

Coding for beginners - how to code your CircuitPet

Let's start! Step by step

Blink, blink!

Let's get down to business!

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



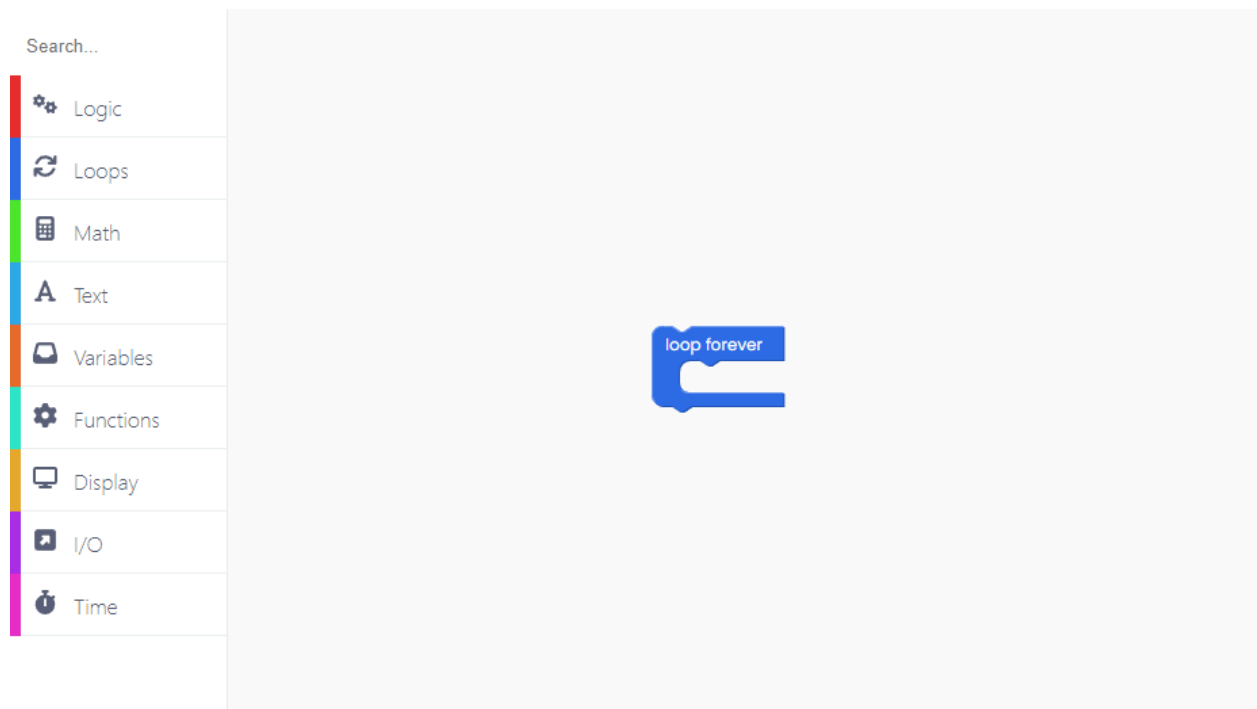
Let's play with the LED!

The first thing we'll learn is how to code CircuitPet's LED to blink in different colors!

The first block that we need to find says "loop forever". This particular block ensures your code is executed all the time.

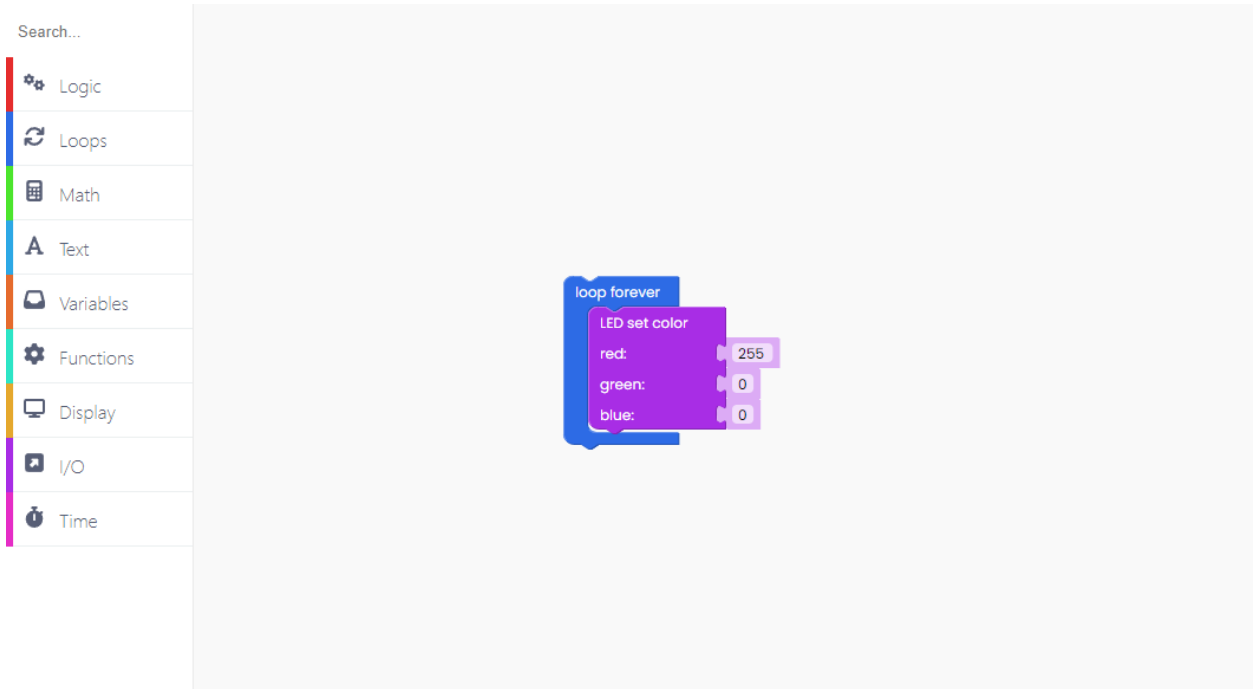
You can find the block in the "Loops" block section.

Drag and drop it onto the drawing area.

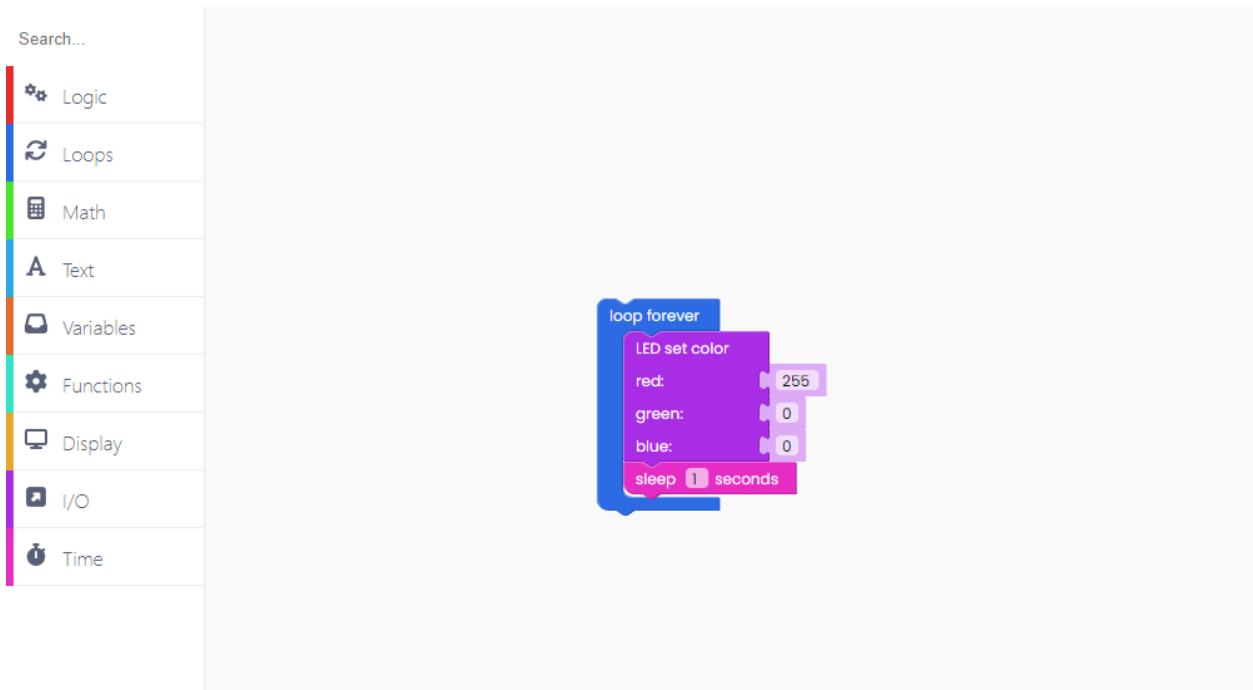


Look for the "LED set color" block in the I/O block section.

Because the LED will light up first in red, set the value of red to 255 (the maximum) and the values of the other colors to 0.

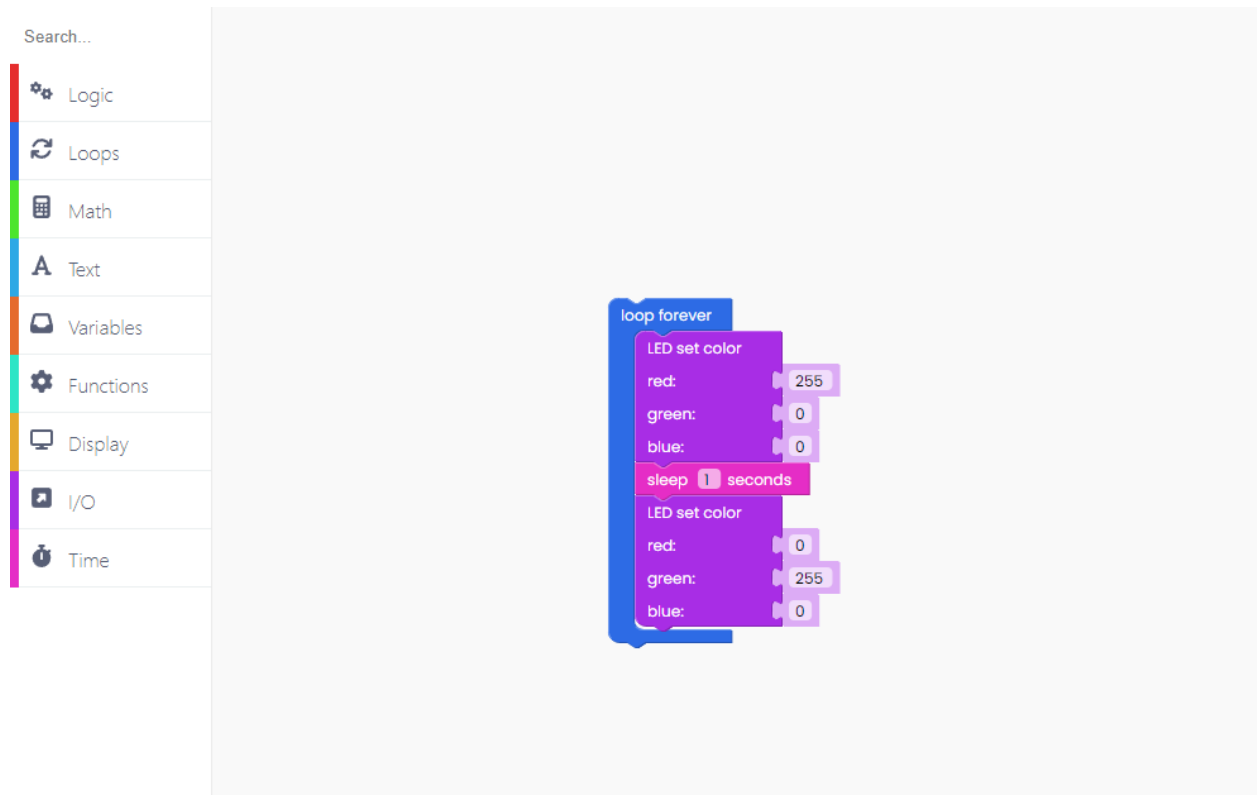


Between red and the other color that will light up, we'll make a pause of 1 second.



Duplicate the I/O block and place it underneath the time block.

Because the LED will light up green this time, set the value of red to 0 and the value of green to 255.



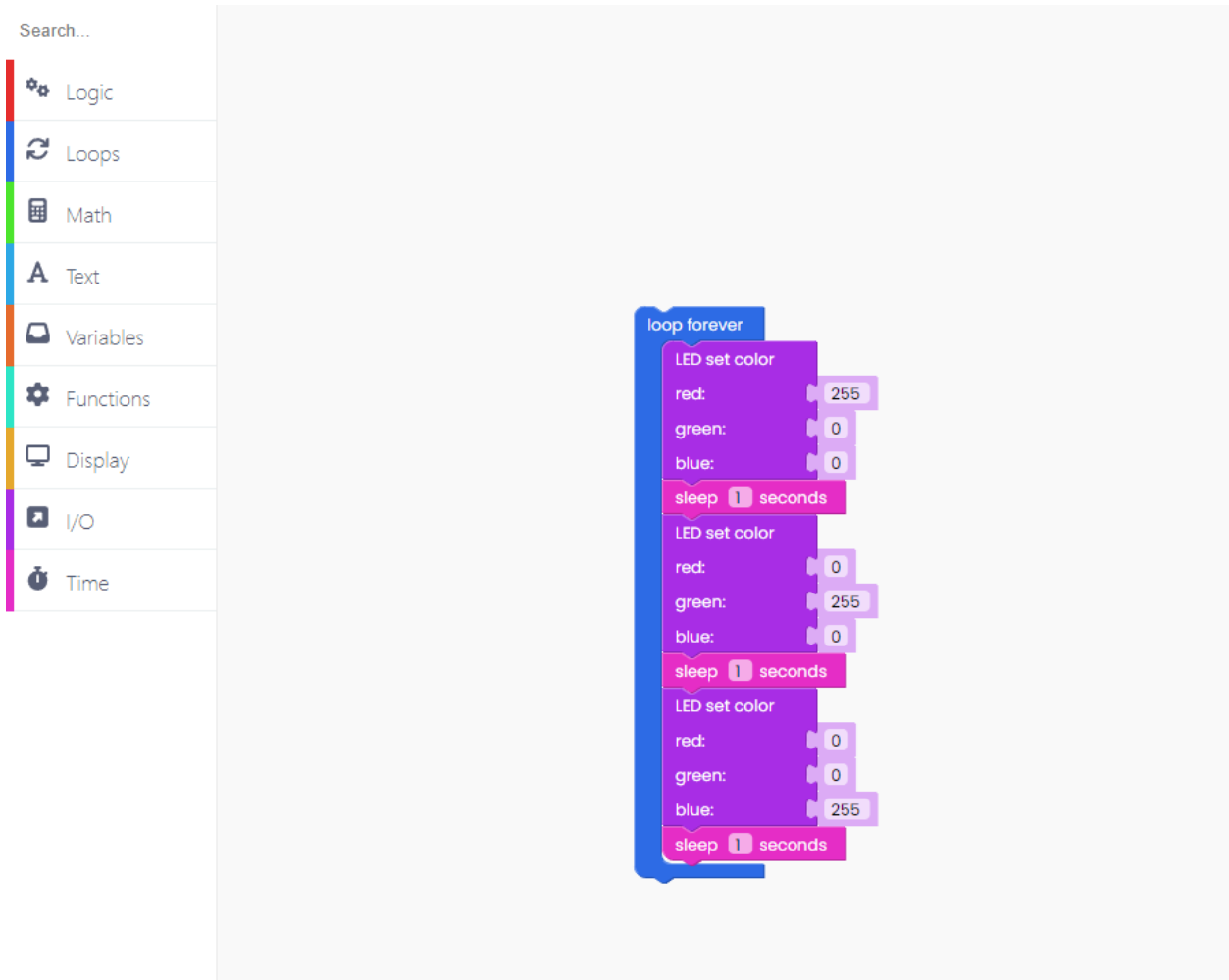
Don't forget the time block.

Search...

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

```
loop forever
  LED set color
    red: 255
    green: 0
    blue: 0
  sleep 1 seconds
  LED set color
    red: 0
    green: 255
    blue: 0
  sleep 1 seconds
```

Repeat the process once again, this time for blue.



Now, hit the big red "Run" button and watch the LED change its color.

Display time!

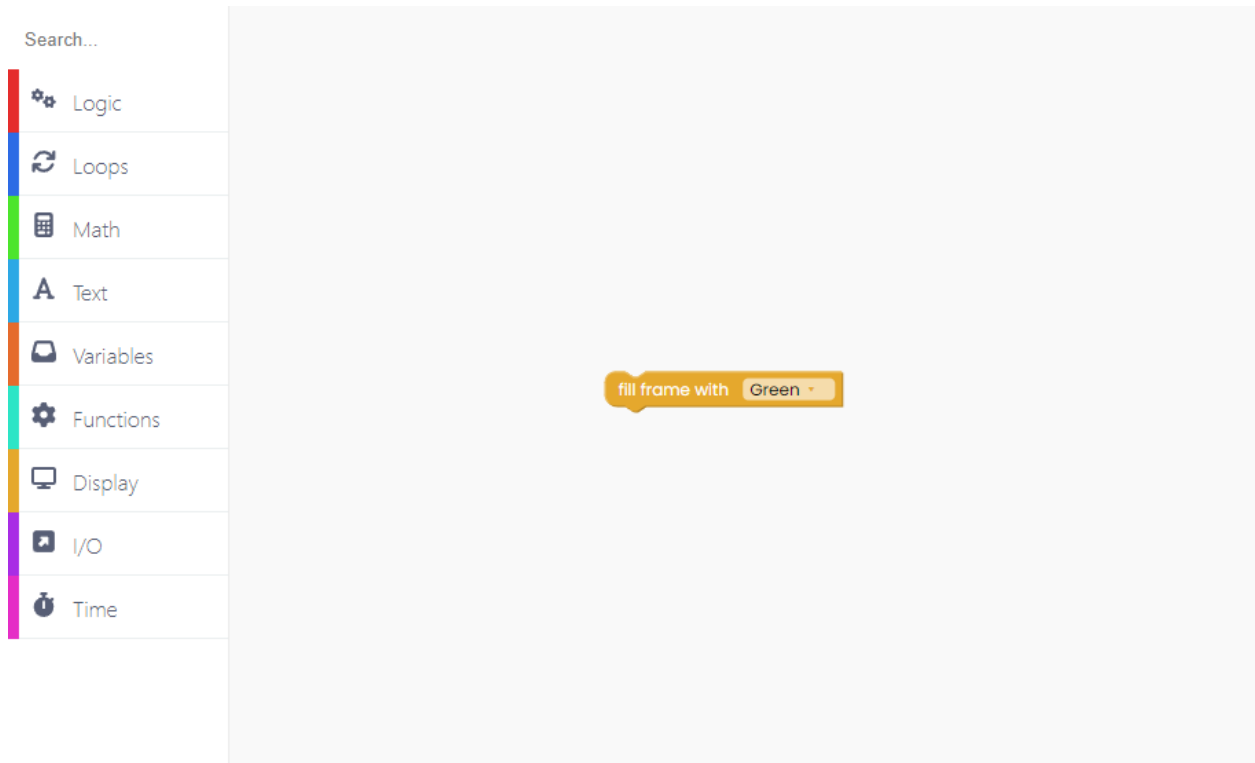
In this chapter, we'll show you how to make your CircuitPet display a text you created on the screen.

The first thing we have to do while coding anything on the screen is to fill the screen in one color.

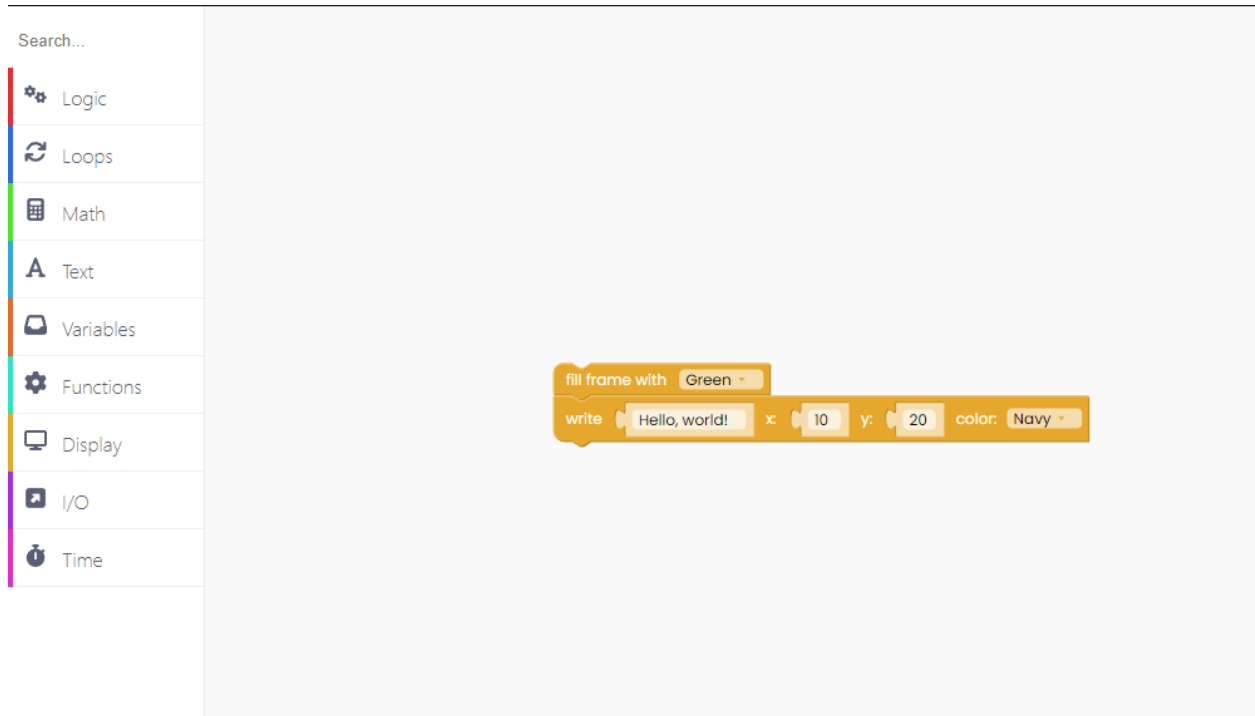
To do that, find the "fill frame with black" in the "Display" block section.

Drop it onto the drawing area.

We'll change the background color into green.

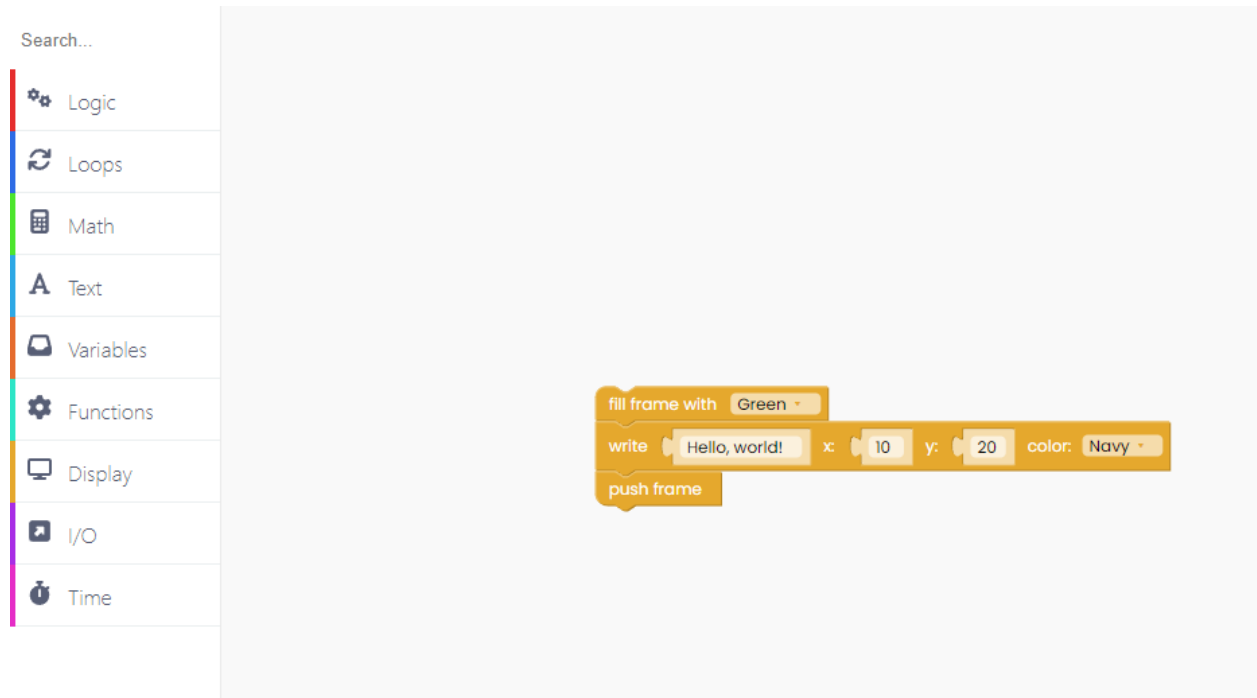


Below that, add this block:



So, the thing we want to write is "Hello, world!" on the coordinates x: 10 and y: 20 in navy.

After this, you have to add the "push frame" block to ensure the text will appear on the screen.



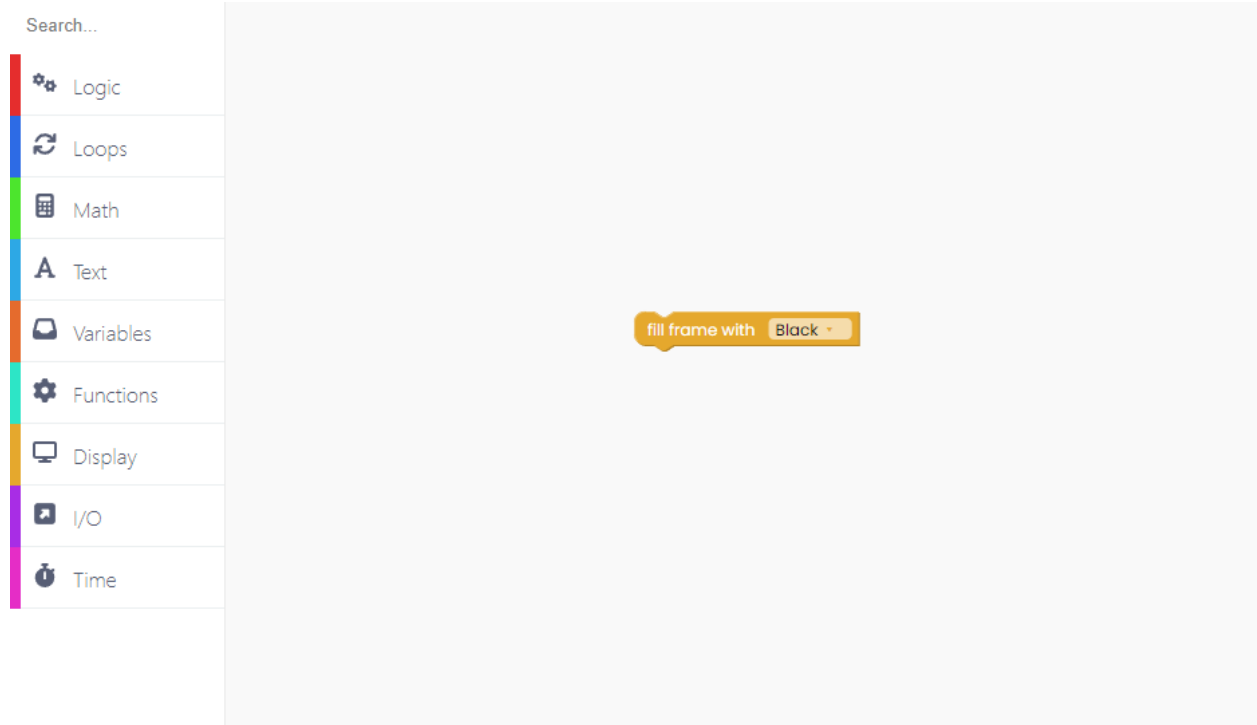
Click on the Run button and watch the text appear on the display.

Play with the inputs!

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

Go to the Display section, and click on the "fill frame with black" block.

This block will clear the display with some color so we can clearly see the sentence (or word) you'll print.

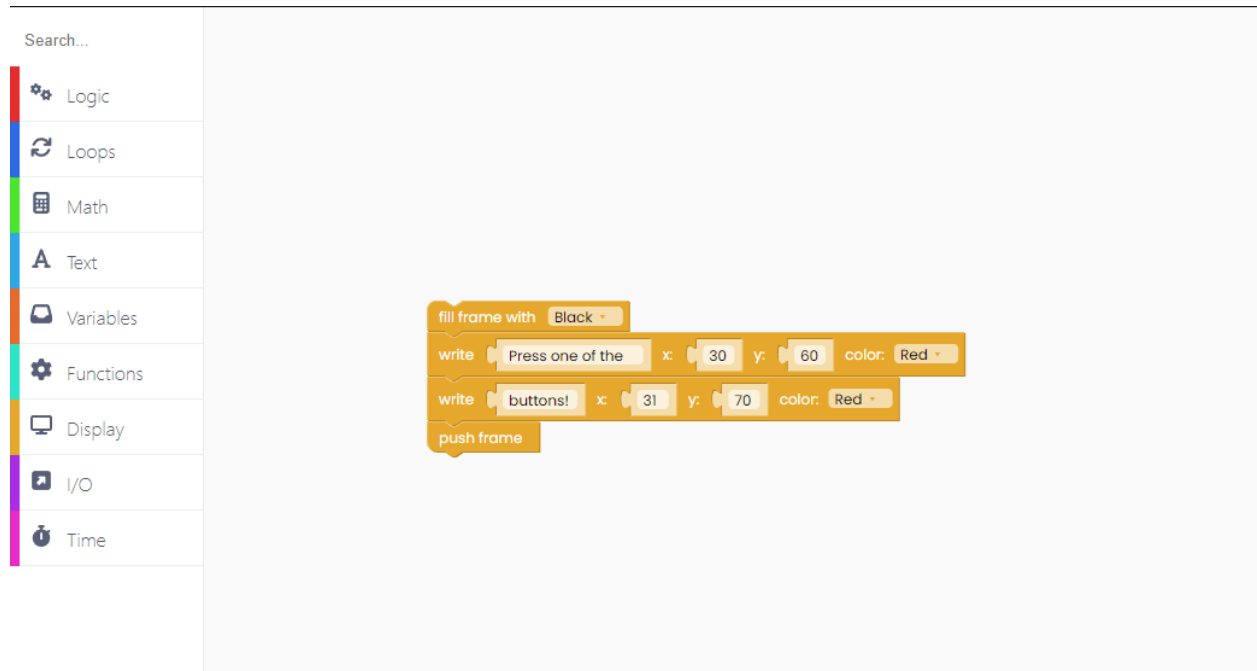


Now we get to choose what we want to put on display.

You can do it just like in the previous example.

Find the "write" block and write what will appear on the screen once the code is executed.

Let's write the instructions for this code because the goal is to push specific buttons and display words on the screen.

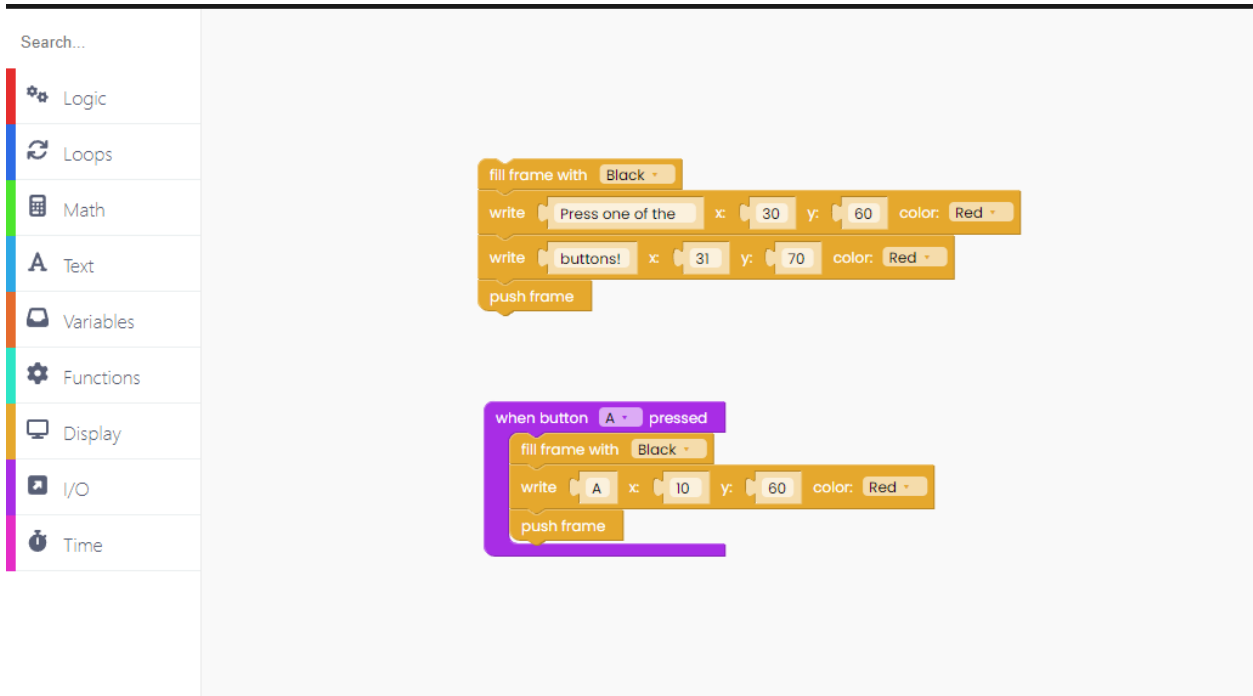


So the text "Press one of the buttons!" will appear in red letters.

Now is the time to code what will happen when the particular button gets pressed.

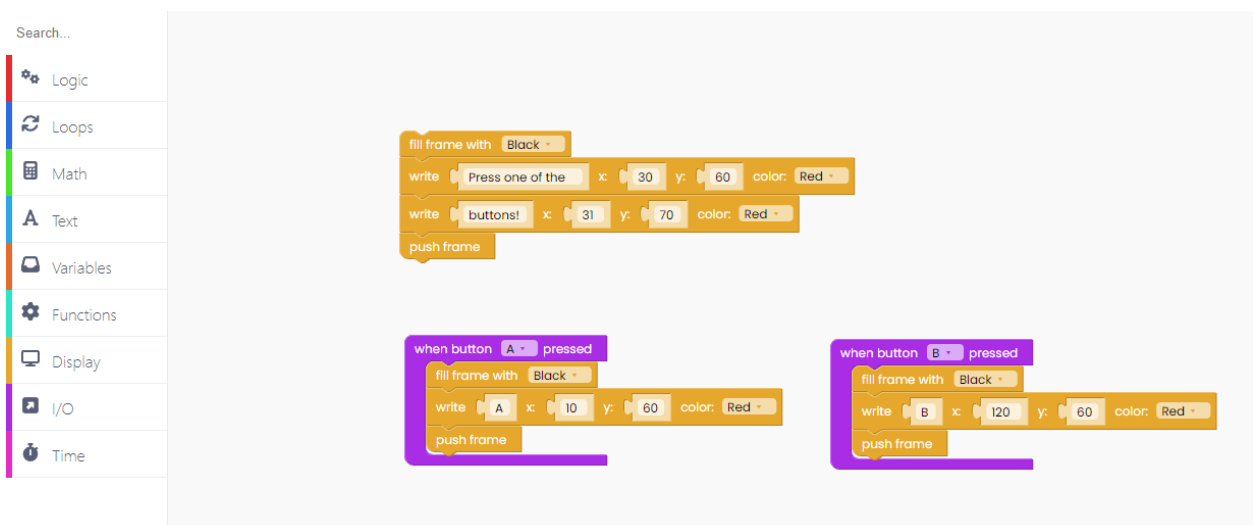
To program that, find the I/O blocks labeled "when button gets pressed".

Let's start with the button A.



So, when we press the button A, the screen goes black, and the letter 'A' will appear on it to indicate that the button A was pressed.

Do the same thing with button B.



Of course, you can play with different coordinates and text color.

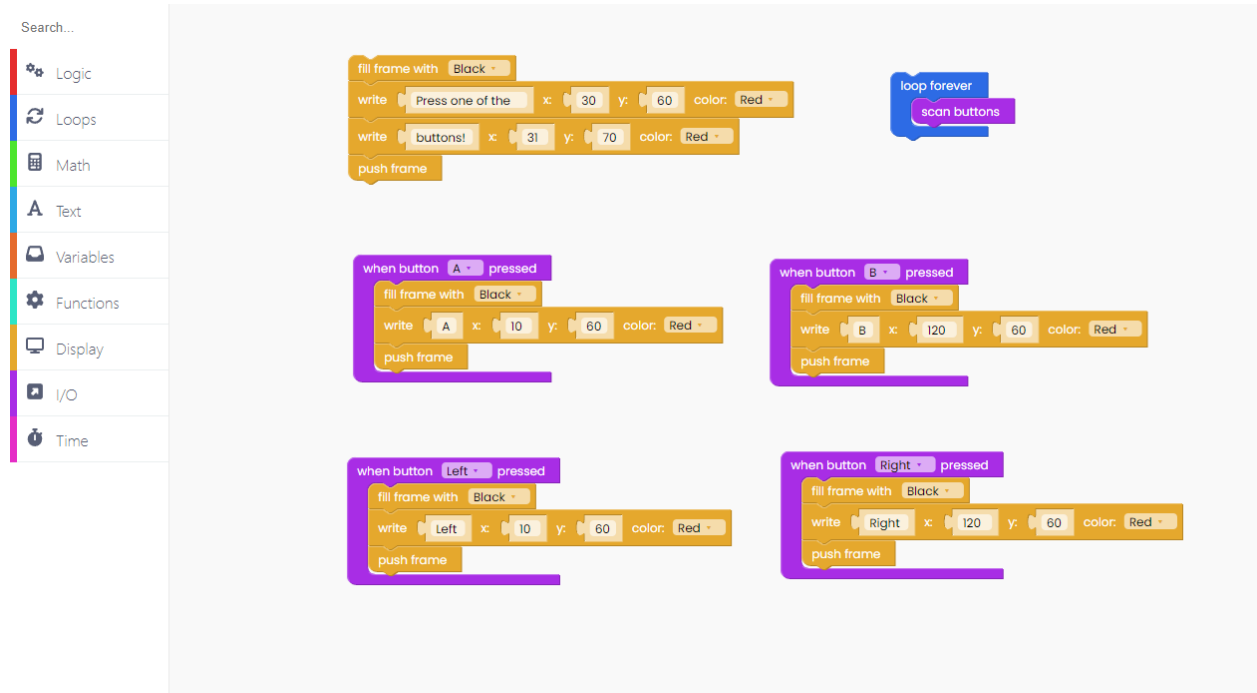
Repeat the process for the two remaining buttons:



The image shows a Scratch interface with a sidebar on the left and a workspace on the right. The sidebar contains a search bar and several categories: Logic, Loops, Math, Text, Variables, Functions, Display, I/O, and Time. The workspace contains five code blocks:

- A top block with three 'write' blocks and a 'push frame' block. The first 'write' block has text 'Press one of the', x: 30, y: 60, color: Red. The second 'write' block has text 'buttons!', x: 31, y: 70, color: Red.
- A 'when button A pressed' block with a 'fill frame with Black' block, a 'write A' block (x: 10, y: 60, color: Red), and a 'push frame' block.
- A 'when button B pressed' block with a 'fill frame with Black' block, a 'write B' block (x: 120, y: 60, color: Red), and a 'push frame' block.
- A 'when button Left pressed' block with a 'fill frame with Black' block, a 'write Left' block (x: 10, y: 60, color: Red), and a 'push frame' block.
- A 'when button Right pressed' block with a 'fill frame with Black' block, a 'write Right' block (x: 120, y: 60, color: Red), and a 'push frame' block.

Don't forget to add the "loop forever" and "scan buttons" block, otherwise your code won't work.



Click on the Run button, and check and start playing with the buttons.

Buzz around!

Let's learn what to do with the buzzer soldered to your CircuitPet.

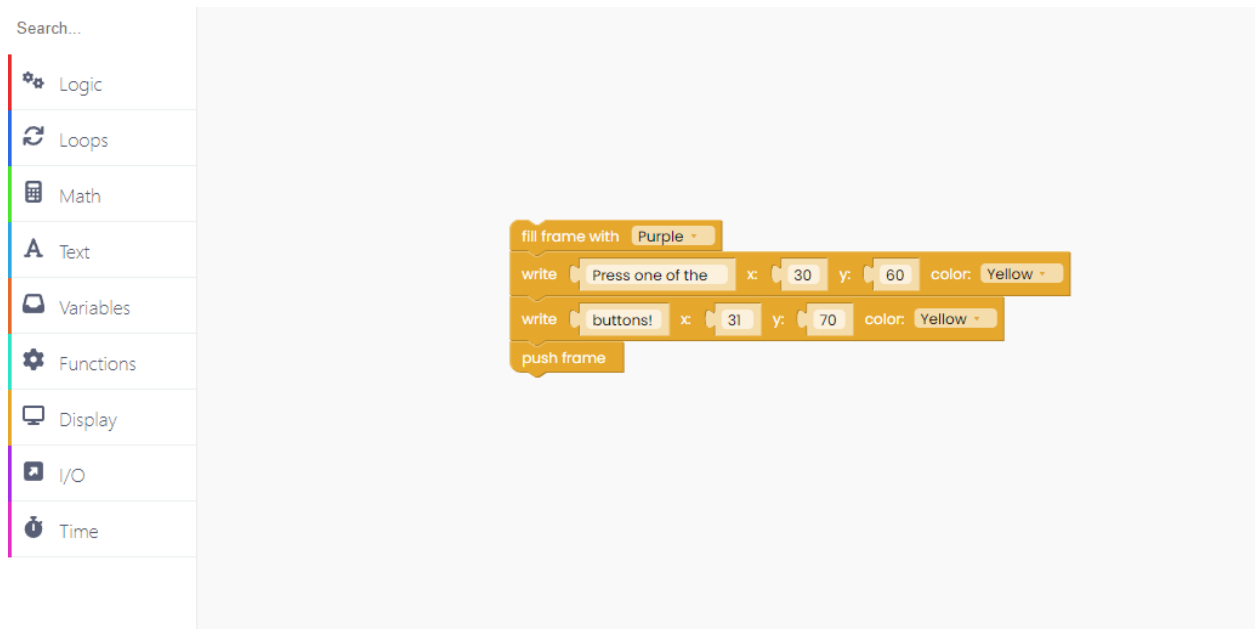
As the word itself says, the buzzer is used to make buzzing sounds.

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!

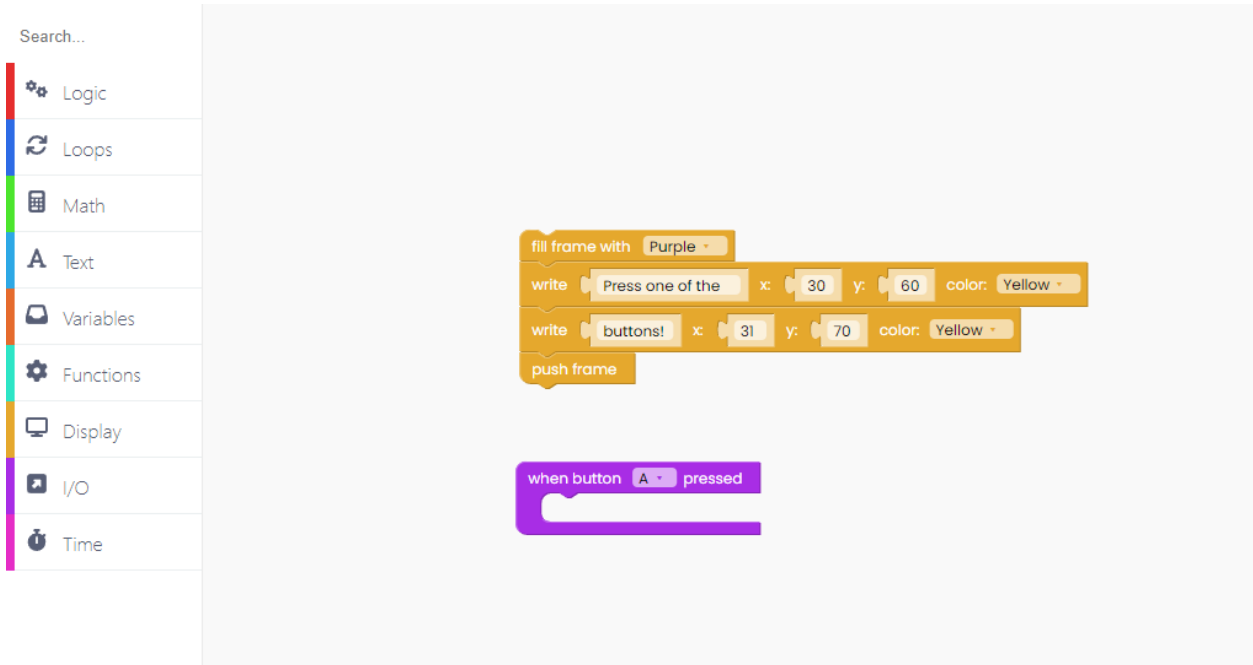
To begin with, let's write the instructions text that will appear on the screen once the code is executed.

The process is the as in the previous example.

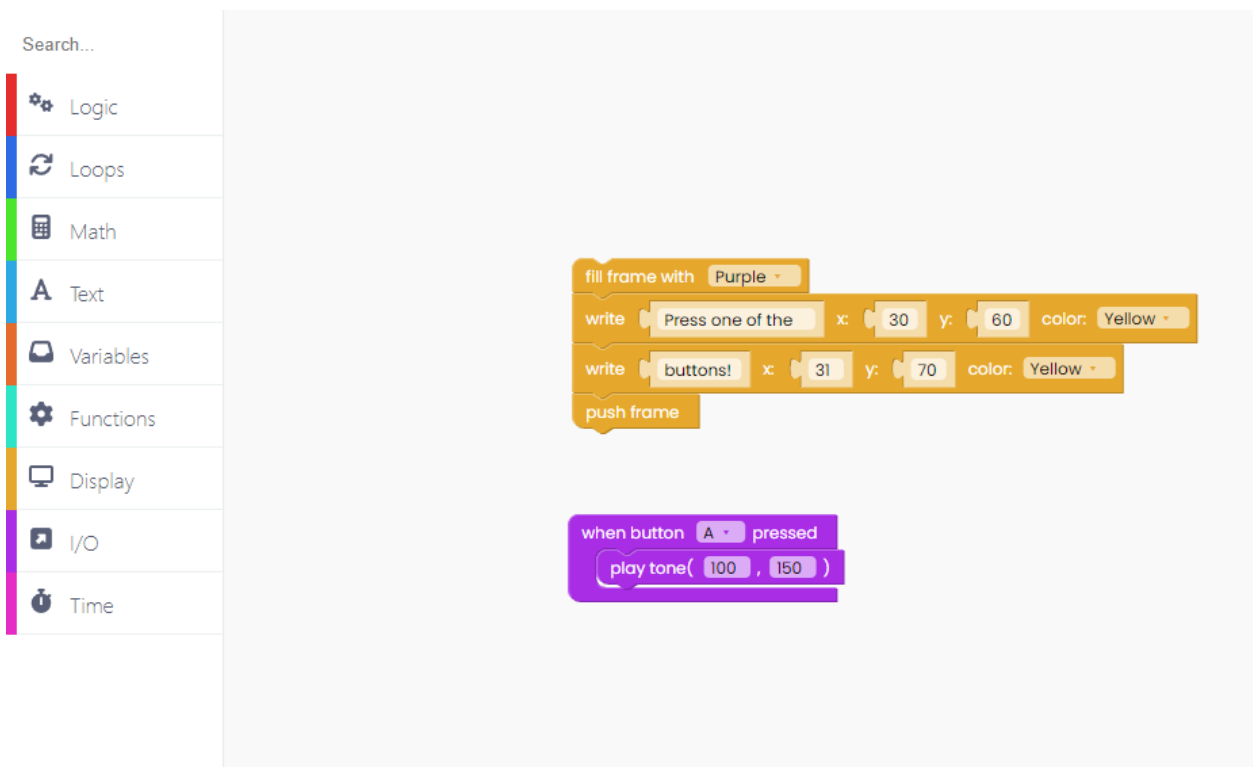


Now that we've created a text, let's change what happens with the buzzer when a specific button is pressed or released.

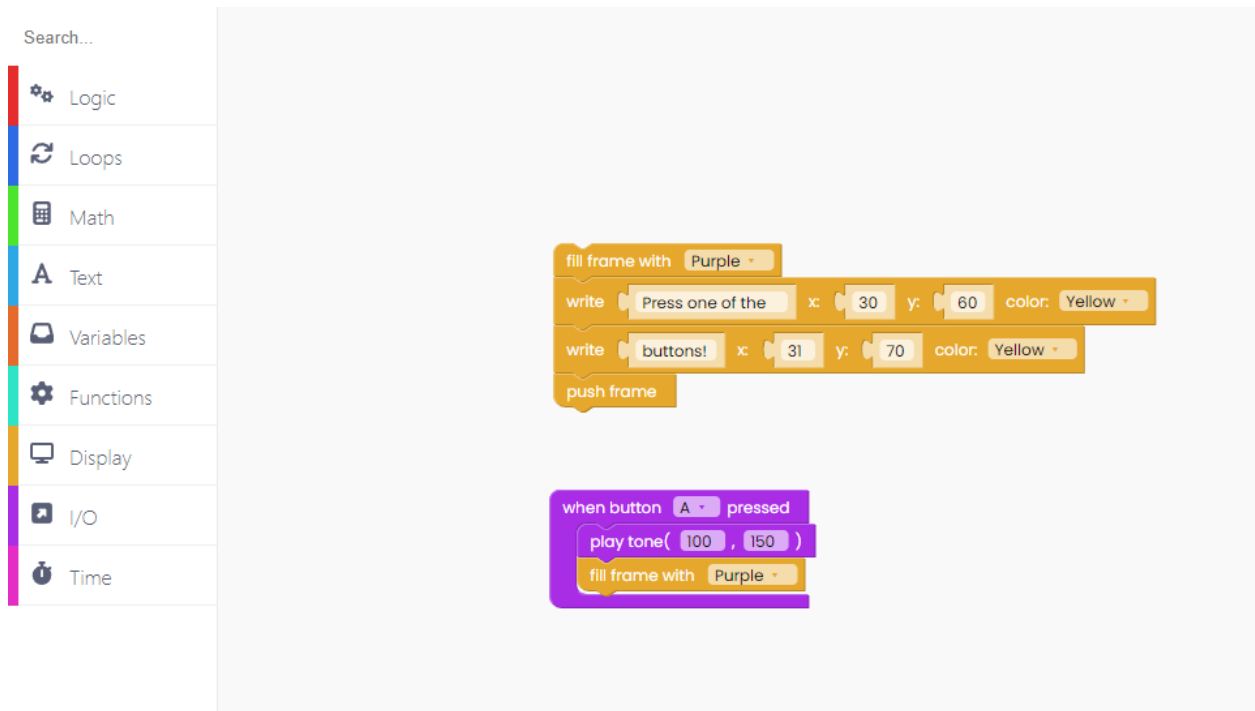
As in the previous sketch, we'll use I/O blocks to determine what happens when specific buttons are pressed.



Let us introduce you to a new block called "Play tone with frequency 100 Hz for 150 milliseconds".



Let's clear the display in purple once we press the A button.



You can choose what will be written on the display once the button is pressed.



For example, the screen can say "buzz" when we press the A button.

Don't forget about the coordinates!

Let's make "x" 65 and "y" 60.

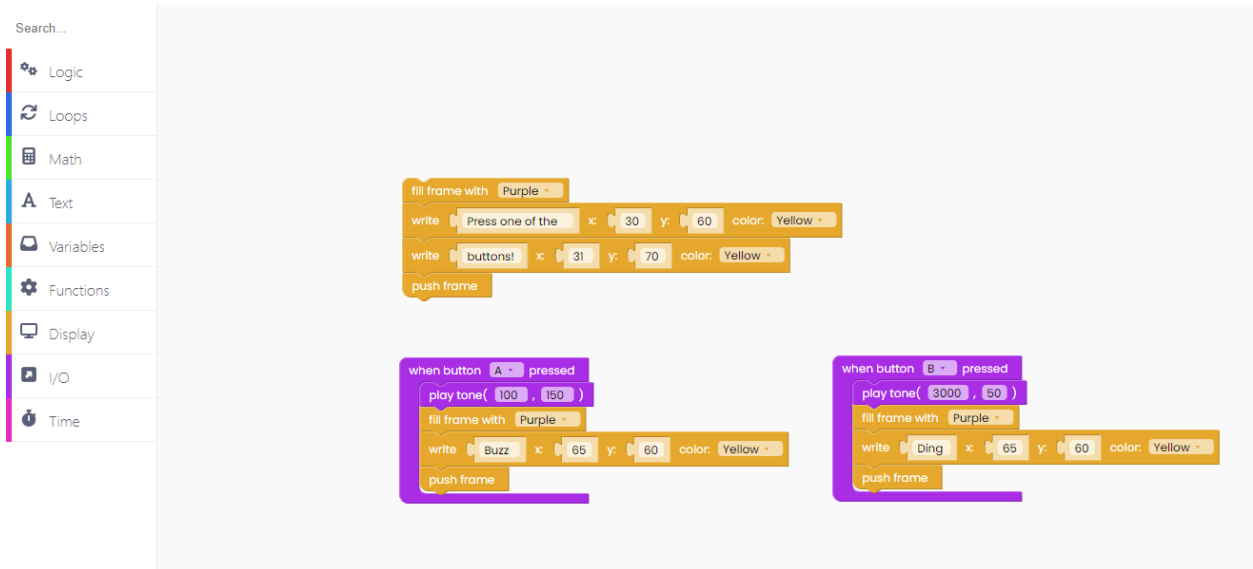
And for the end, drag the "push frame" block for a sketch to work.

First button - DONE!

Since we'll need the same blocks, we can easily duplicate them.

Let's do one button at a time.

The first one we'll do is for when a B button gets pressed. We decided to play tone with a frequency of 3000 Hz this time, and change the time to 50 ms.

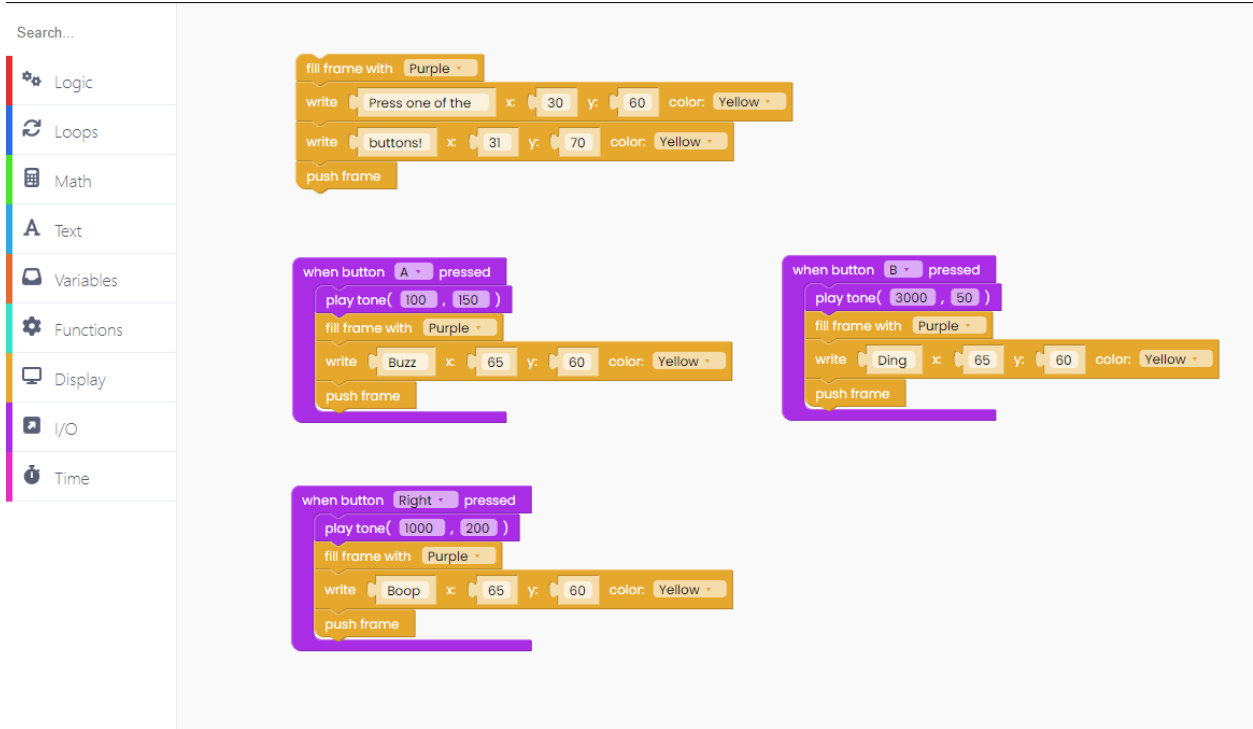


The screen's color stays purple, but we'll print "Ding" this time.

Let's duplicate the block again.

Now, we'll use a right button for playing tone with a frequency of 1000 Hz for 200 milliseconds.

In the meantime, on the screen will be written "Boop". Note that we kept coordinations the same for all prints.



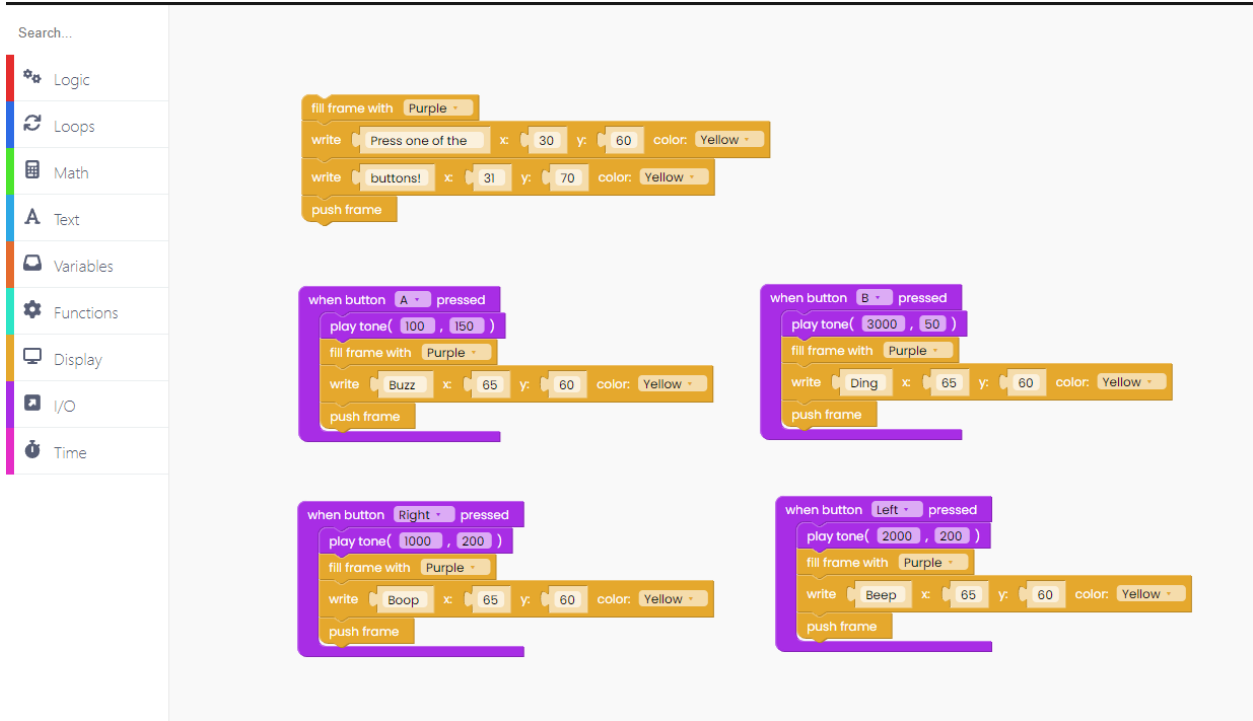
Yes, you guessed it right!

It's time to duplicate some more blocks.

The last button we'll use is the left one.

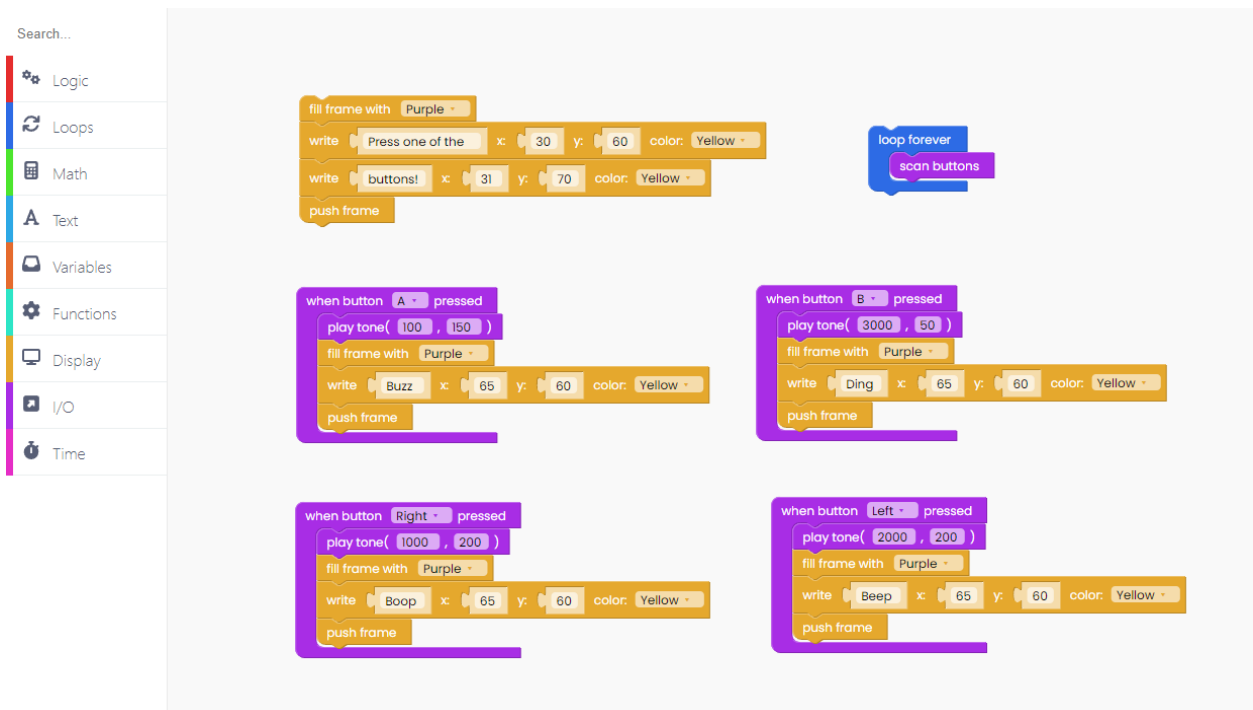
If you press the left button, the buzzer will play tone with a frequency of 2000 Hz for 200 milliseconds.

While the tone is playing, the screen will say "Beep".



Great! We used all four buttons from CircuitPet.

As always, don't forget to use the "loop forever" and "scan buttons".

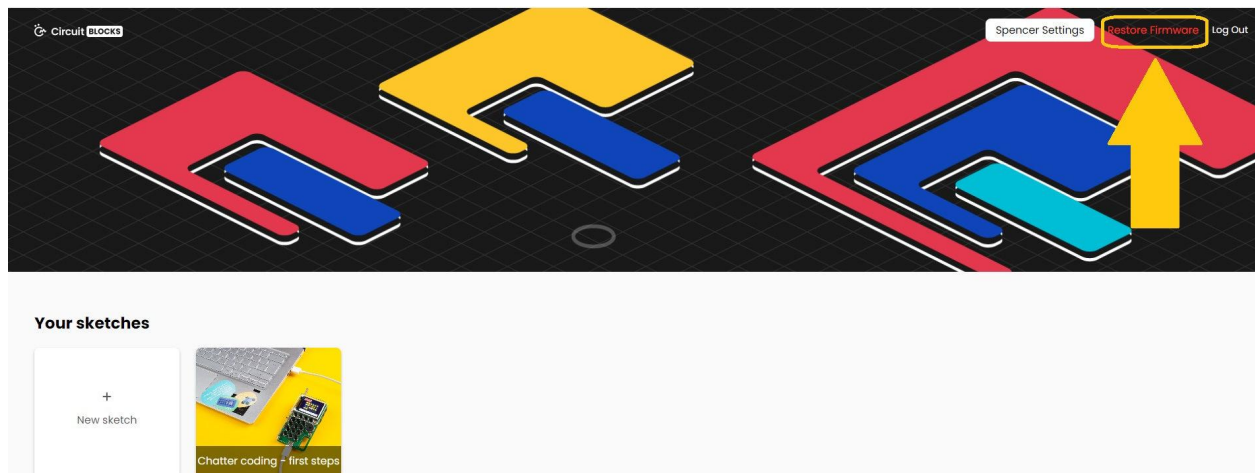


Click on the Run button and hear those sounds!

Restore CircuitPet's firmware

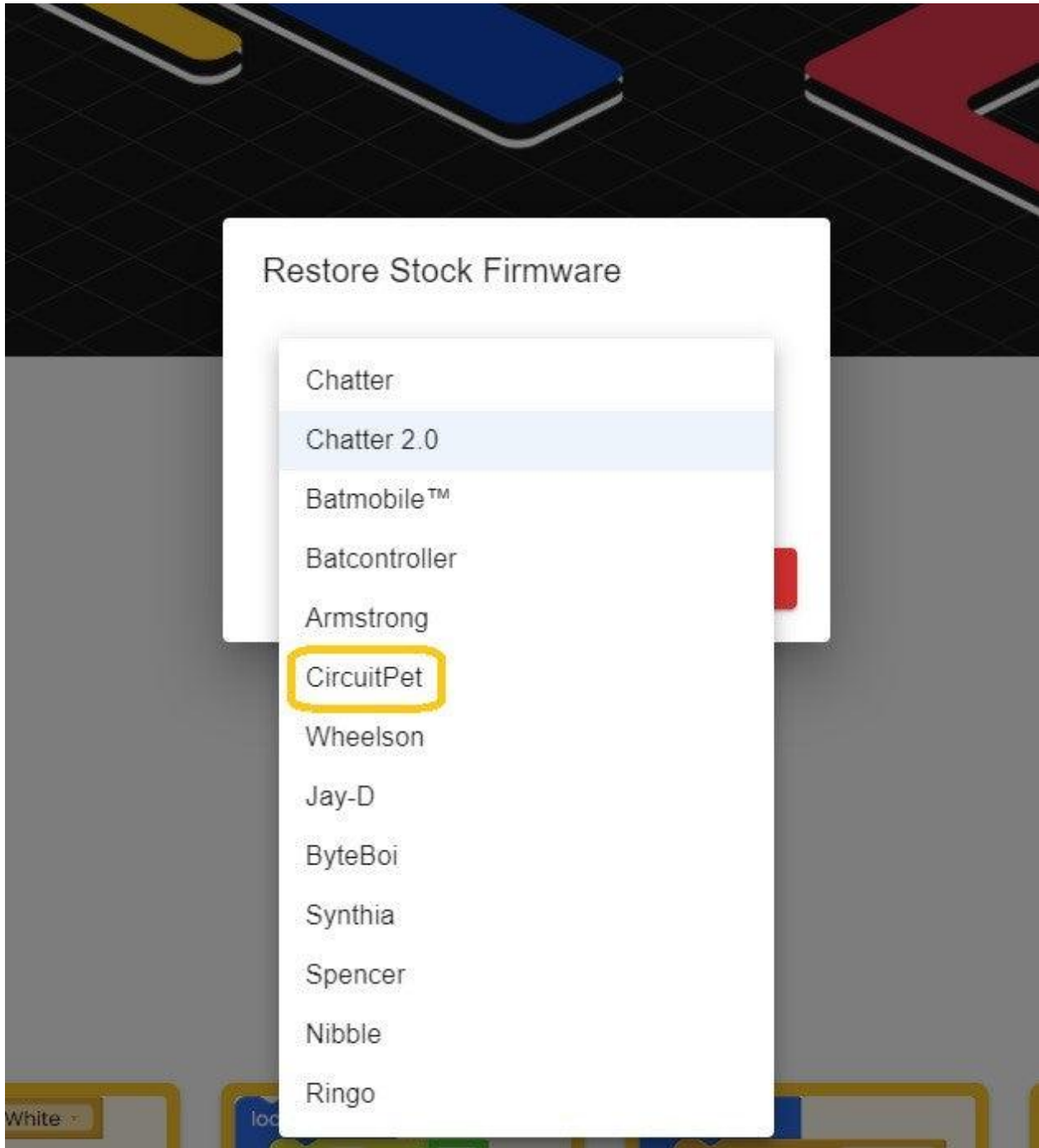
If you want to restore CircuitPet's firmware, follow these steps.

Just connect your CircuitPet 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 CircuitPet, of course.



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

What's next?

You've reached the end of first CircuitPet's coding tutorial, congratulations!



We hope you're as excited as we are about CircuitPet'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 CircuitPet 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!

