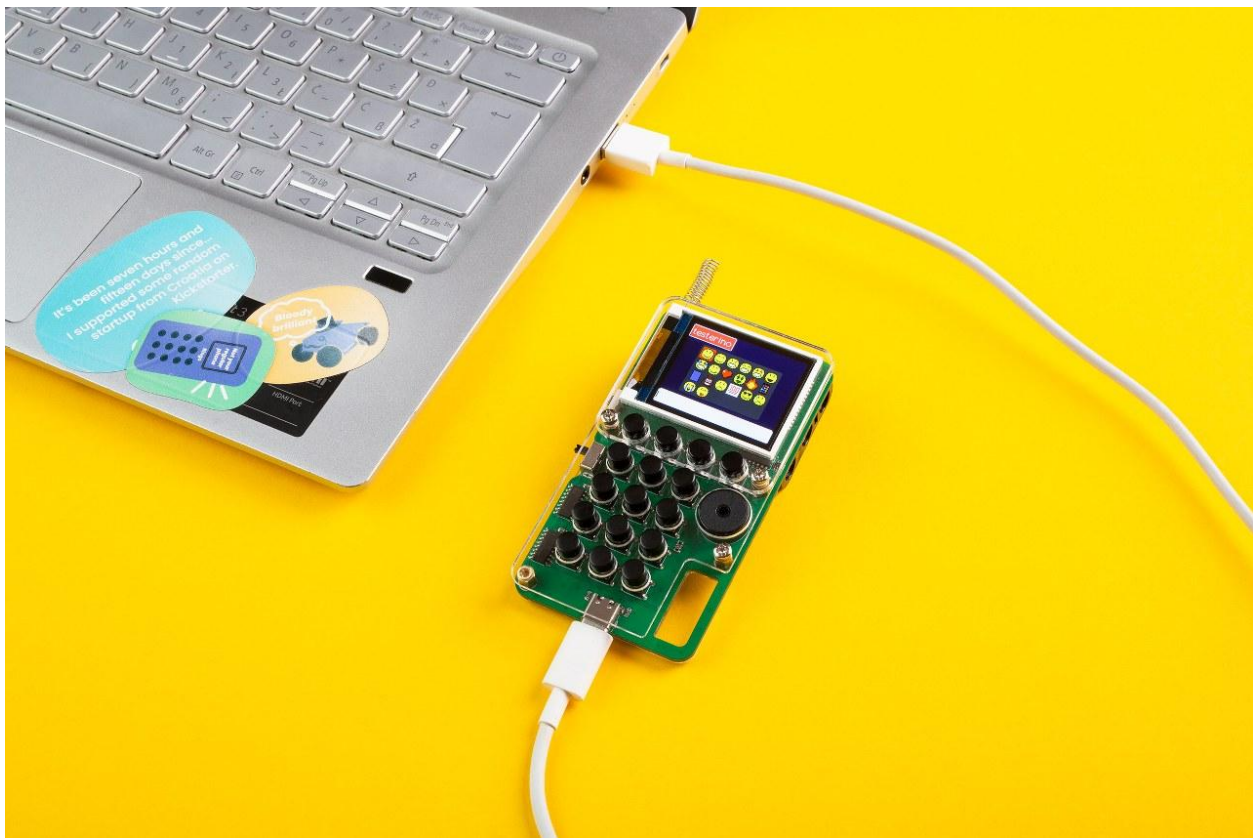
Coding for beginners - how to code your Chatter

Chatter

Let's draw on the display!

Let's get down to business!

Before doing anything, you need to connect your Chatter to your computer's USB port and turn it on.



If everything is okay, CircuitBlocks should say, "Chatter connected".

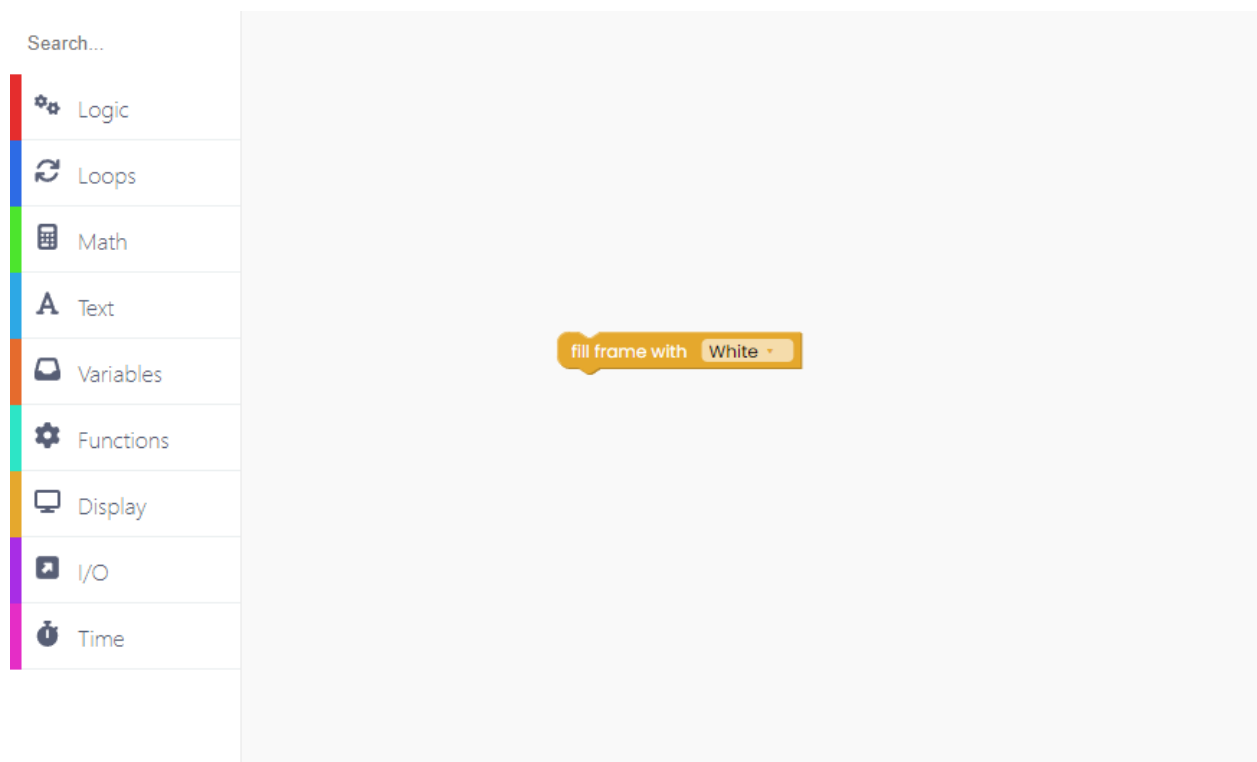
Let's draw something!

We will kick things off as simply as possible.

The first thing we're going to learn is how to clear Chatter's display and draw on it!

You'll only need to use a Display block section to do that.

Firstly, drag and drop the "fill frame with black" block.



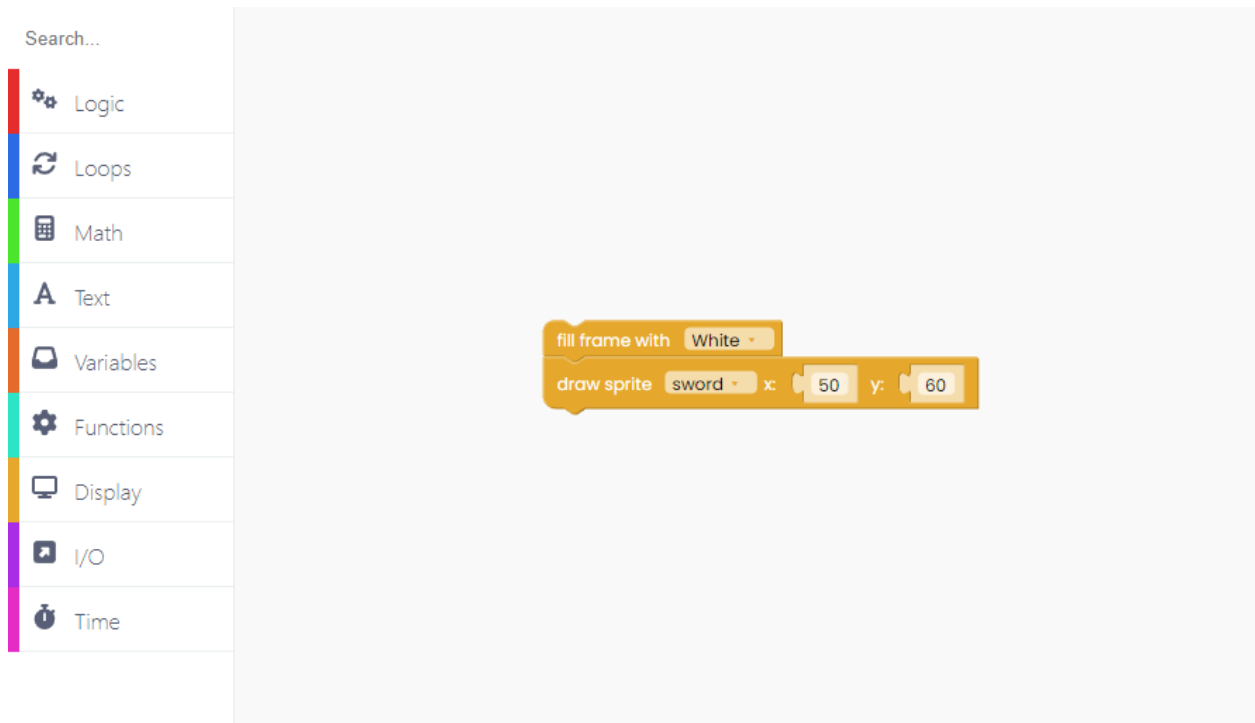
We changed the background color to white, but you can choose any color you want.

You can choose different colors by clicking on the word "black".

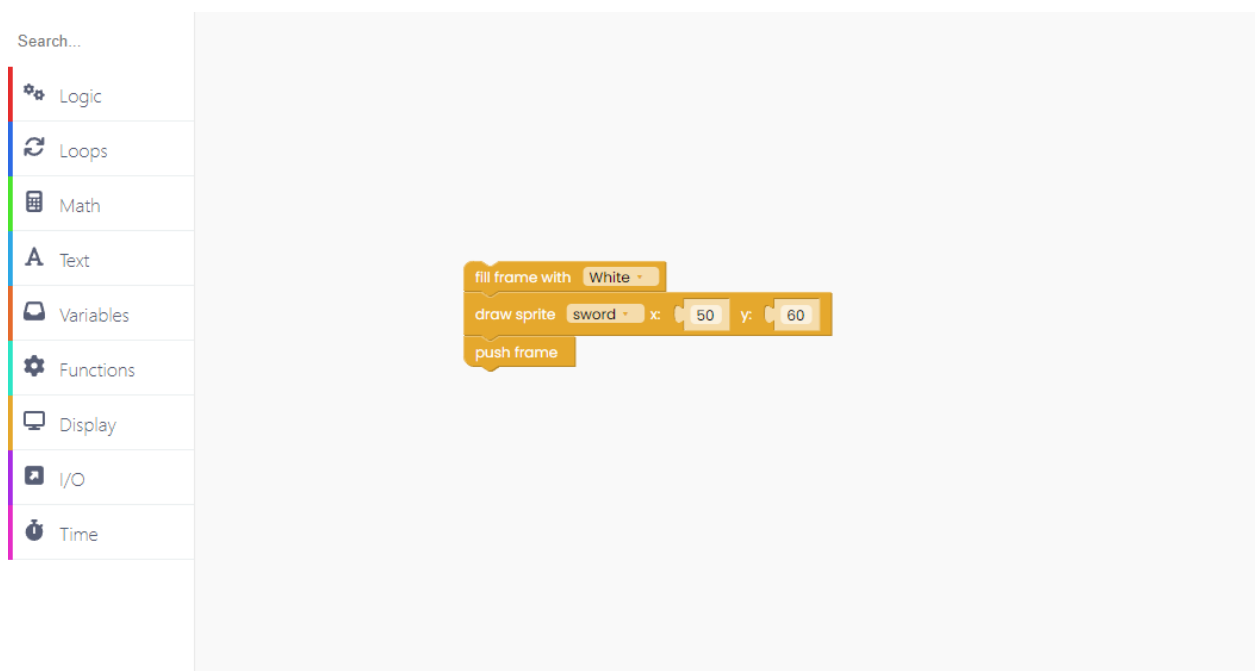
Now, look for the block labeled "draw sprite ... x ... y".

We chose to draw the sword, but you can use any of the available sprites.

Both coordinates are set to 50.



To guarantee that your code executes successfully, you must always include the "push frame" block at the end of the "Display" codes.



Now, click the Run button and the sword should appear on the Chatter's screen.

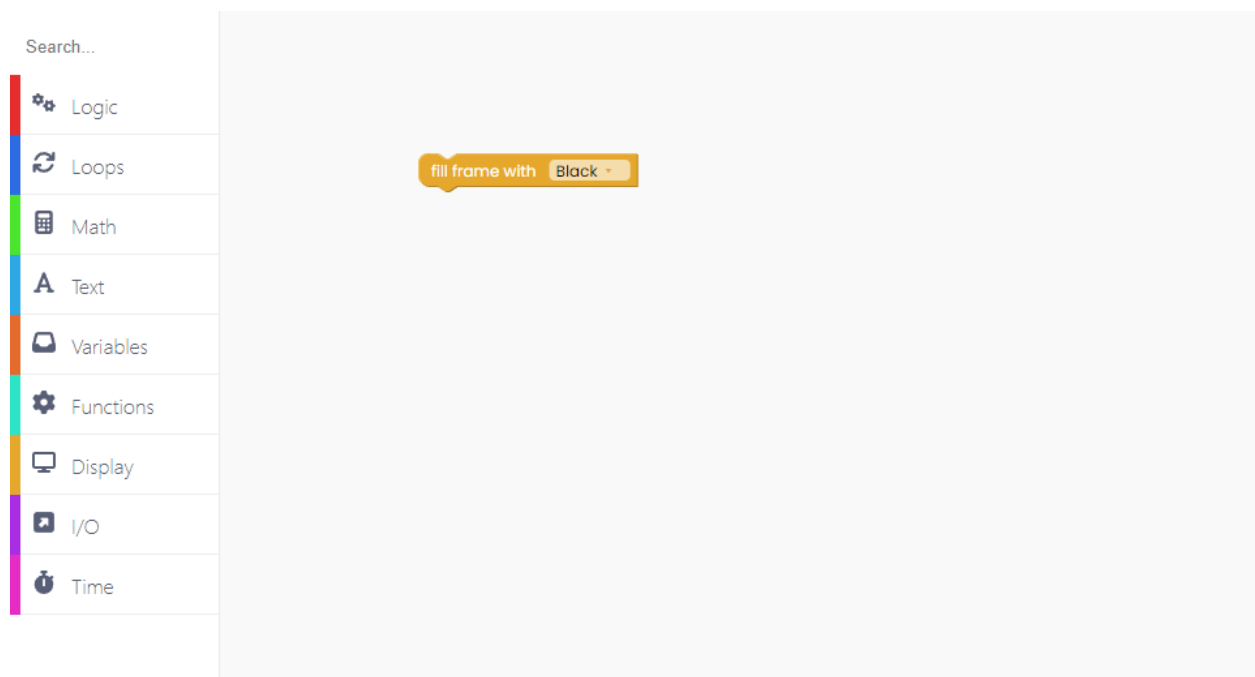
Click, click...

Now that you know a thing or two about CircuitBlocks, it's time for a bit more advanced sketch.

Firstly, we want to code what will happen first once we run the code.

We'd like to write the instructions on the screen for you to follow.

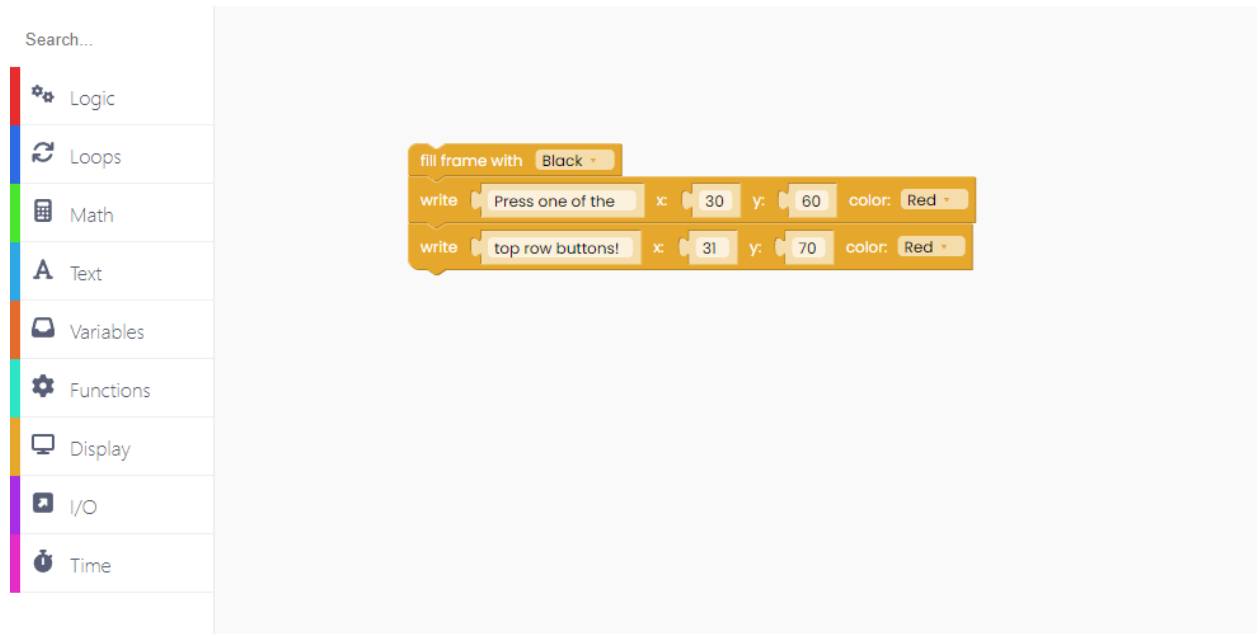
The "fill frame with black" code will be used first.



After we took care of the background, it's time to write the instructions.

Please note that you can choose any background color you want.

Now, in the "Display" block section, look for the "write" block.



The directions, as you can see, are written in red and say "Press one of the top row buttons!".

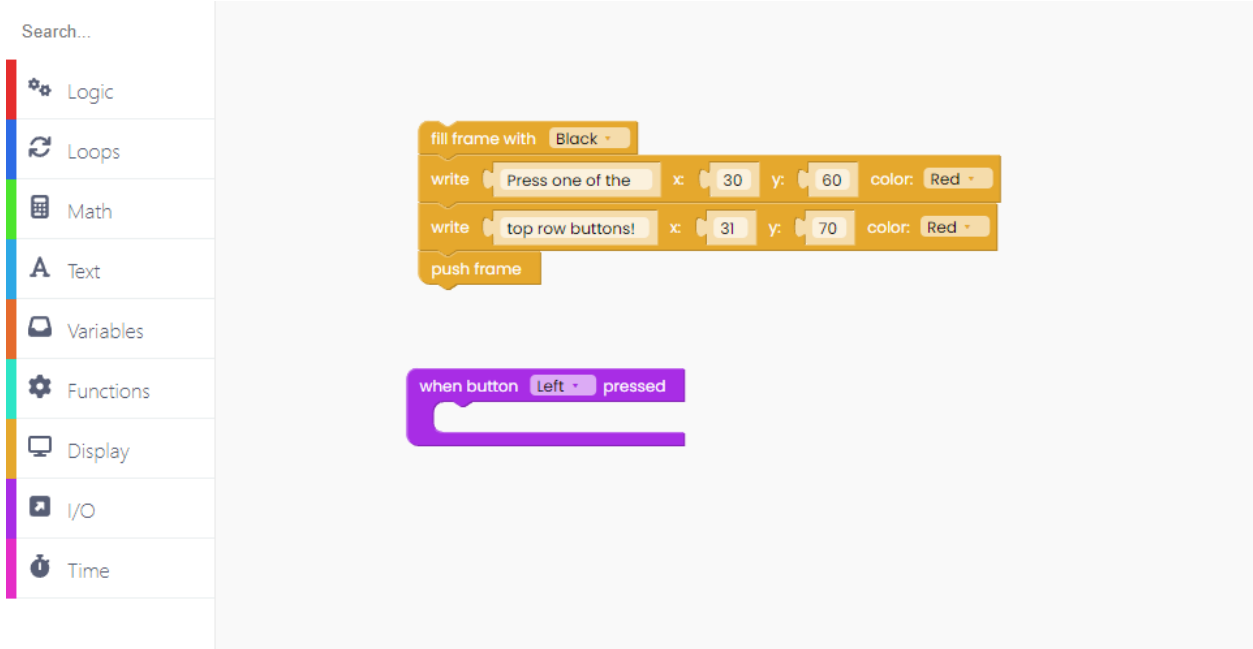
To fit the display, we divided the sentence into two blocks.

To guarantee that the code executes properly, add the "push frame" block at the end of any "Display" code.



Let's now code what happens when we press and release the buttons.

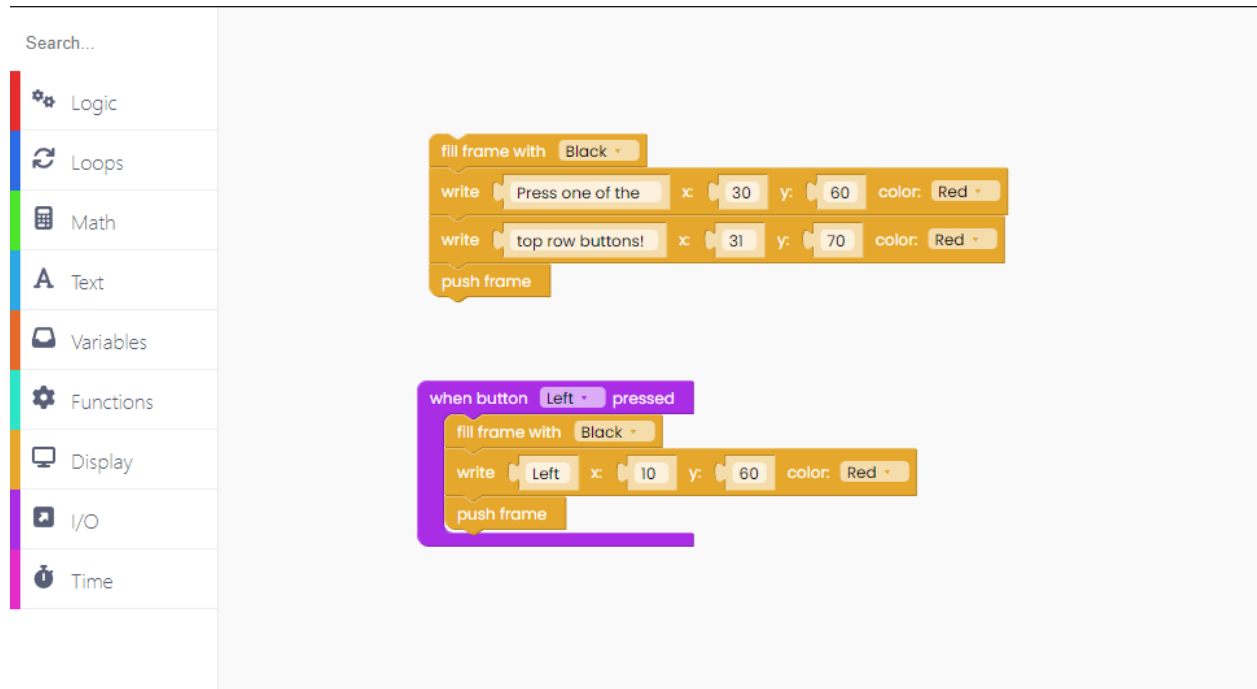
To do that, find the "When button pressed" button from the I/O block section.



So, within the purple block, we want to write the code that we want to run only once when we push that button.

We want the screen to go black and to show which button has been pressed.

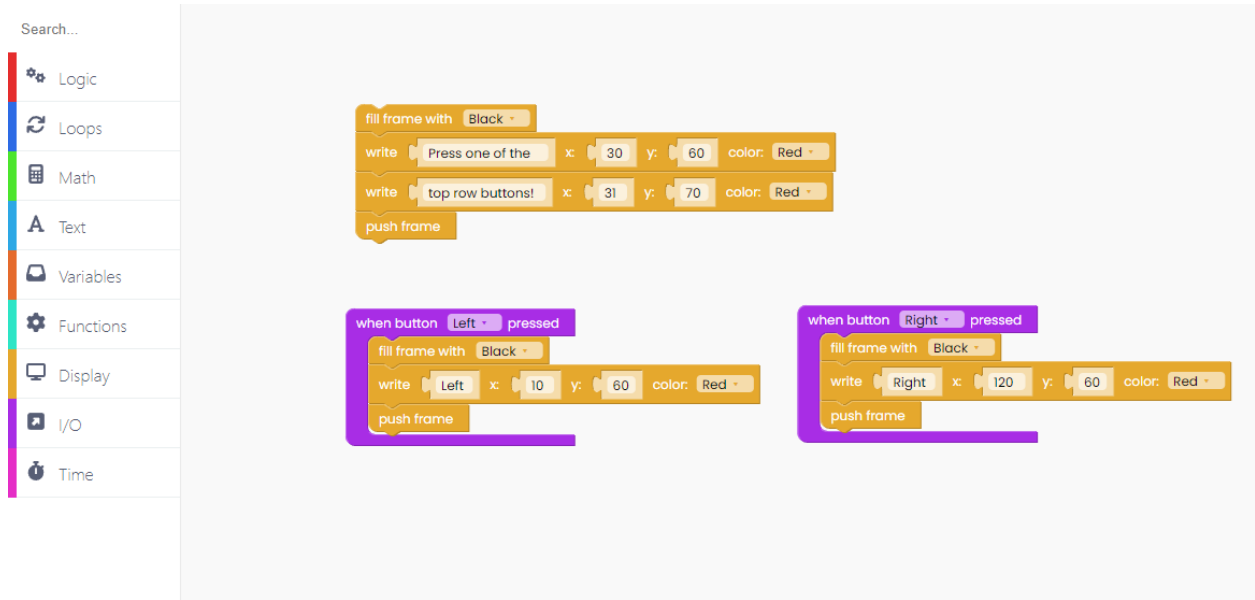
Don't forget to include the "push frame" block at the end, or your code will fail.



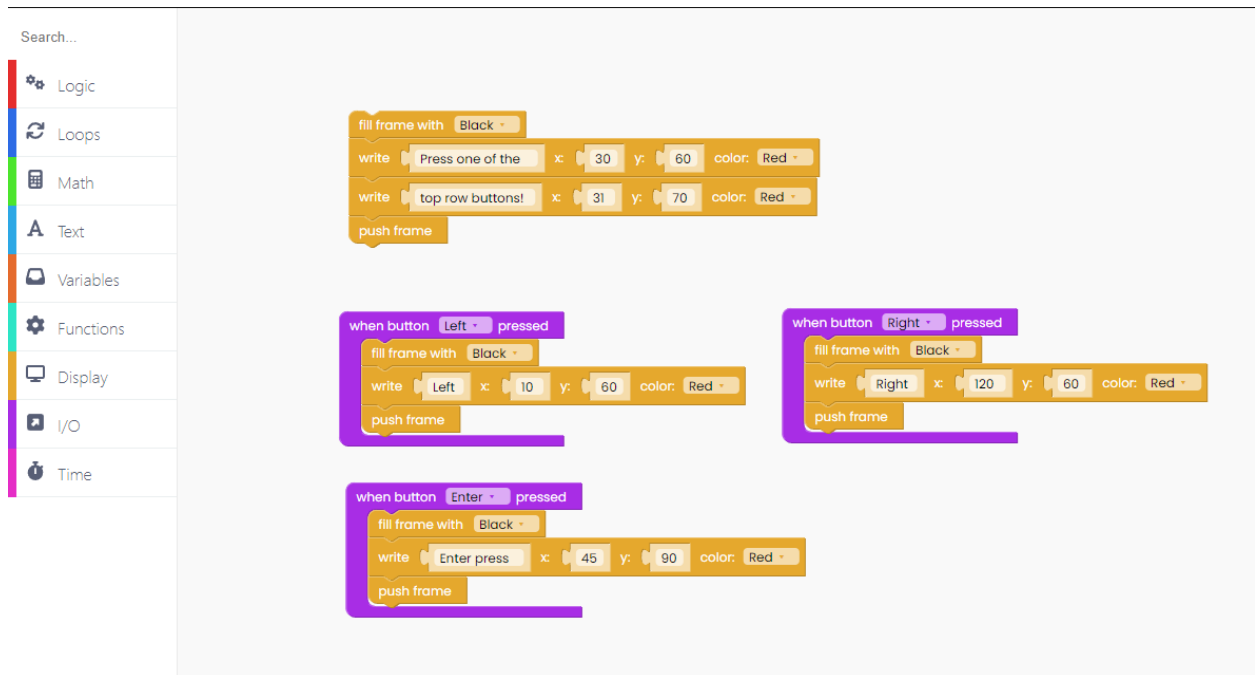
The image shows a Scratch code editor interface. On the left is a sidebar with a search bar and categories: Logic, Loops, Math, Text, Variables, Functions, Display, I/O, and Time. The main workspace contains two code blocks. The first block is a sequence of three orange blocks: 'fill frame with Black', 'write Press one of the' (with x: 30, y: 60, color: Red), and 'write top row buttons!' (with x: 31, y: 70, color: Red), followed by a 'push frame' block. The second block is a purple 'when button Left pressed' block containing three orange blocks: 'fill frame with Black', 'write Left' (with x: 10, y: 60, color: Red), and 'push frame'.

Let's make a copy of this I/O block.

Change left to right, and feel free to play with the coordinates.

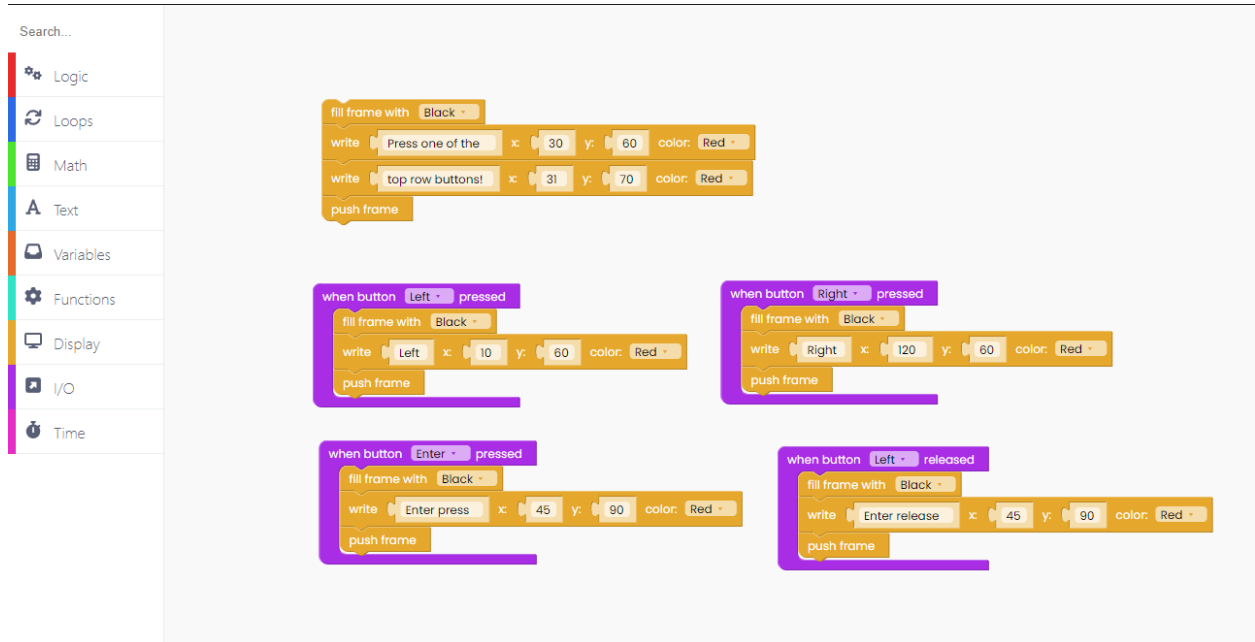


Duplicate it once more for when you press enter.

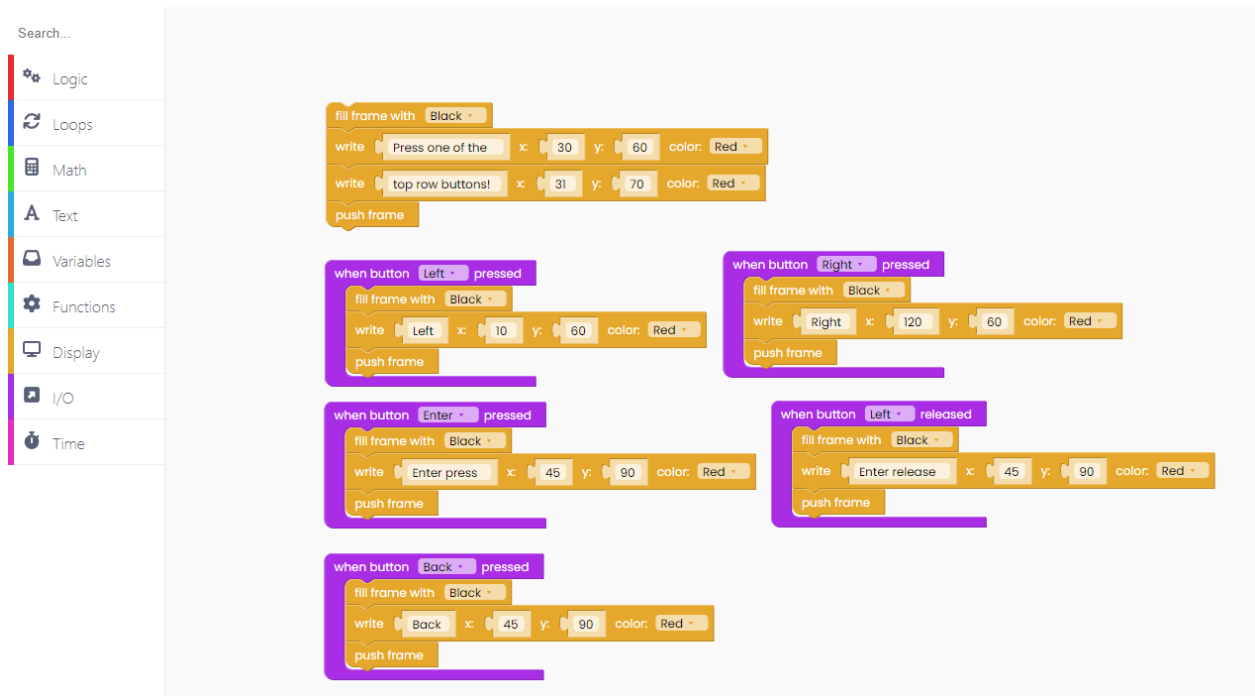


We want the text to change to "Enter release" when we release the enter key.

That block can also be found in the I/O block section.



Finally, we'll add a press block for the back button.



When coding with the buttons, it's critical to include the "loops forever" and "scan buttons" blocks at the end to guarantee the buttons are always scanned and the code runs properly.

Click the Run button and start playing with the buttons.

Buzzzzzer

Let's learn what to do with the Piezo buzzer you soldered onto your Chatter.

As the word itself says, the buzzer is used for making buzzing sounds.

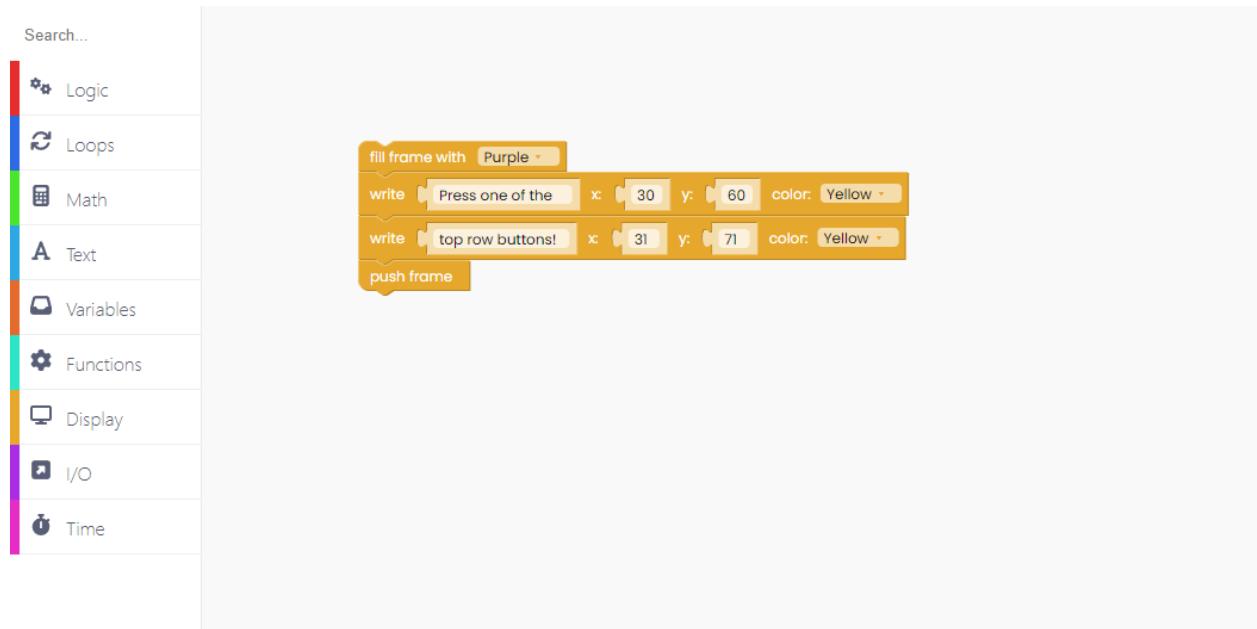
We'll make a very similar sketch to the last one, but this time, pressing the buttons will trigger a particular sound to come out of the buzzer.

Let's start!

Let's start by coding what will happen when we run the code.

We want the instructions to appear on the screen.

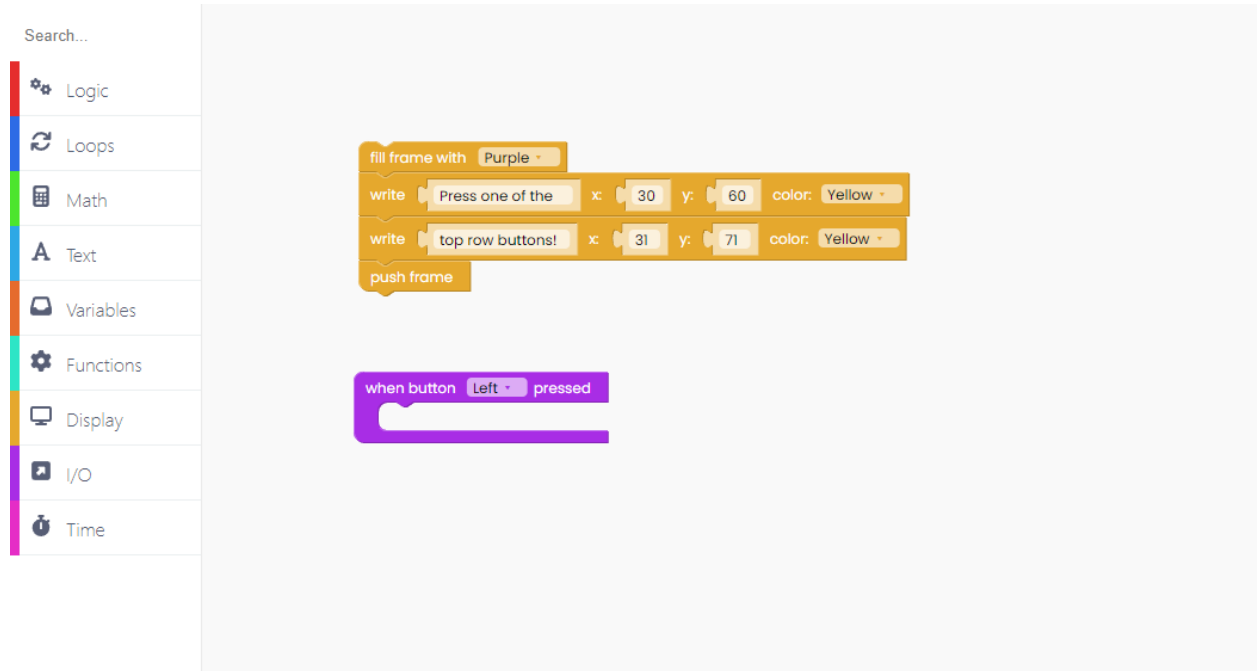
We'll use the same code as in the last example.



So we want the display to turn purple and show the message "Press one of the top row buttons!".

Now let's see what happens with the buzzer when a certain button gets pressed.

Like in the previous sketch, we'll use I/O blocks to identify what happens when specific buttons are pressed.



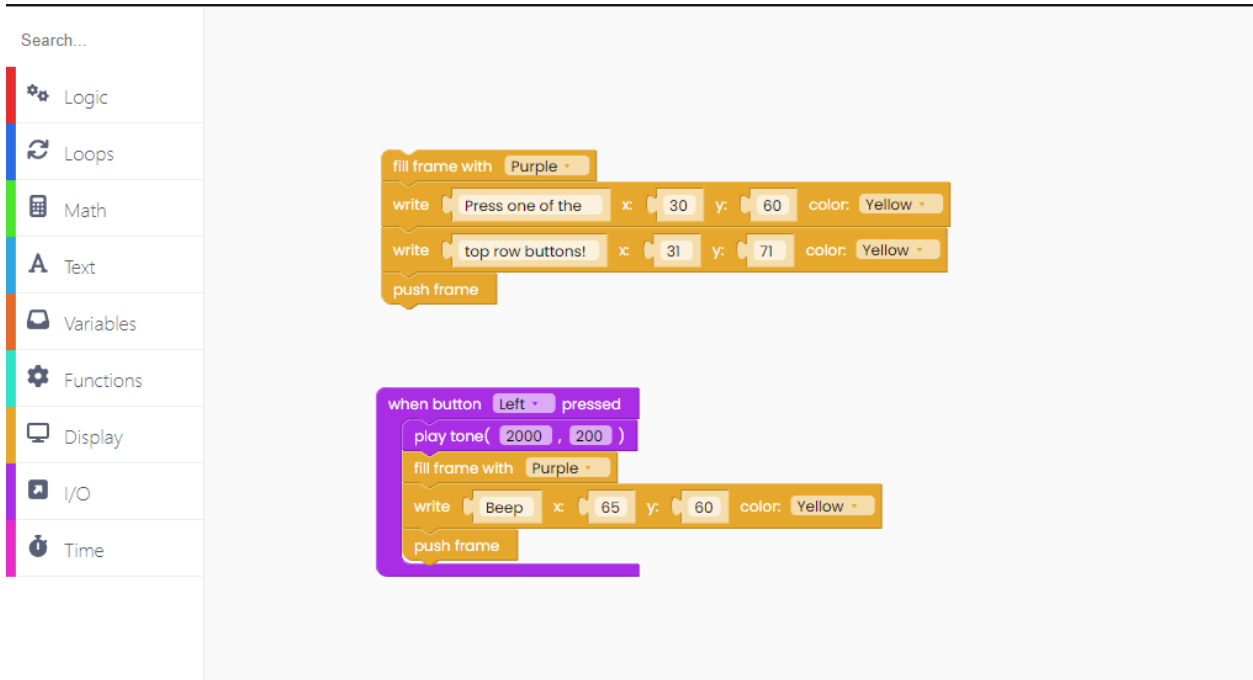
Let's introduce you to a new block called "Play tone (2000, 200)".

The first number represents the frequency, while the second number determines how long the tone will play.

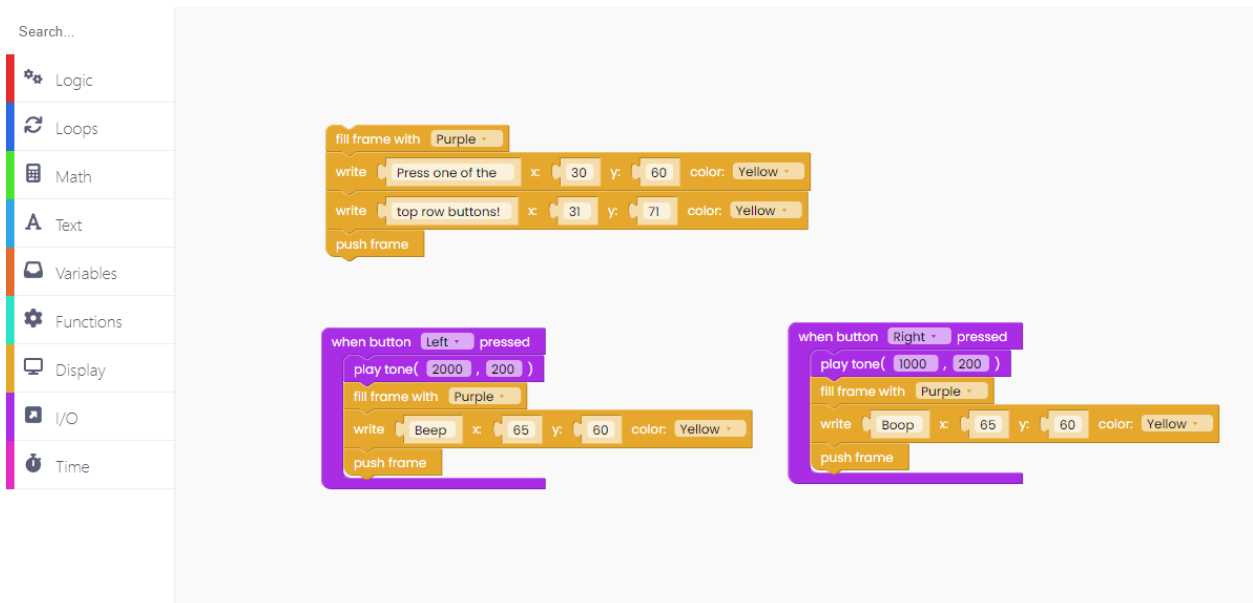
Aside from that, we want the screen to change color and to write what sound the buzzer makes.

That will be "Beep" for the first one.

This is what your code should look like by now:



Let's duplicate the I/O block and see what happens when the right button is pressed.



Duplicate it once more for the enter.

Search...

- Logic
- Loops
- Math
- Text
- Variables
- Functions
- Display
- I/O
- Time

```

fill frame with Purple
write Press one of the x: 30 y: 60 color: Yellow
write top row buttons! x: 31 y: 71 color: Yellow
push frame

when button Left pressed
  play tone( 2000 , 200 )
  fill frame with Purple
  write Beep x: 65 y: 60 color: Yellow
  push frame

when button Right pressed
  play tone( 1000 , 200 )
  fill frame with Purple
  write Boop x: 65 y: 60 color: Yellow
  push frame

when button Enter pressed
  play tone( 100 , 150 )
  fill frame with Purple
  write Buzz x: 65 y: 60 color: Yellow
  push frame
  
```

And, for the back button.

Search...

- Logic
- Loops
- Math
- Text
- Variables
- Functions
- Display
- I/O
- Time

```

fill frame with Purple
write Press one of the x: 30 y: 60 color: Yellow
write top row buttons! x: 31 y: 71 color: Yellow
push frame

when button Left pressed
  play tone( 2000 , 200 )
  fill frame with Purple
  write Beep x: 65 y: 60 color: Yellow
  push frame

when button Right pressed
  play tone( 1000 , 200 )
  fill frame with Purple
  write Boop x: 65 y: 60 color: Yellow
  push frame

when button Enter pressed
  play tone( 100 , 150 )
  fill frame with Purple
  write Buzz x: 65 y: 60 color: Yellow
  push frame

when button Back pressed
  play tone( 100 , 150 )
  fill frame with Purple
  write Buzz x: 65 y: 60 color: Yellow
  push frame
  
```

As noted in a previous example, you must include the "loop forever" and "scan buttons" blocks at the end of any code involving buttons.



Your code is ready!

Click on the Run button and check it out.

Write on the screen

And the final sketch will be something simple and enjoyable.

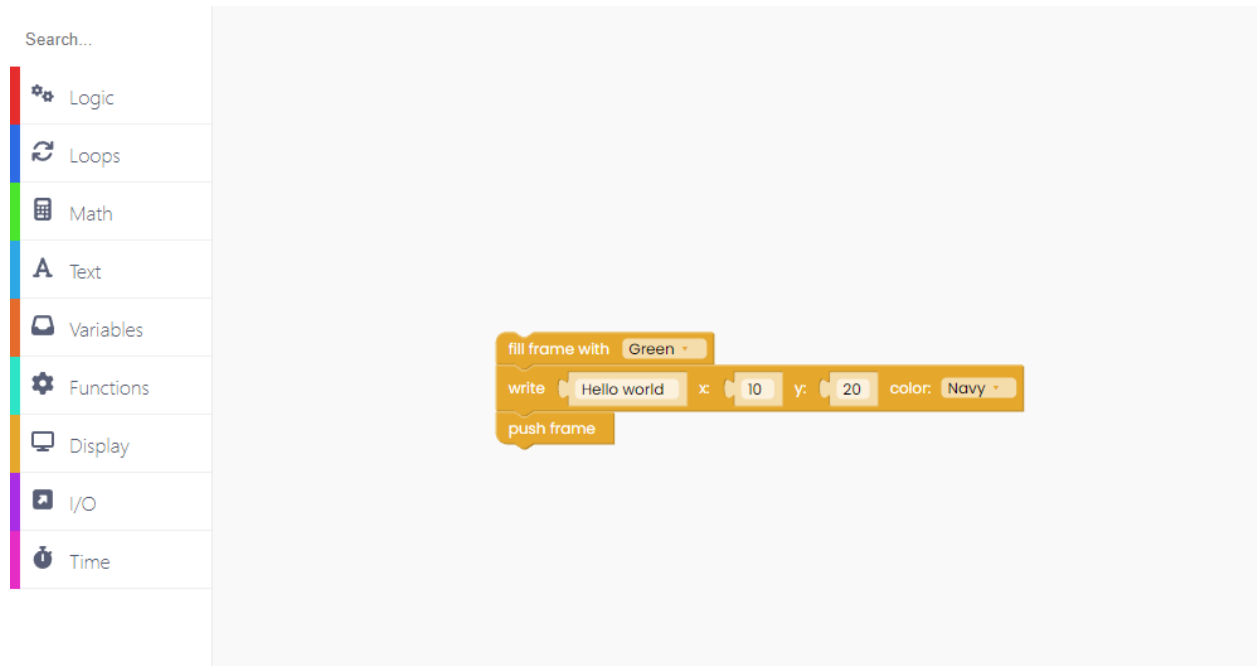
That is, after the code is executed, something should be written on the screen.

Try to code the text by yourself before checking our code below.

Were you successful?

All of the blocks can be found in the "Display" block section.

This is what our code looks like:

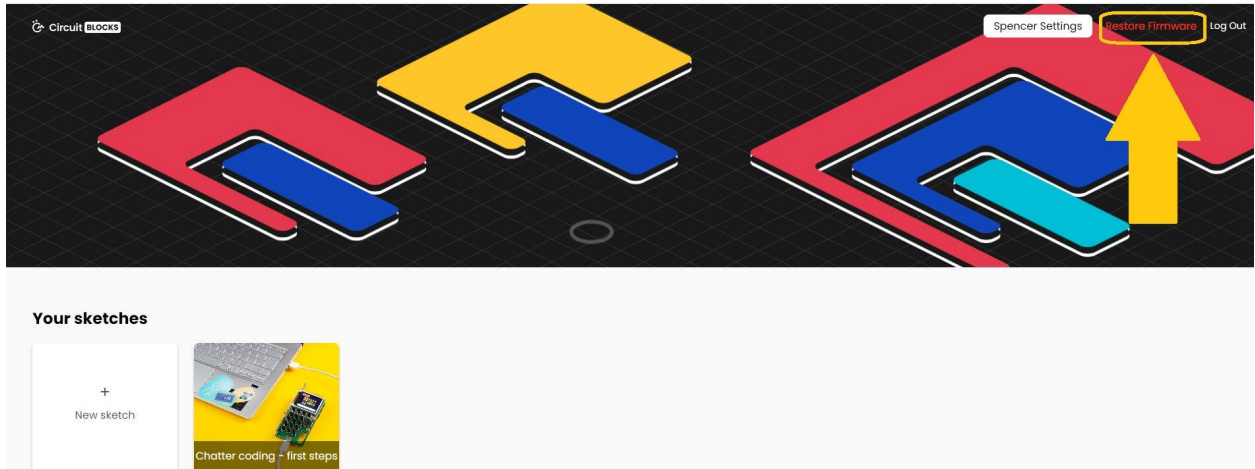


Click on the Run code and check it out.

Restore Chatter's firmware

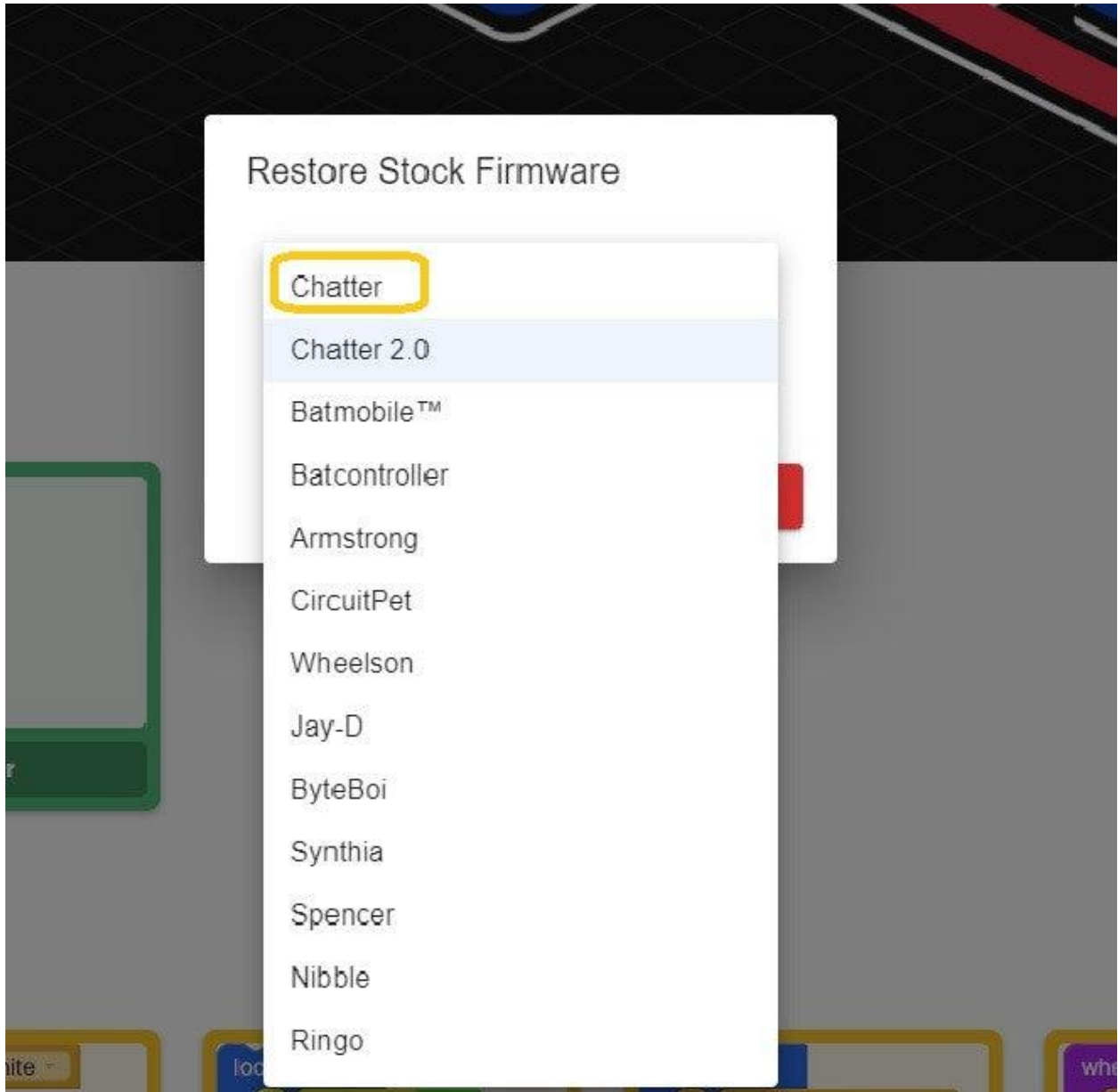
If you want, for any reason, to restore Chatter's firmware, follow these steps.

This is quite simple, just connect your Chatter to the USB port of your computer and press the "Restore firmware" button on the top right.



You will be prompted with a window to choose the device you are restoring the firmware for.

Choose Chatter, of course.



Wait for a few seconds, and your Chatter will be back and running like usual.

What's next?

You've reached the end of our first Chatter coding tutorial, congratulations!



I hope you're as excited as we are about Chatter's future since there are so many cool things we want to do with it in future firmware and CircuitBlocks updates.

In the meantime, continue exploring on your own and show us what you've done with your Chatter by sharing it on the [CircuitMess community forum](#) or via our [Discord channel](#).

If you need any help with your device, as always, reach out to us via contact@circuitmess.com, and we'll help as soon as we can.

Thank you, and keep making!