Chatter 2.0 coding - first steps

Introduction

The beginning

Welcome to the Chatter 2.0 coding tutorial

Thank you for supporting CircuitMess, and welcome to the Chatter 2.0 coding tutorial.

We'll use **CircuitBlocks** for coding your newly-assembled encrypted wireless communicators.

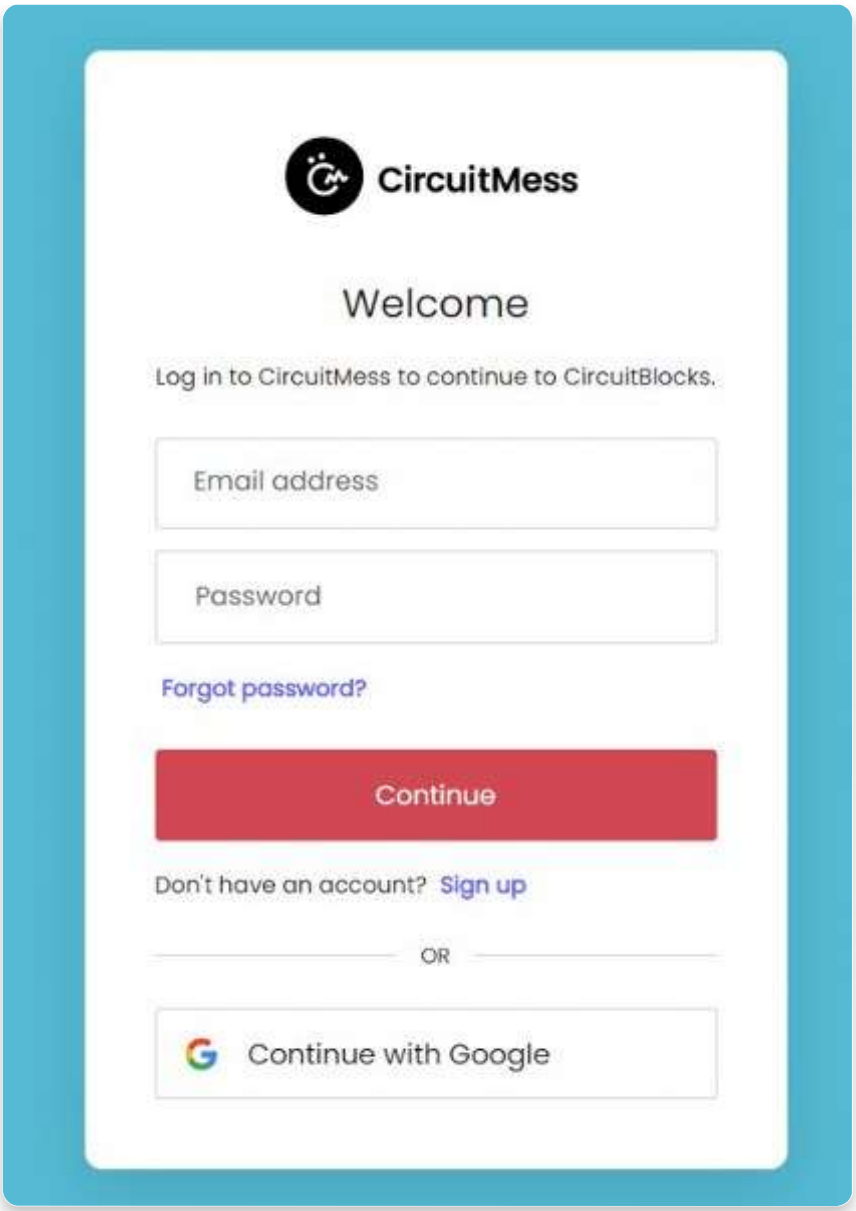
CircuitBlocks is a custom-made coding app that we've designed.

You will code your Chatter 2.0 in CircuitBlocks' graphical block-based coding interface that will help you make your first steps in the world of physical computing.

Go to **https://code.circuitmess.com/** to **access CircuitBlocks**.

You must create an account or use your CircutiMess account if you already have one.

You can also sign up with your **Google account**.



This is what you'll see after you sign up:

The installation and user interface will be explained in the following chapters.

Installation

User interface

When you enter CircuitBlocks, you will see a window that looks like this.

Starting a **new project (also known as a "sketch")** is as simple as clicking the '**New sketch**' button.

Saved sketches will appear next to that button, and you can access them anytime.

If you have any **problems** with CircuitBlocks, please contact us at **contact@circuitmess.com,** send a **screenshot** of your sketch, and explain what went wrong, and we will assist you immediately.

Creating a new project (sketch)

Press on the big "**New sketch**" button.

You'll get an option to name your sketch.

We named our first sketch "Write on the display" because that's what we are going to do for the first example.

When you click on create, you'll see the following screen:

Before you can code on your Chatter, you must first complete the following steps:

1. **Connect** it to the PC using the **USB** cable from the kit.

2. **Turn Chatter on.**

3. Click on the **More info** button.

4. Choose **Download the agent**.

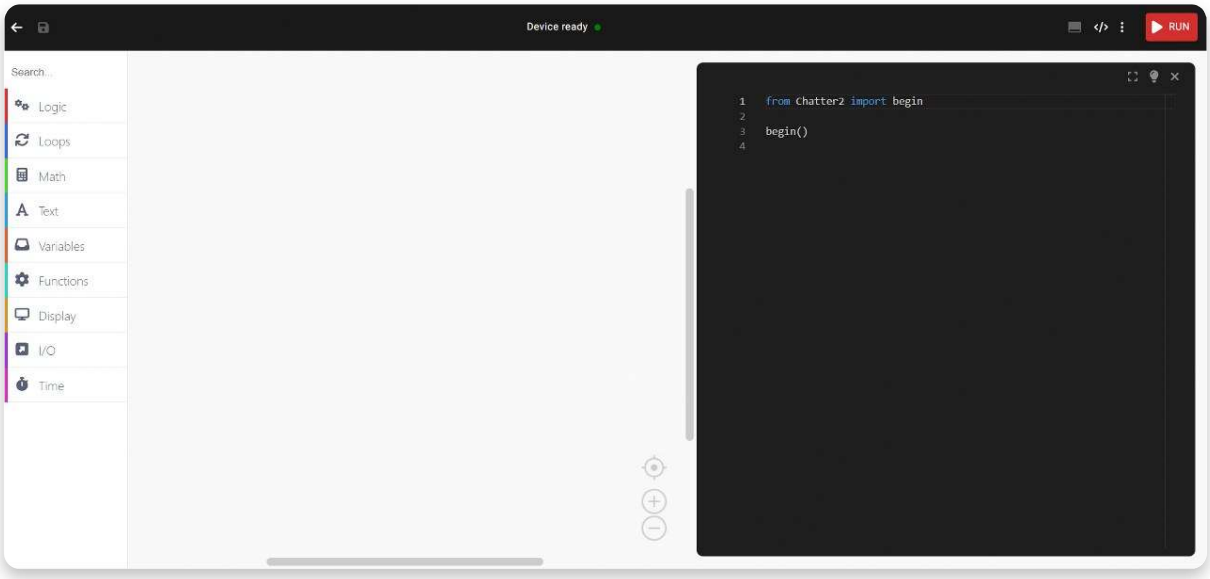
5. **Install CircuitBlocks Agent**.

After you install it, it should connect automatically, and this is the message that will appear on your screen:

6. **Install MicroPython on your Chatter.**

Click **finish,** and you're good to go!

The basics



Now that we are ready to start coding, we should become familiar with the user interface and the block sections.

On the top of the screen, there is a **toolbar** with a few buttons.

The **block selection bar** is located on the far left - you can take the blocks from there and drop them into the "drawing" area in the middle of the screen.

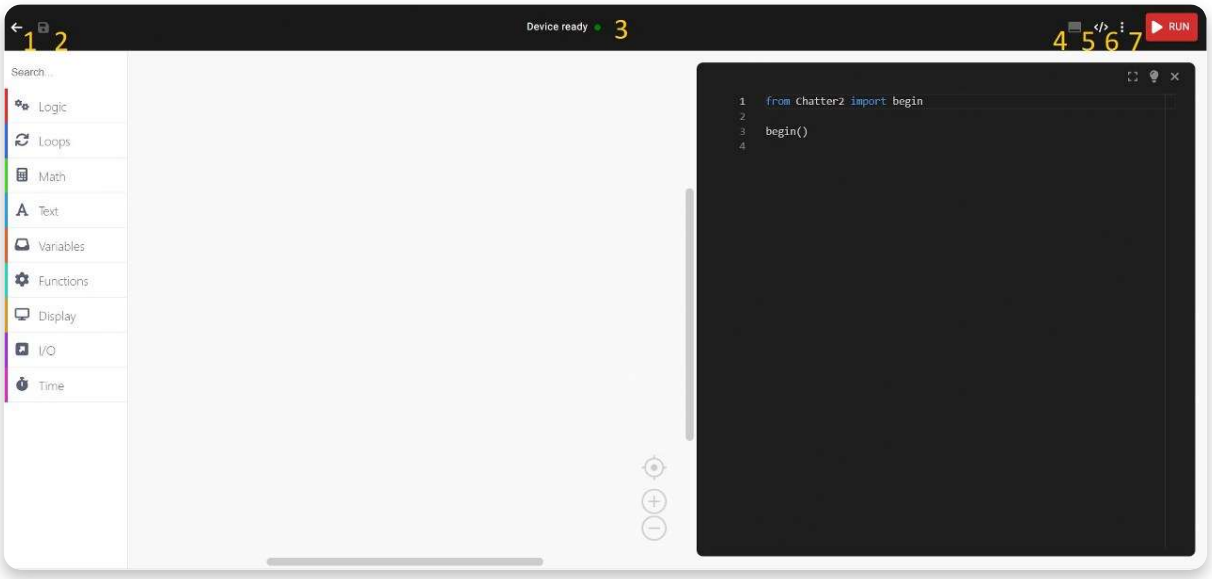
In the middle of the screen is where you'll be "drawing" your code with colorful blocks.

On the right side of the screen, you will see **code written in MicroPython** appear magically by itself when you drag and drop the colorful blocks.

Here, you can drag and drop colorful blocks that represent parts of code and see what your program would look like in MicroPython. When you get skilled enough,

you will be able to switch directly to textual coding in MicroPython without the need for colorful blocks.

Toolbar

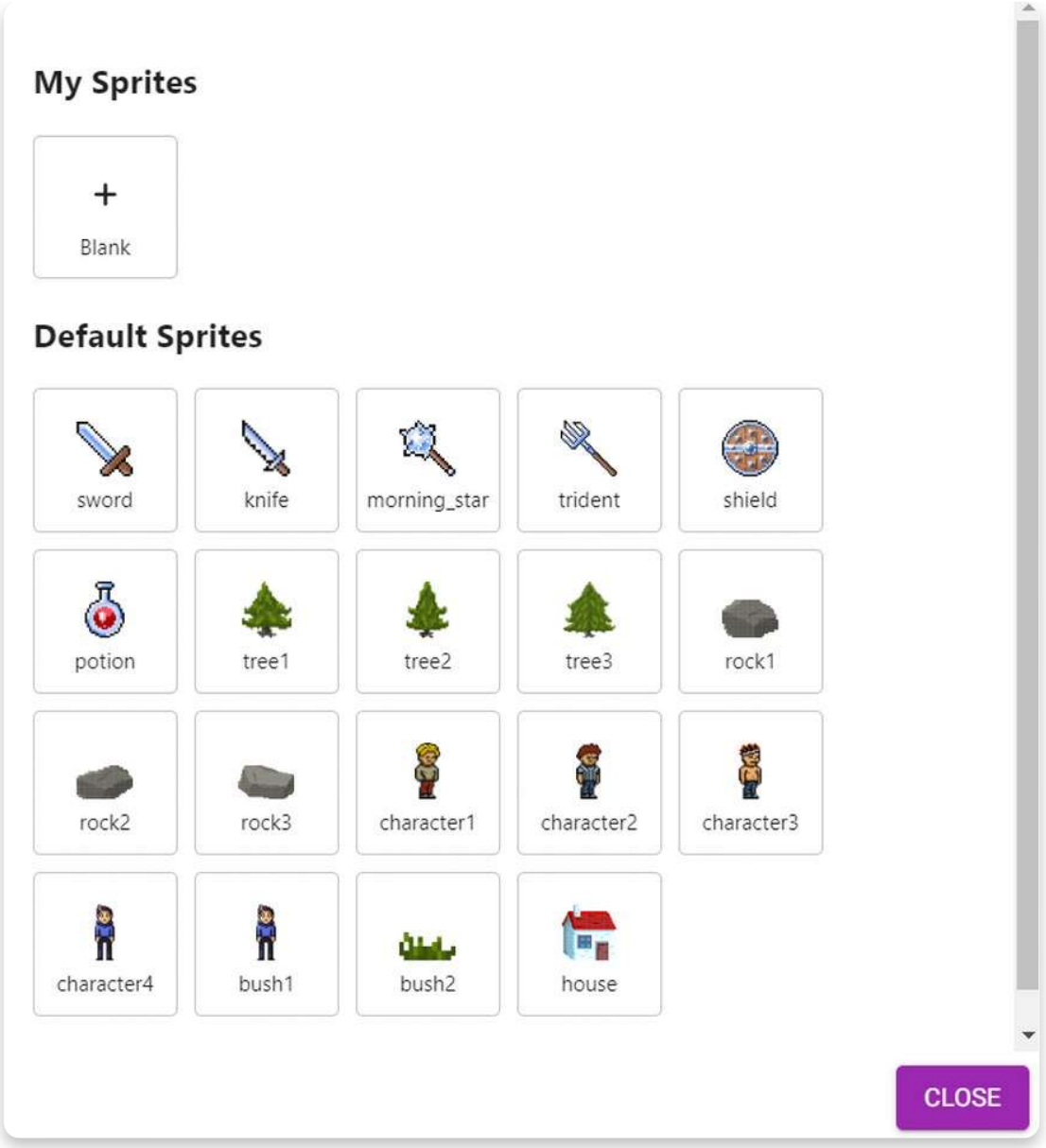


Here's a short explanation of what the buttons in the window toolbar do:

1. **Back to the main menu** – returns you to the home screen without saving
2. **Save/Save As** – saves your sketch, make sure to press this button from time to time, and before closing CircuitBlocks
3. **Chatter 2.0 connection indicator** - the red dot turns green if your Chatter 2.0 is connected to your computer via a USB cable
4. **Serial monitor** - this button opens a window that we call the "Serial monitor". "Serial" is a nickname for a type of communication that is happening between Chatter and your computer. In this window, you will later be able to see the messages sent from Chatter to your computer via the USB port.
5. **Hide code editor** - to hide the right side of the user interface showing the code in MicroPython.
6. **More options** - you can click here to enter Sprite Editor, Device Connection Info, Soft Reboot, Restore Stock Firmware, and to Log out.
7. **Run** - This button will translate the code you have constructed in CircuitBlocks to *machine code* that Chatter understands (beep boop beep boop 1011100101) and send the code to your Chatter via the USB port

Let's go over that three dots button.

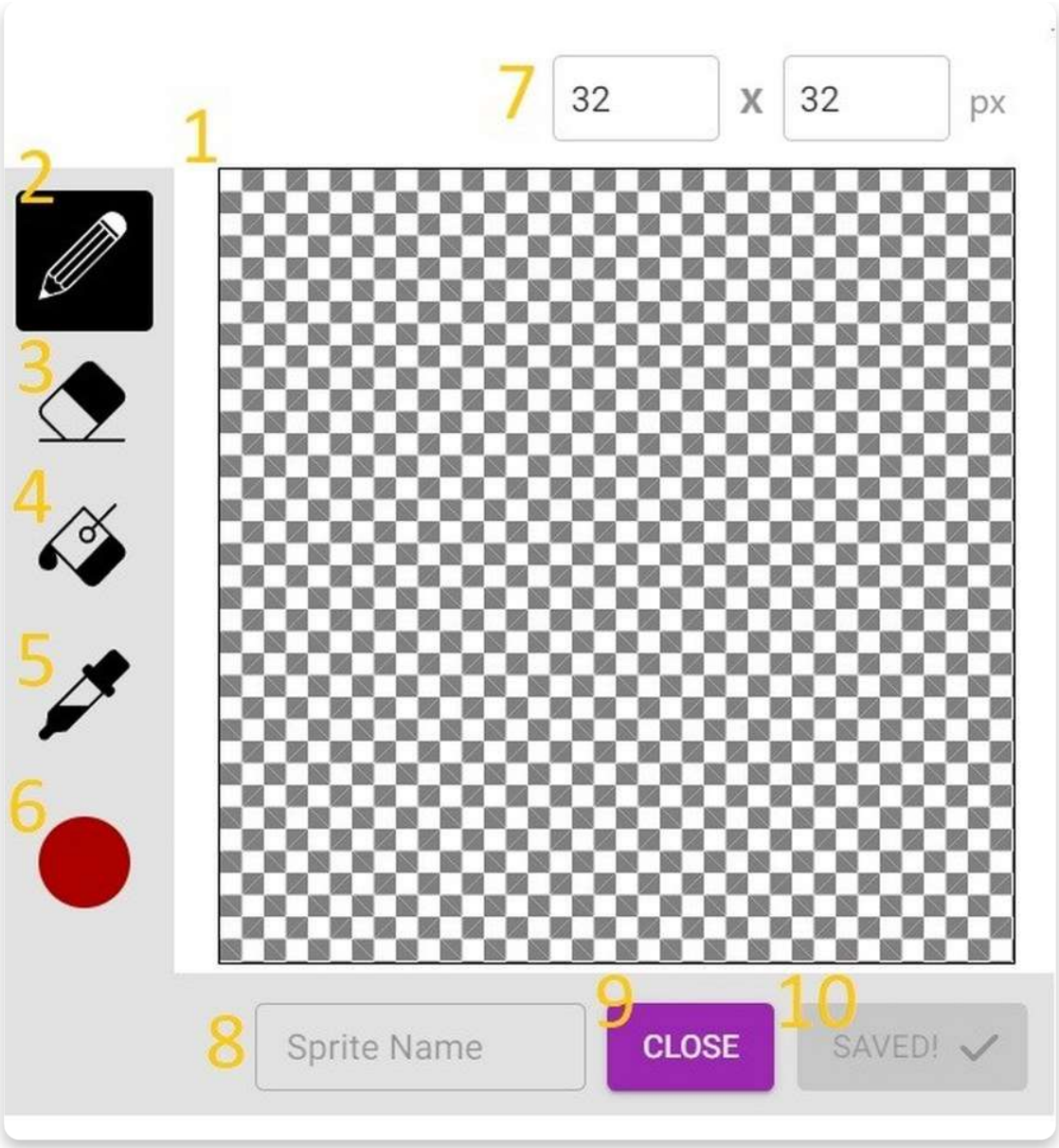
The first thing you can enter is the Sprite editor that looks like this:



Here you can choose from a variety of different premade sprites or create your own sprite.

Also, if you click on one of the premade sprites, you can change it!

This is what it looks like when you start creating your own sprite:



Let's see what each icon means:

1. **Drawing section** - the section in which you will draw your sprite.

2. **Paint brush** - tool for drawing your sprite.

3. **Eraser** - to erase any unwanted mistakes.

4. **Paint bucket** - used to paint a selected area with a specific color or to paint an uncolored area with a specific color.

5. **Color dropper** - this tool is used to match any color in an image.

6. **Color picker** - to select a color you want to use.

7. **Resolution of the sprite** - this is the size of the sprite and it is measured in pixels. A pixel is the smallest controllable element of a picture represented on the screen.

8. **Sprite name** - you can give your sprite whatever name you want.

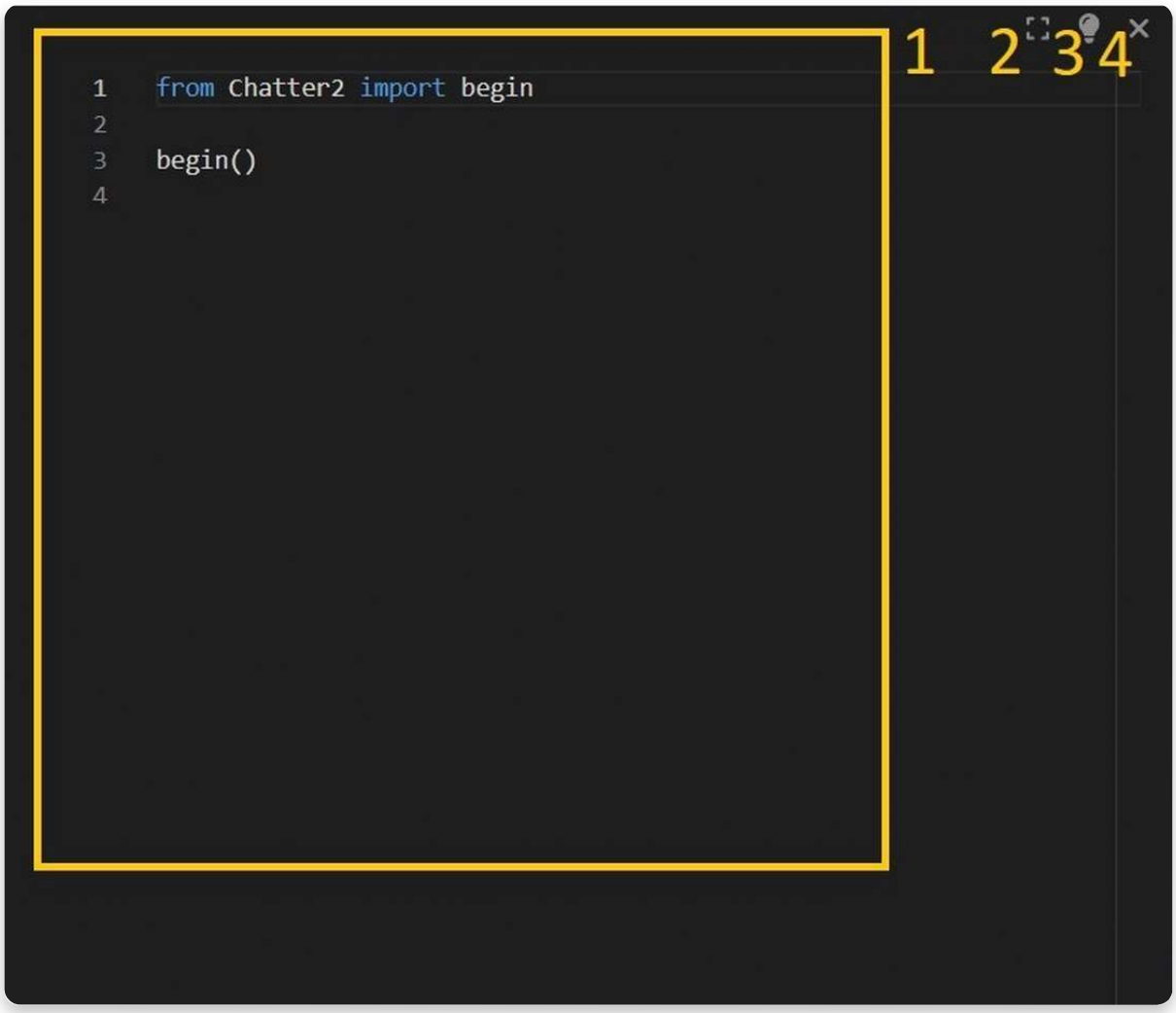
9. **Close** - for exiting the drawing area and saving the sprite.

10. **Save indicator** - this indicates whether or not the sprite or changes you made to the sprite were saved. After you save the sprite, it will appear in the My Sprites section.

Code window and drawing board

Let's go over all the buttons in the code window and drawing board!

Code window



The so-called "Code window" has the following parts:

1. **Main code screen** - code written in C++ will appear here as you drag and drop colorful blocks on the left side of the screen. You'll see that some parts of the code are colored in funny colors. Programmers call this *syntax highlighting.* Basically,

what is happening is that different categories of code commands are colored differently so that programmers can understand the code more easily.

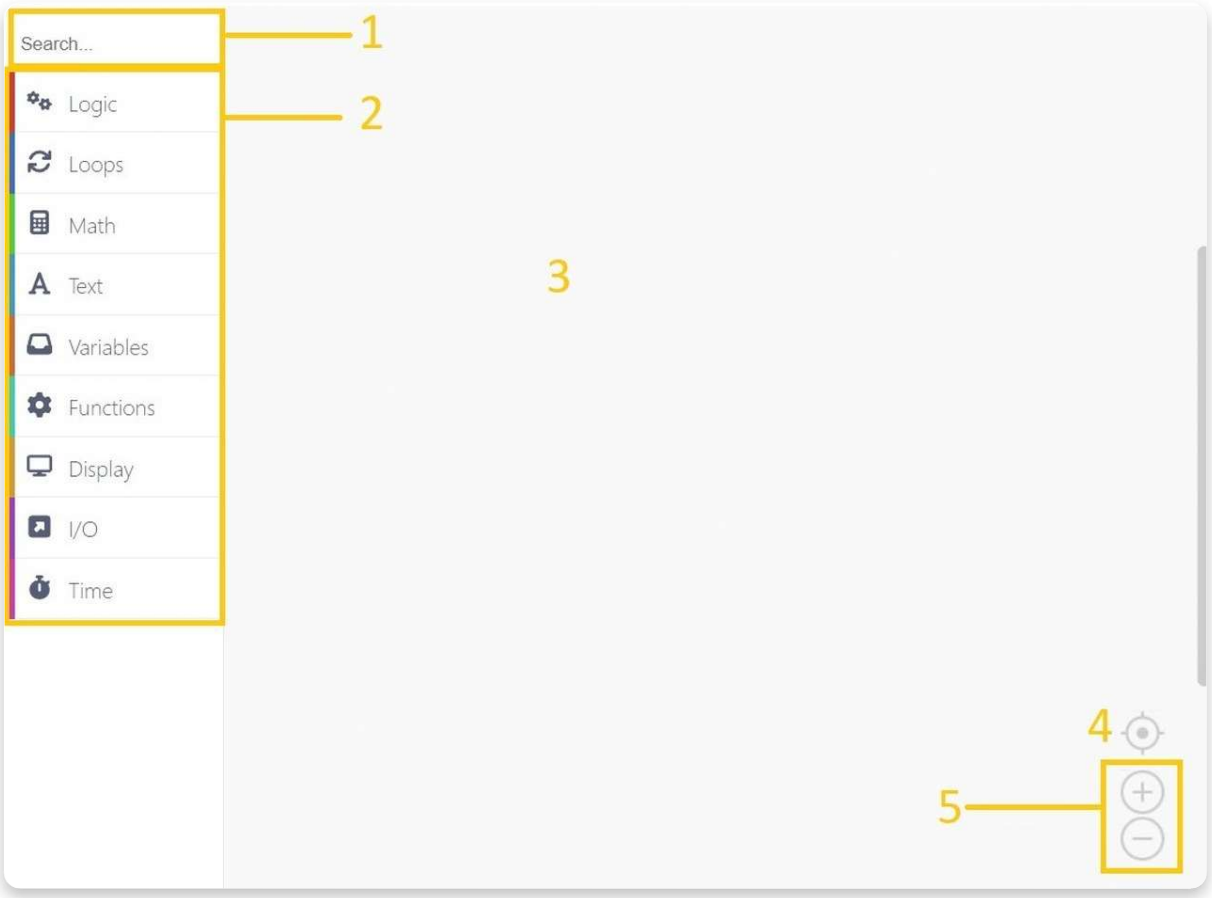
2. **Expand** - stretches the code window across the entire screen. Press it again to resize it to the half-screen again.

3. **Light/dark theme switch** - you can toggle the background and text color of the code window with this button.

4. **Close** – closes the code window, the same functionality as the toolbar's 'Close Code' button.

Drawing board

The drawing board is where the magic happens.



It has the following parts:

1. **Search bar** - type the component's name you are looking for here.

2. **Component selector** – the blocks are divided into different categories here. Each category has a specific color designated to it.

3. **Drawing area** – you will drag the blocks from the component selector and drop them into the drawing area. This is how code is made. Easy peasy!

4. **Center tool** - if you get lost when scrolling across the drawing area, press this button, and it will center your view on the blocks you have dropped on the drawing area.

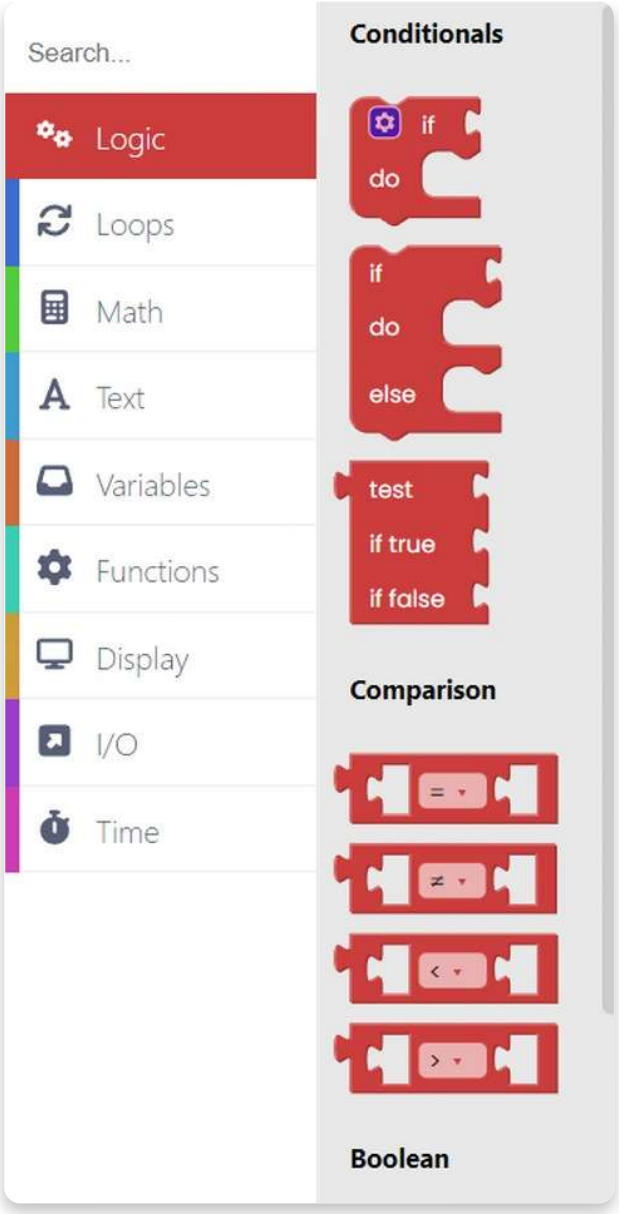
5. **Zoom buttons** - to zoom in and out of the drawing area.

Block sections

There are a total of nine sections in CircuitBlocks. We've organized them so that you'll be able to find everything in a maximum of two clicks.

The sections themselves are pretty self-explanatory, but we'll go through them all to get a little better understanding of the whole concept.

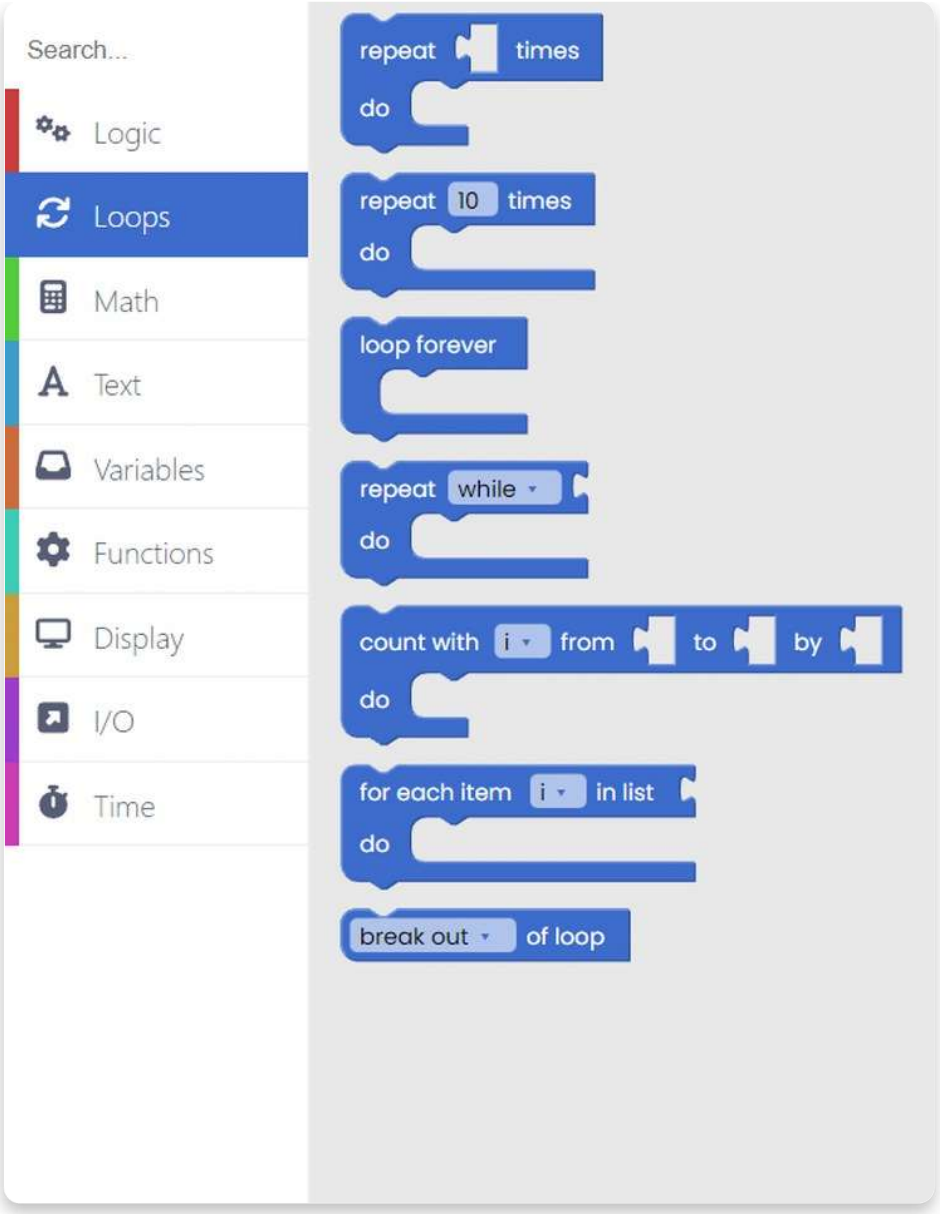
Logic



This is where the base of every code is located.

Every **if, if-else, else** function, comparisons, **and/or, not, true/false** and other logical operators.

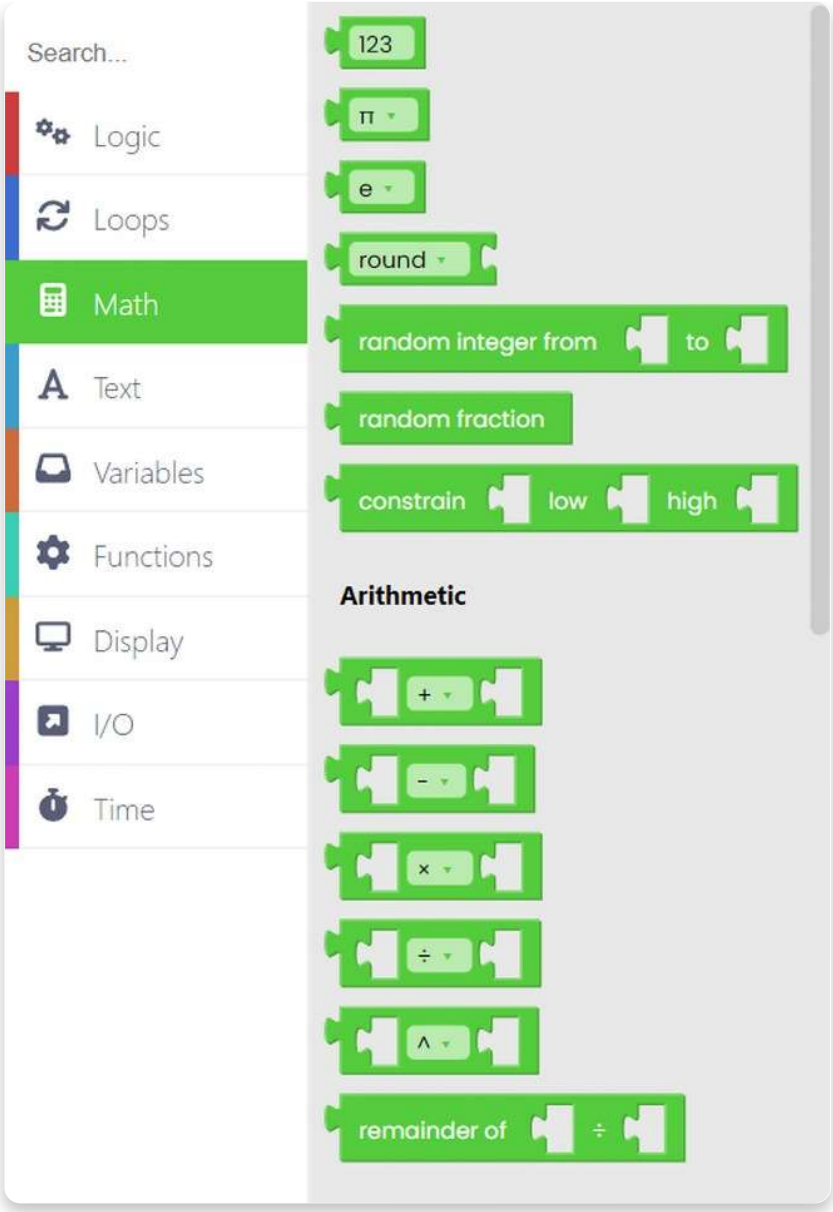
Loops



Loops are functions that repeat everything inside for a specific amount of time.

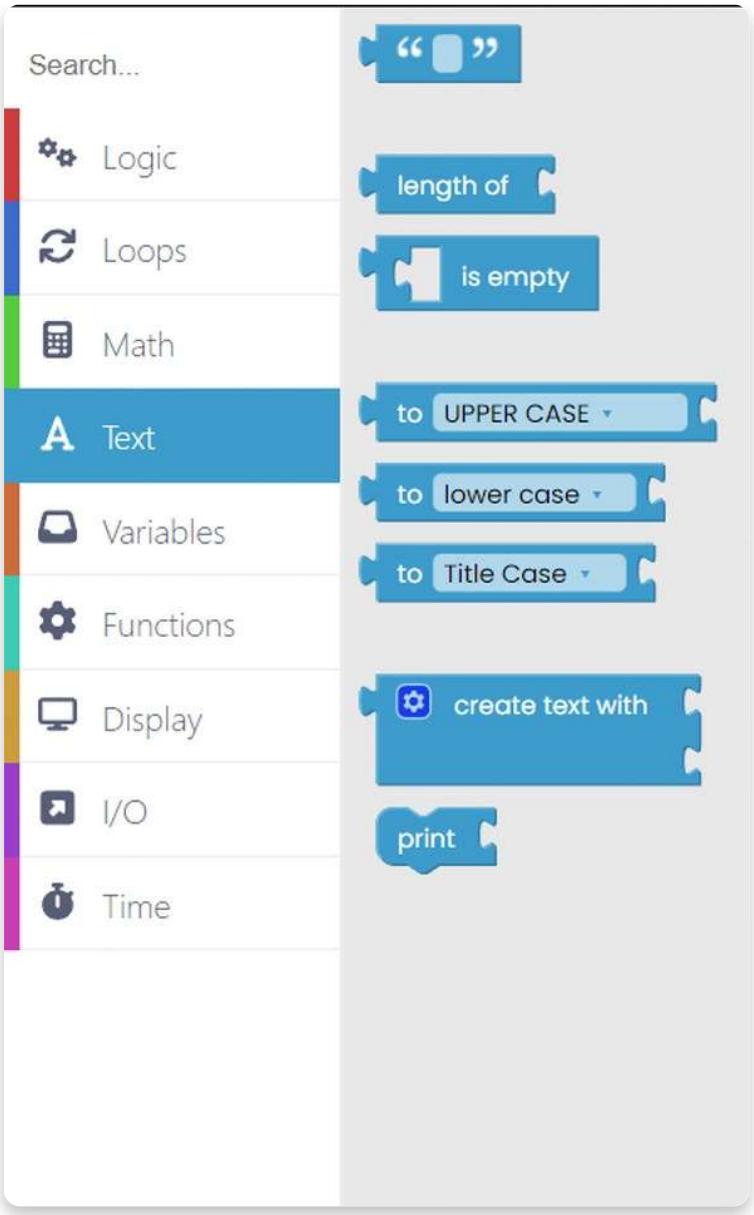
They can have conditional and repeat for as long as that condition is met or have a pre-determined number of repeats.

Math



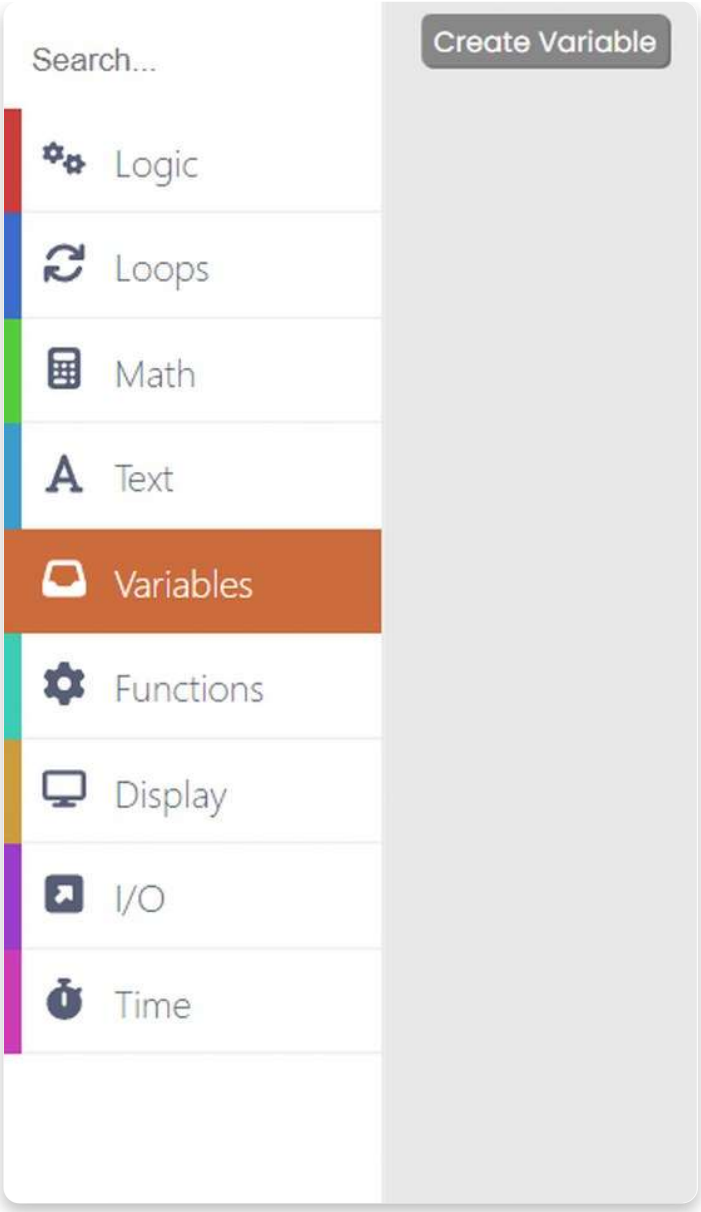
Pretty much every math function is located here. From basic operations to rounding numbers and working with angles, you will easily trigger your inner Einstein or Pythagoras in a matter of seconds!

Text



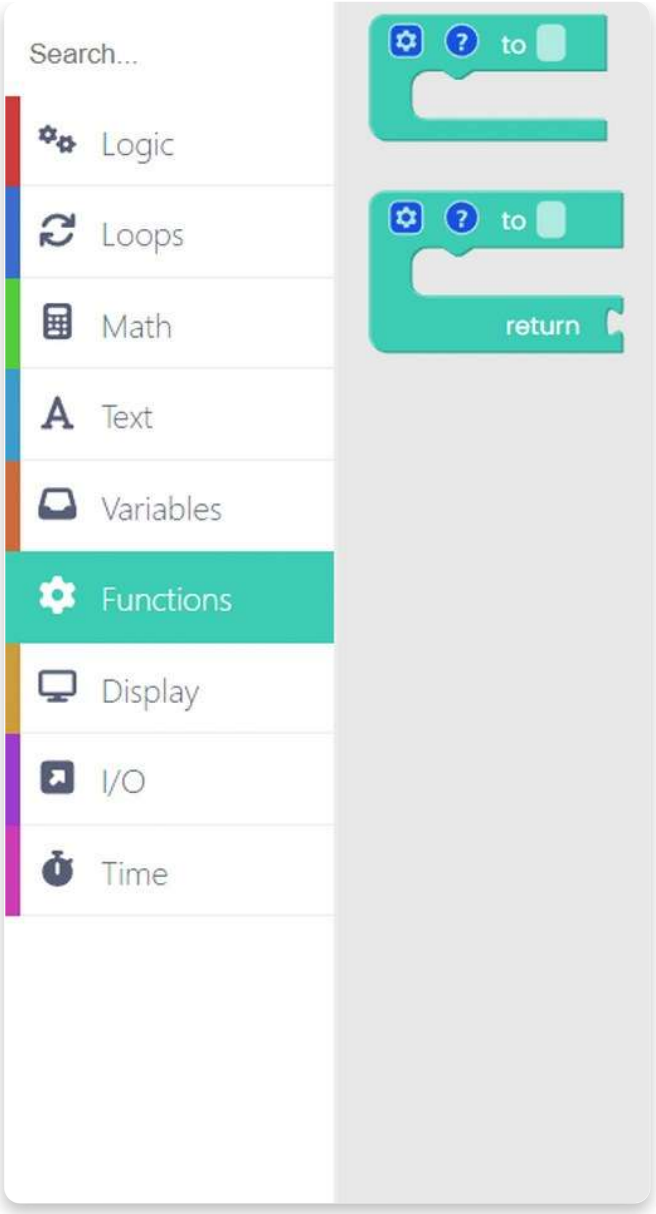
Strings, characters, and string manipulation. Great place for creating new text and implementing it in your programs.

Variables



Create a variable of any type and set its name and desired value. CB will automatically recognize the variable type (int, double, string, boolean), so you don't need to worry about that.

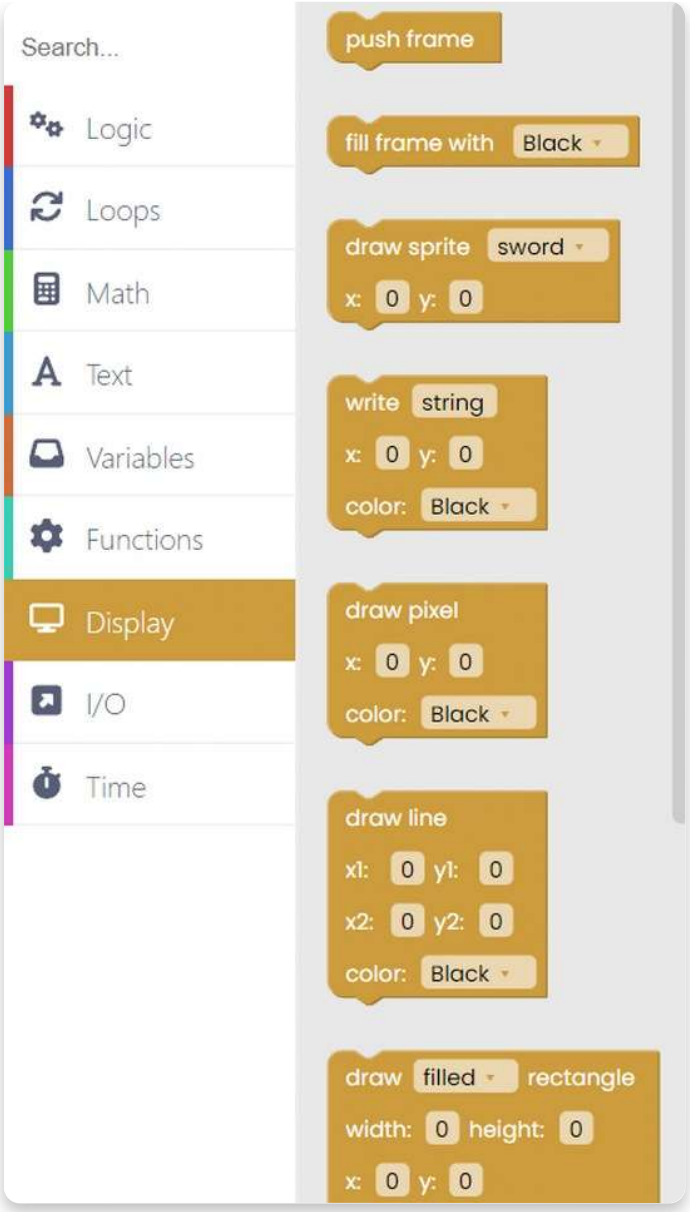
Functions



The Default Arduino function (explained on the previous page) is located here.

You can also create your functions which can then be inserted as one of the main parts of your program.

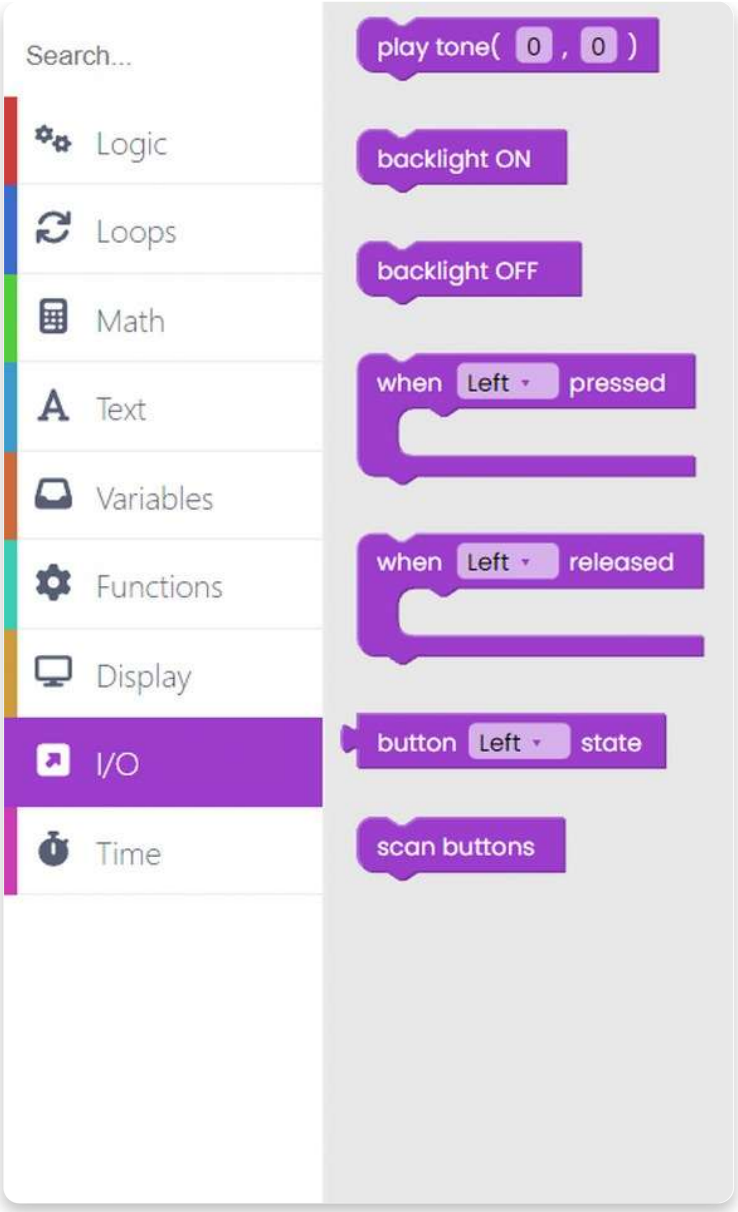
Display



Well, all these blocks are really not important if you don't see anything on the screen!

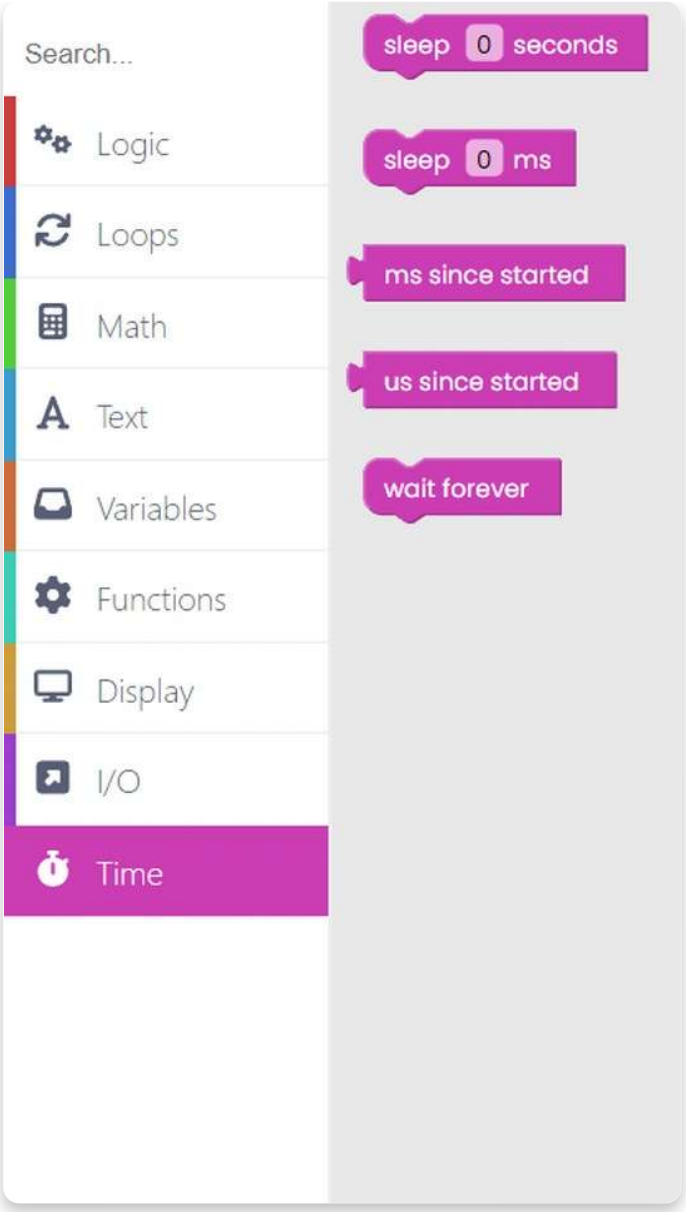
Here is where all the magic translates to those colored pixels. You can create so much through these blocks.

Input/Output



Everything regarding Chatter's components is located here.

Time



Delays, timers, and other time-related stuff, great for creating cool animations and video games.

Search bar

There is also a **search bar** above all function sections to ease the search for that one specific block you just can't seem to find.

Just type in whatever comes to your mind, and all blocks that have anything to do with the written word will be shown on the right-hand side.

Now, you really can't say that it's impossible to find something.

You've learned everything about the blocks!

It's time to move on to the next lesson...

Let's start! Step by step!

Let's write on the display!

Let's get down to business!

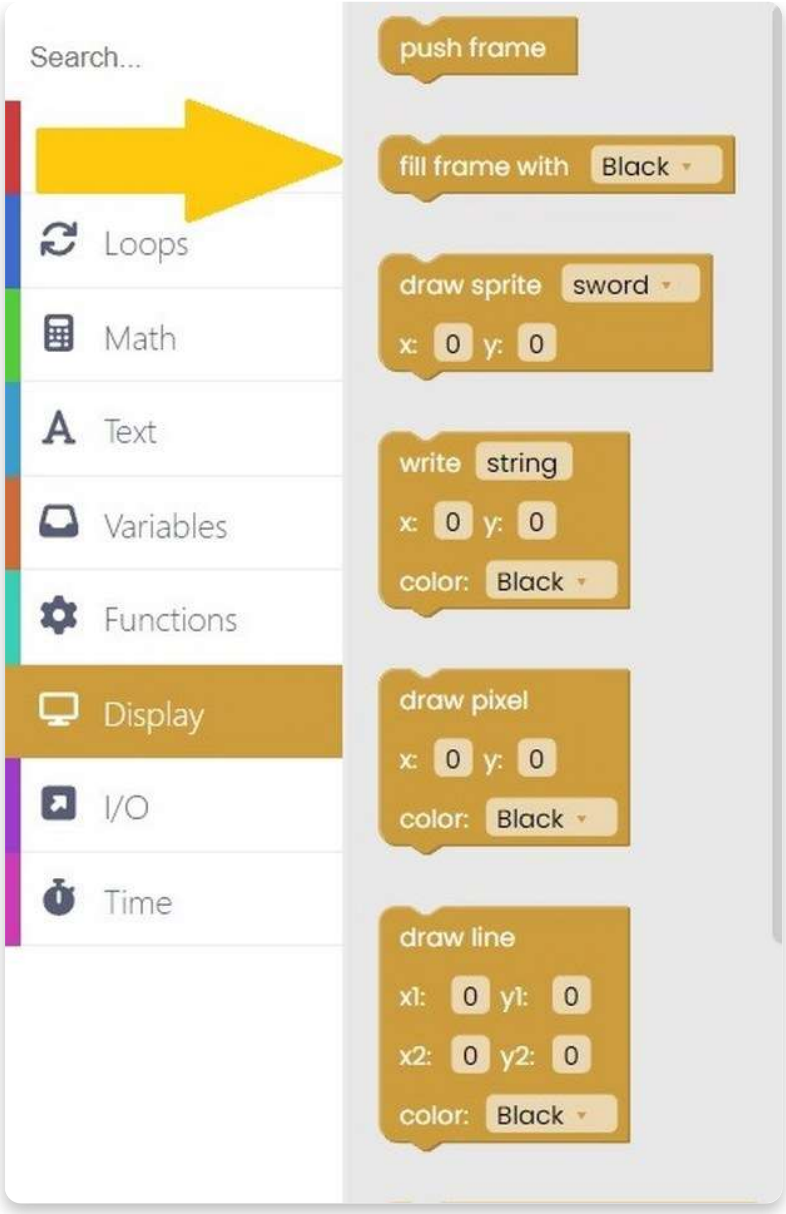
Before doing everything, make sure you have completed all the preceding steps: you have downloaded CircuitBlocks Agent and MicroPython, connected your device to the PC, and turned it on.

The first thing we'll learn is how to write text on Chatter's display.

You only need a Display block section to write something on Chatter's display. It's quite simple!

Please click on the mentioned section, and choose a "**fill frame with**" block.

We'll use this block to choose the background color that will appear once we run this sketch.



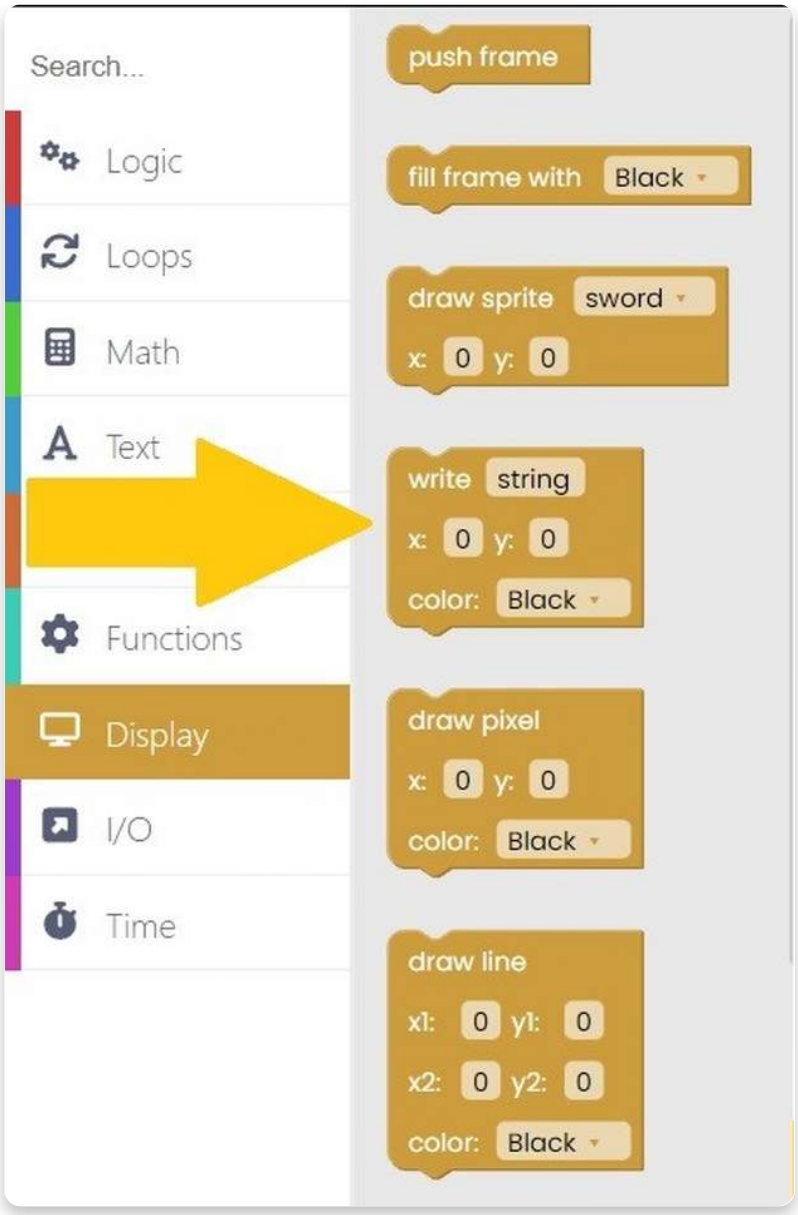
Drag and drop the selected block into the drawing area.

You can choose whichever color you like, but this time, we chose navy.

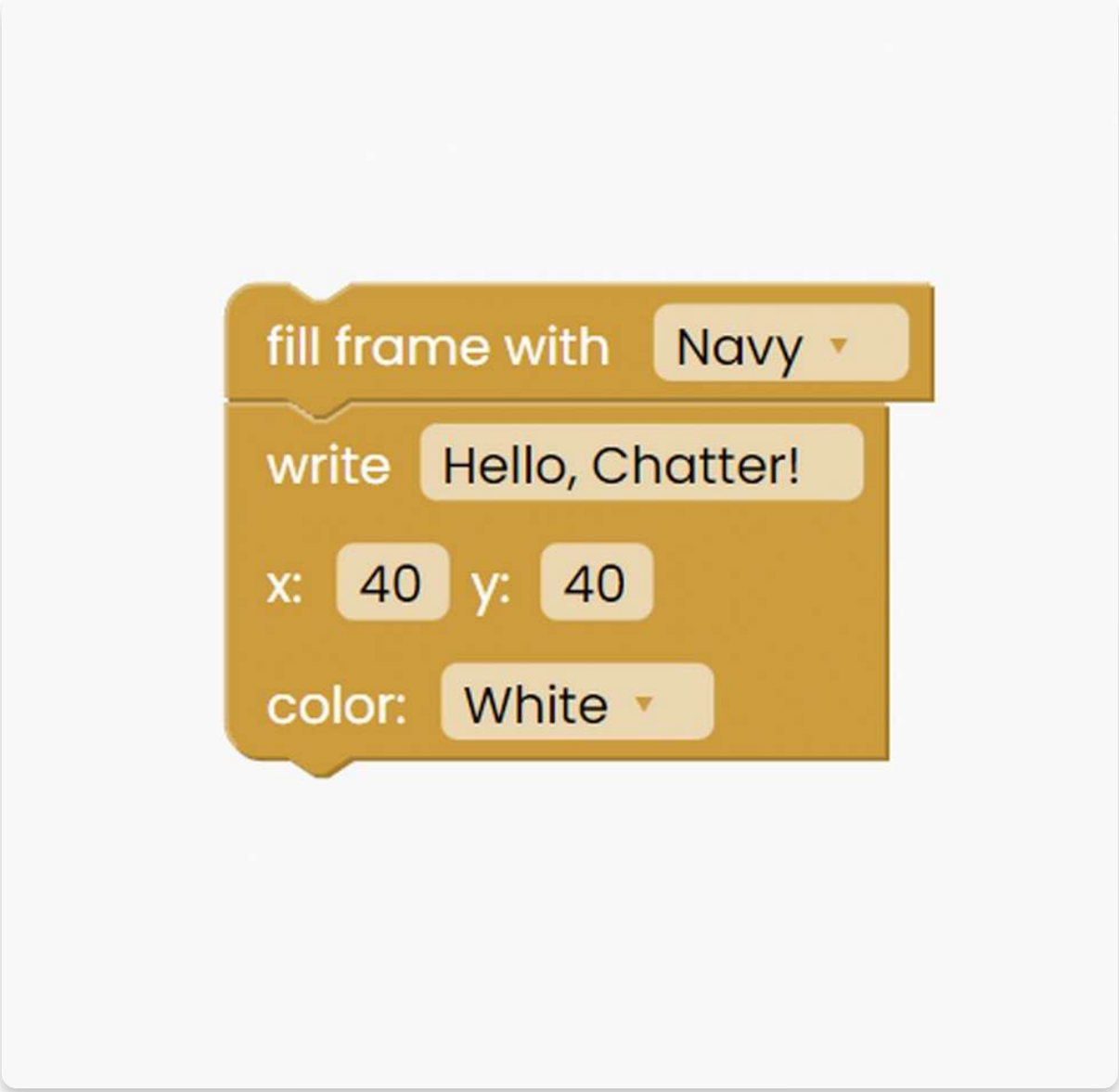
Now, let's drag and drop the block saying "**write string**". This block will be used whenever you want to write anything on Chatter's display.

If you want to write some other word instead of a string, you can simply delete the string, and write the word of your choice.

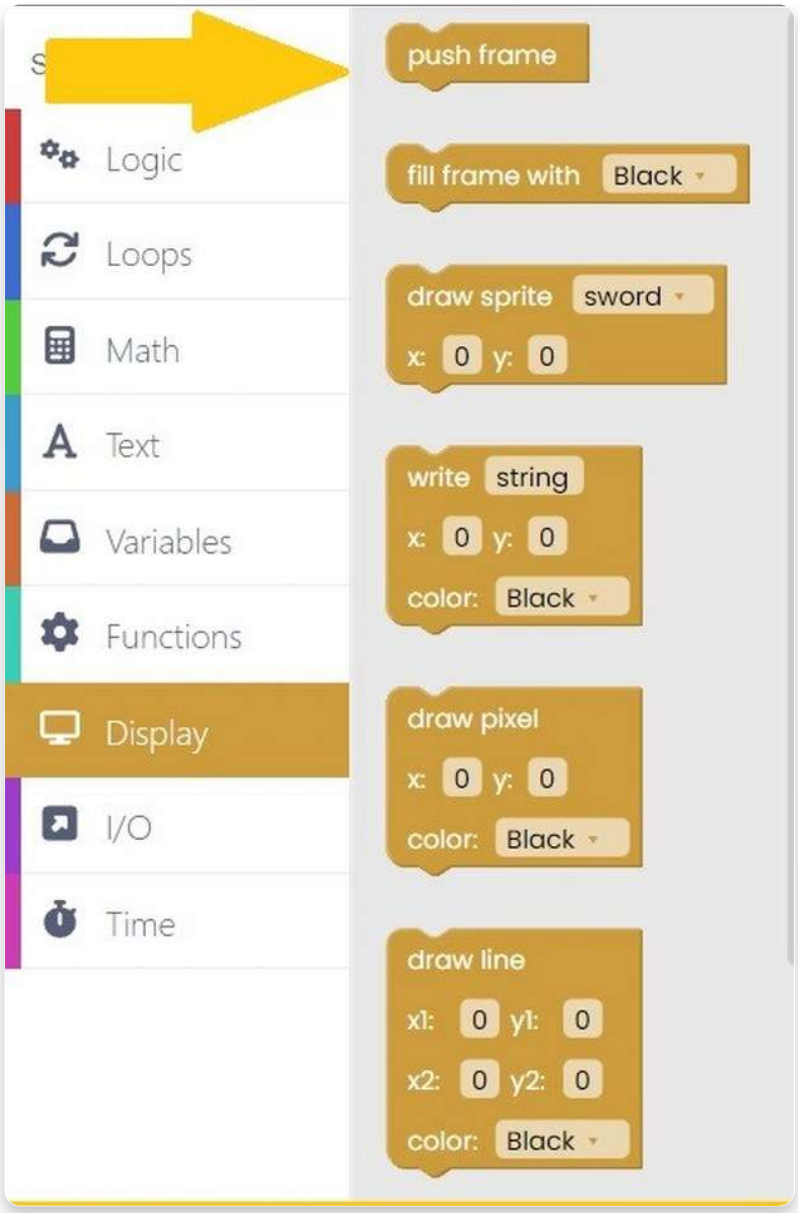
You can also choose the color of the text, and the **x** and **y** mark **coordinates** where the text will appear on display.



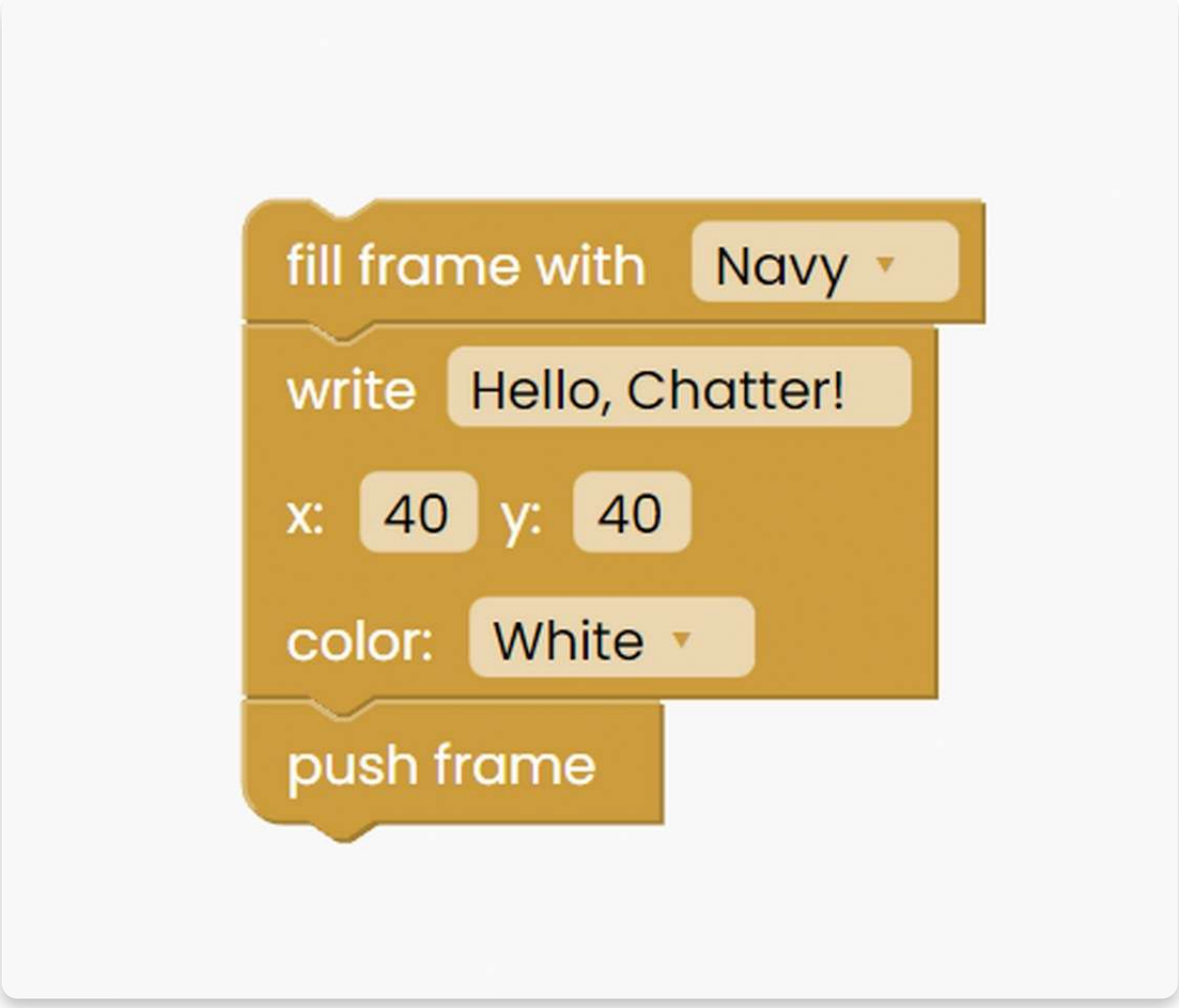
We decided to write "**Hello, Chatter!**" on the screen, and we put both coordinates to be 40.



The last thing you need to do in this sketch is to click on a "**push frame**" block.



We need to use this block to ensure this code will appear on display.



When you press the big red **Run button**, the text will appear on Chatter's screen.

Drawing different shapes on Chatter's display

In this sketch, we added two more examples for you.

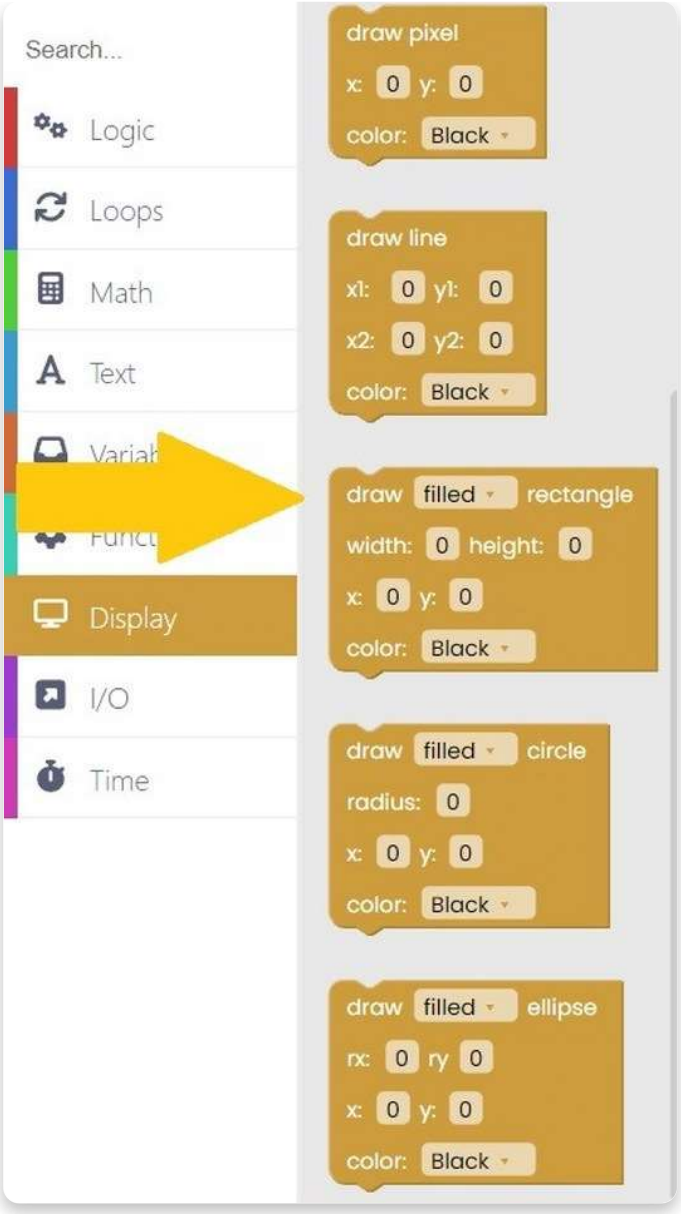
You will discover **how to draw ellipses and rectangles**.

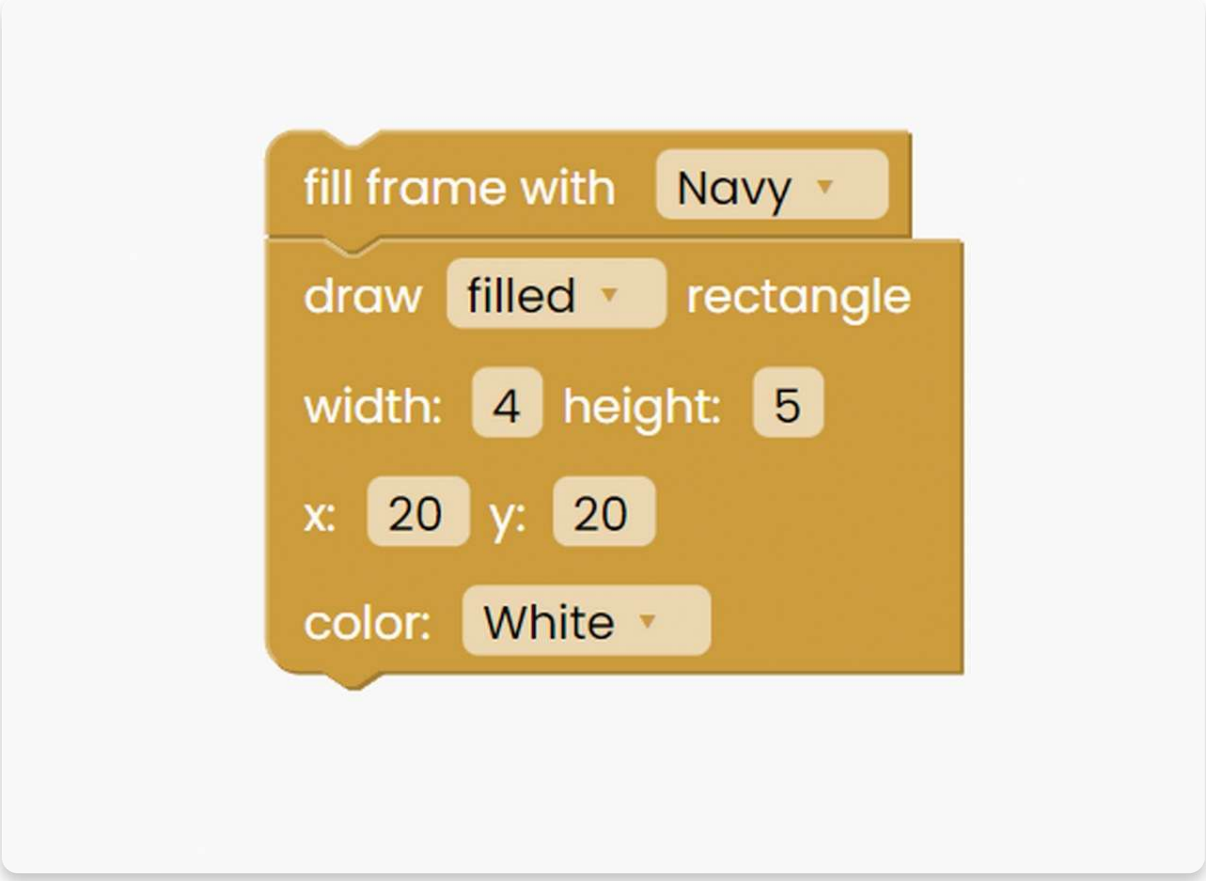
We'll stay in the display block section a little bit more.

As always, it is good to start by choosing the **background color**. We simply **duplicated the block** we made for the text example to save us some time.

You can **duplicate** blocks by clicking on the right button on your mouse and choosing the duplicate option from the menu.

Now, go back to the **display block section** and choose the "**draw filled rectangle**" block.


Search...
Logic
Loops
Math
Text
Display
I/O
Time
draw pixel
x: 0 y: 0
color: Black
draw line
x1: 0 y1: 0
x2: 0 y2: 0
color: Black
draw filled rectangle
width: 0 height: 0
x: 0 y: 0
color: Black
draw filled circle
radius: 0
x: 0 y: 0
color: Black
draw filled ellipse
rx: 0 ry 0
x: 0 y: 0
color: Black

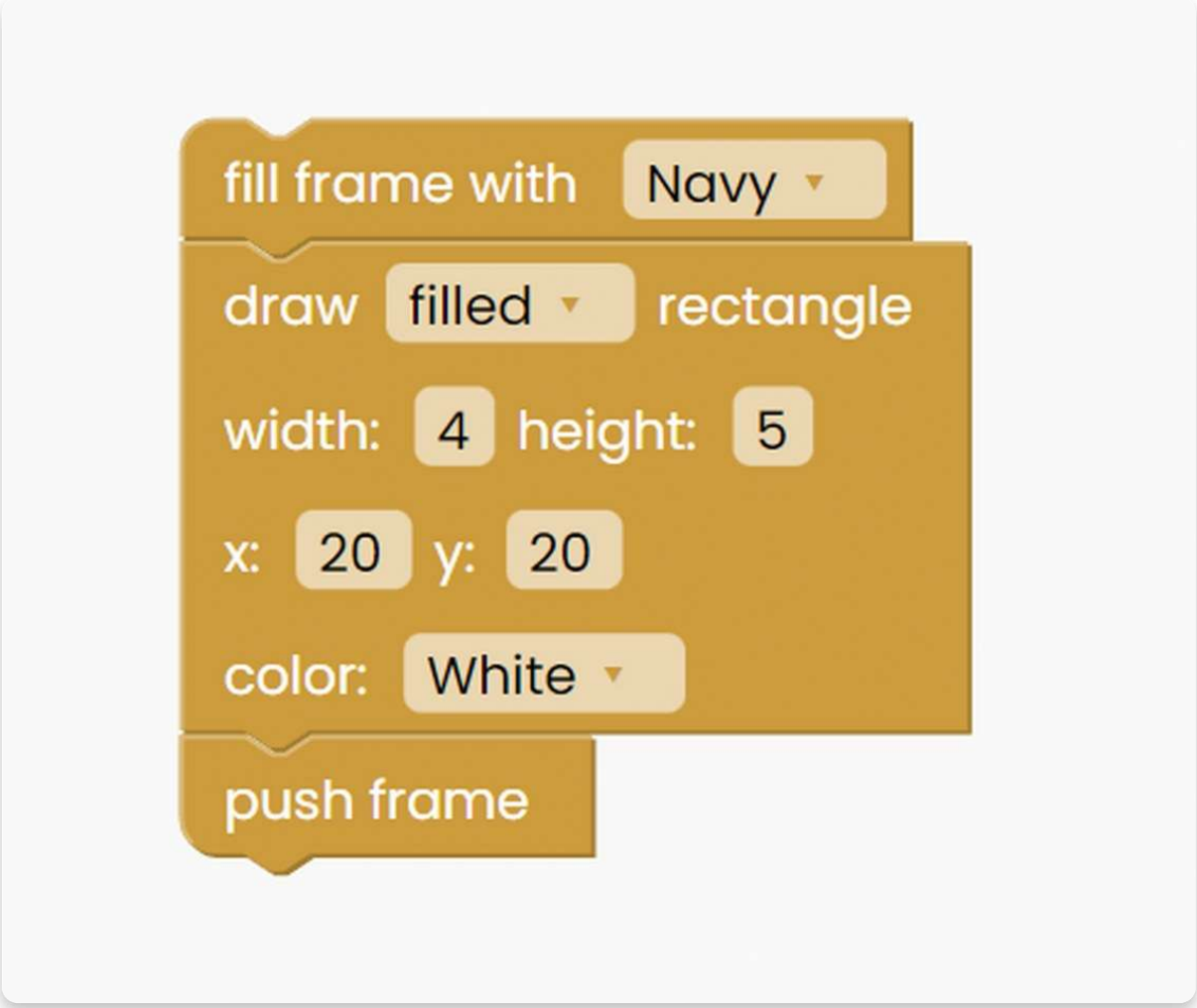


As you can see, you can specify whether your rectangle will be filled or outlined, which color it will be, and determine its size and coordinates.

We decided to create a white-filled rectangle with a navy background. It will be 4 (width) x 5 (height), with coordinates of x 20 and y 20.

The only thing left to do is add the "**push frame**" block to ensure our rectangle appears on the screen.

Here's what your code should look like:



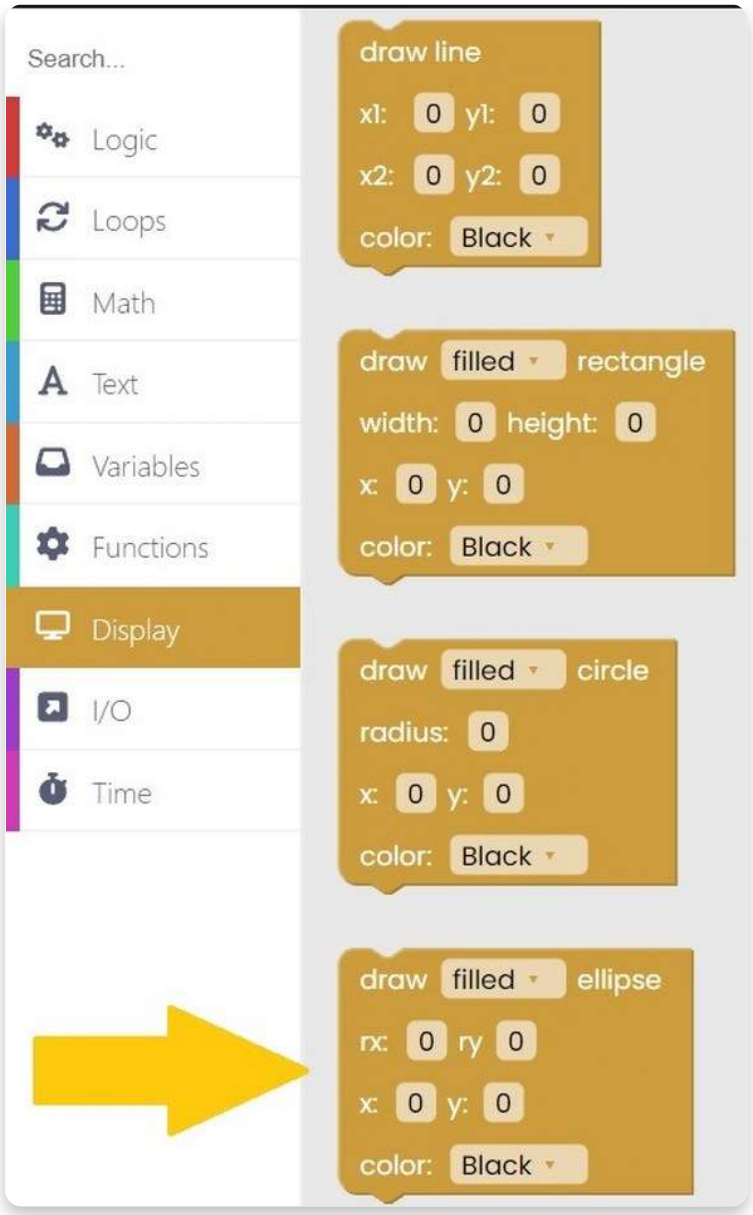
Click on the **Run button**, and check Chatter's display.

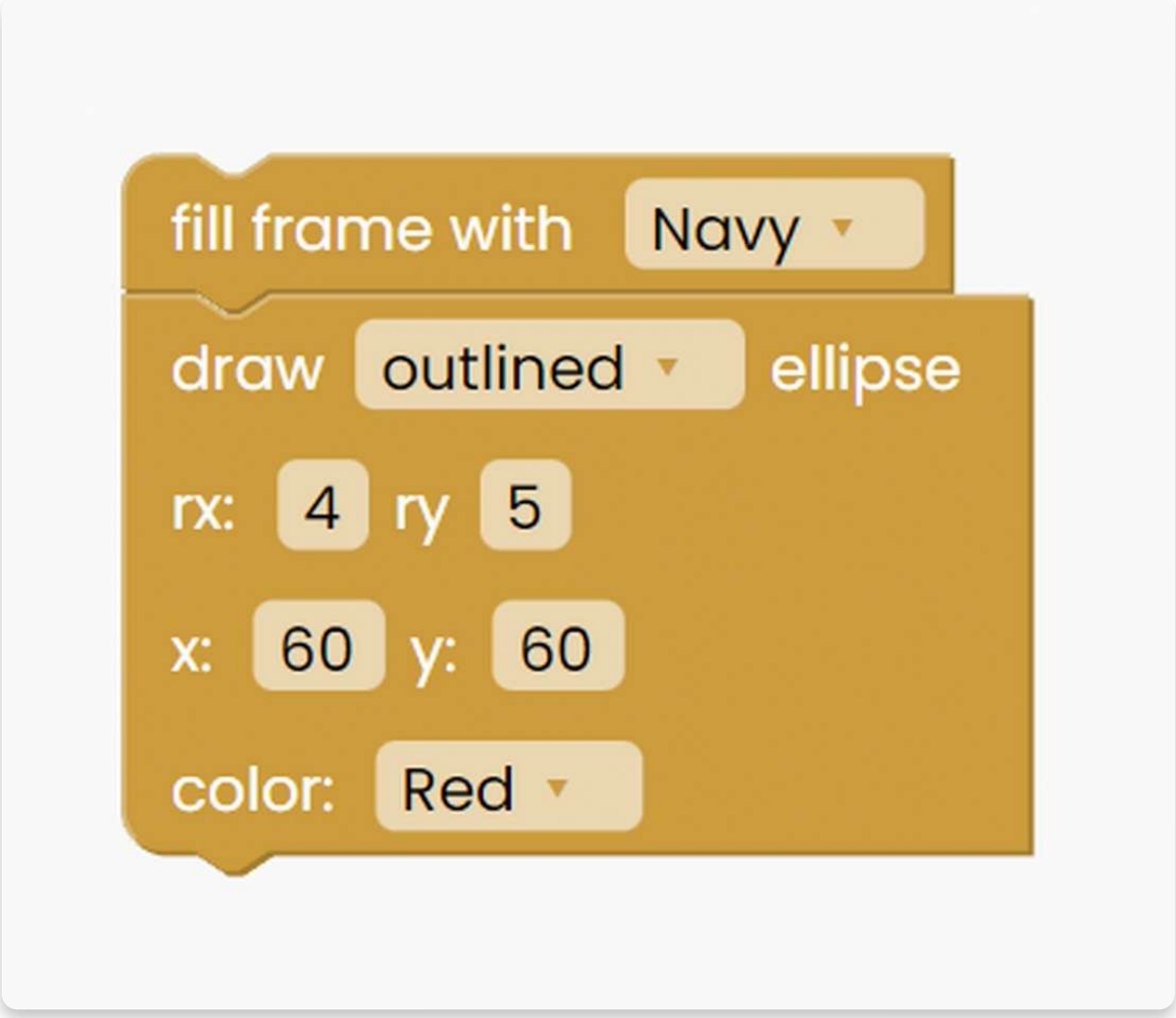
Finally, let's try drawing another shape - an ellipse.

If you drew the rectangle in the previous example successfully, this will be a piece of cake for you.

Return to the **display block section** and select the "**fill frame with**" block, or simply **duplicate** it as we did before.

Then, in the display block section, look for the "**draw filled ellipse**" block and drag it into the drawing area.

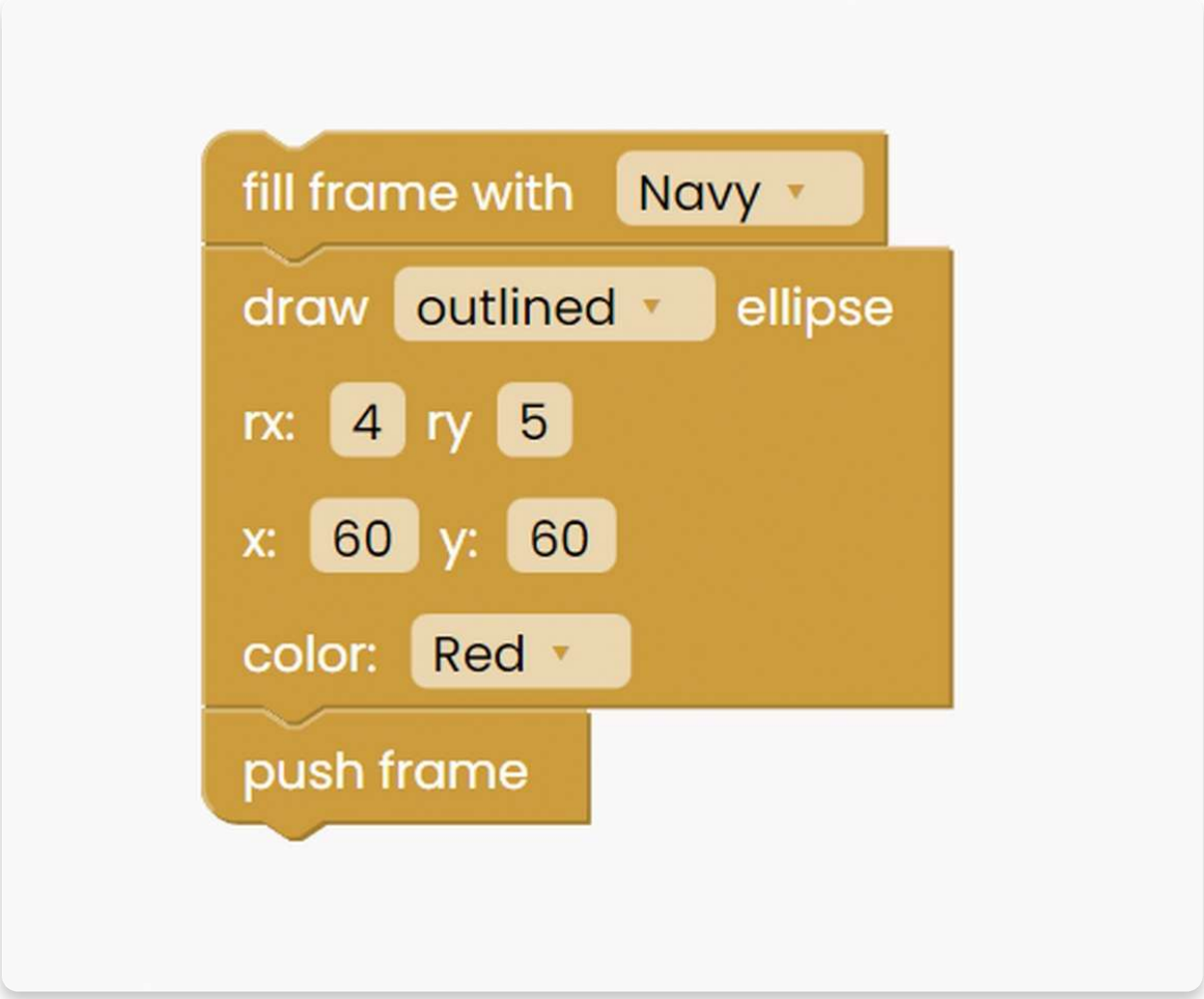

Search...
Logic
Loops
Math
Text
Variables
Functions
Display
I/O
Time
draw line
x1: 0 y1: 0
x2: 0 y2: 0
color: Black
draw filled rectangle
width: 0 height: 0
x: 0 y: 0
color: Black
draw filled circle
radius: 0
x: 0 y: 0
color: Black
draw filled ellipse
rx: 0 ry 0
x: 0 y: 0
color: Black



We chose navy as the background color once more.

We did, however, choose to **outline** the ellipse in red this time. The size and coordinates are shown in the photo above.

As you already know, one last thing left to do is to put the "push frame" button at the end of the code to make sure it will be shown on Chatter's display.



Now, click the Run button, and check out Chatter's display. Cool, right?

You can now change the shapes, colors, sizes, and text, but we're about to start a new chapter in which we'll learn how to code Chatter's buttons.

Click, click

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

The first thing we'll have to do is similar to the previous sketch.

Go to the display block section and choose the "**fill frame with**" block.

Now we'd like to write the instructions that are shown as soon as we press the Run button.

We'll need the "**write string**" block for that. You can find it here:

As shown in the above image, we'll leave the background black and write "**Push the buttons**" in yellow.

Finally, we must drag the "**push frame**" block to ensure that the **instructions appear on the display**.

Amazing!

You can click **Run to see if your code is working properly**, and then proceed with the sketch and change what happens when a particular event is triggered.

When you press Chatter's buttons, we trigger a particular event.

Luckily, we have a specific block defined for that, and it's under the **I/O** section. I/O stands for "**Input/output**".

Chatter's buttons are the so-called input devices because they send an electrical impulse to Chatter's computer when triggered. Chatter's display is an example of an output device because Chatter sends signals to it to display information.

The first block from the I/O section we'll use is "**when left pressed**".

Drag and drop it under the instructions part of the code.

We decided to start with the button **1**, so you can change the button you're working on to keep up with us.

So, what we want to do is to write a different word on Chatter's display by clicking on different buttons, and this is how to do so:

Since we want to write something on Chatter's display once we press the button 1, we have to **clear the display in particular color**.

Find the "**fill frame with**" block and drop it inside the purple **I/O block**:

As you can see, pressing the button 1 causes the background color change to red.

Let's write something on the screen now.

First, find the following block:

We'll start with the word "**Hello**" in black and with the coordinates **60x50**.

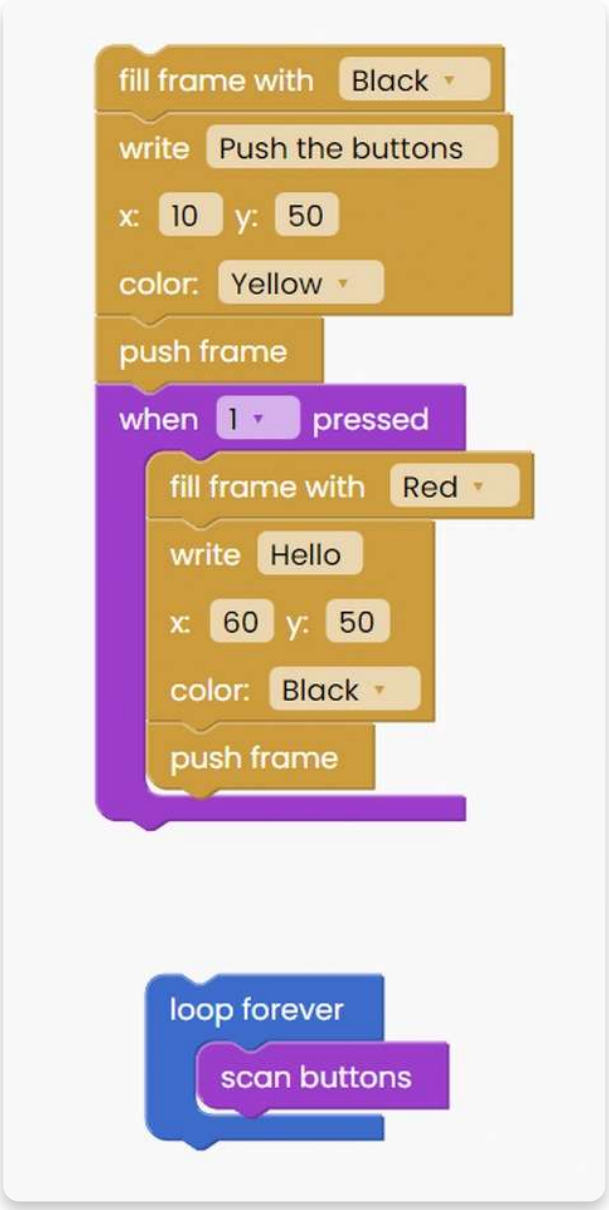
After that, add the "**push frame**" block under the write block to ensure that your text and background color change when you press that specific button.

Before we can test whether pressing this specific button causes the desired event, we must first find the "**loop forever**" block to ensure that your buttons are being scanned at all times. **Please remember to use this every time you code something with the buttons, or it will not work.**

The "**loop forever**" button can be found here:

Drag and drop it on the drawing area and look for the **I/O** "**scan buttons**" block:

Drag this I/O block inside the "**loop forever**" block. Just like this:



You can now test everything by pressing the **Run button** and then clicking on button 1.

Let's now repeat these steps for three more buttons.

You can either repeat the previous steps or click the purple block to duplicate it. This will duplicate the purple block and everything within it.

This time, we want to trigger an event once we press the **button 2**.

We'll now change the **background color** to **blue** and write "**Hi**" in **black letters** on the same **60x50 coordinates**.

Let's duplicate the **I/O block** two more times.

Your code should look like this:

So we chose the left and right buttons for the final two buttons.

You can see the colors and the text we wrote in the image above.

Of course, you can now write any other word you want, as well as change the colors and coordinates.

To see the **colors and text changing on the display**, press the **Run button** and then click on the **buttons 1, 2, left, and right**.

Sound on

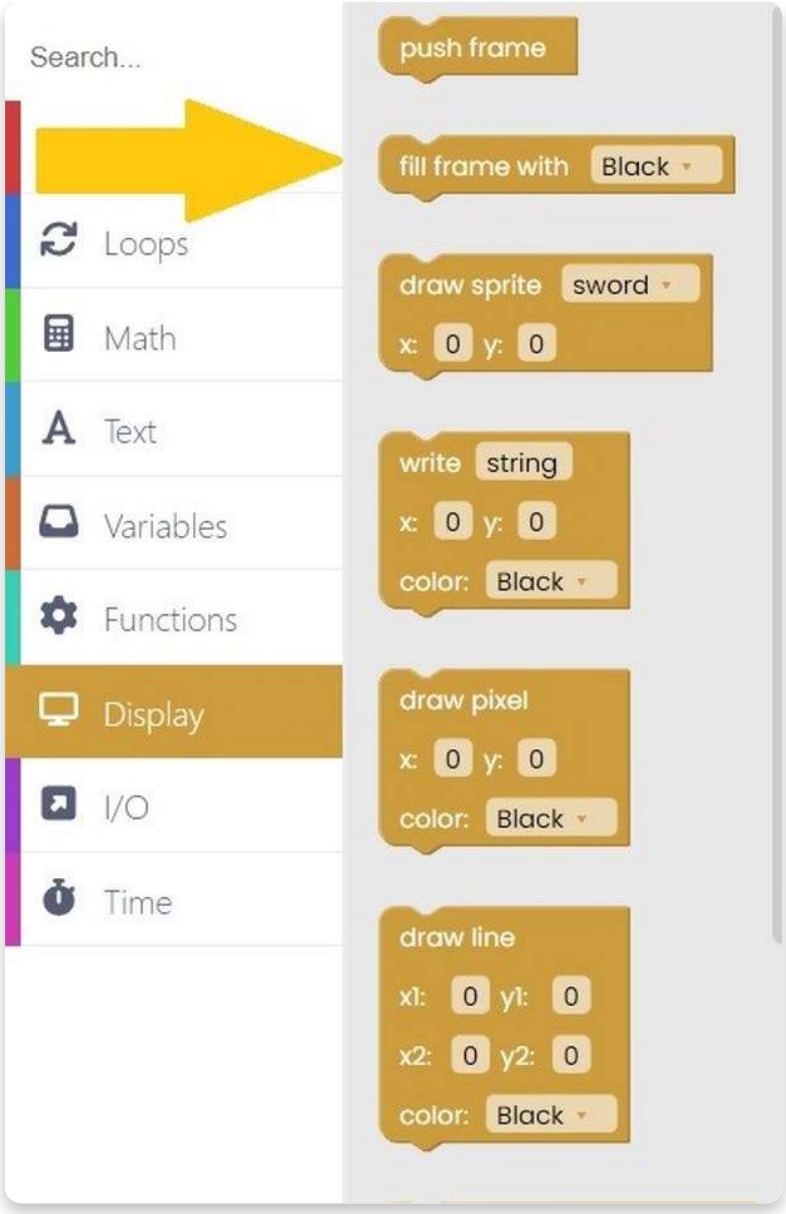
Let's learn what to do with the **Piezo buzzer** that's soldered onto your Chatter.

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.**

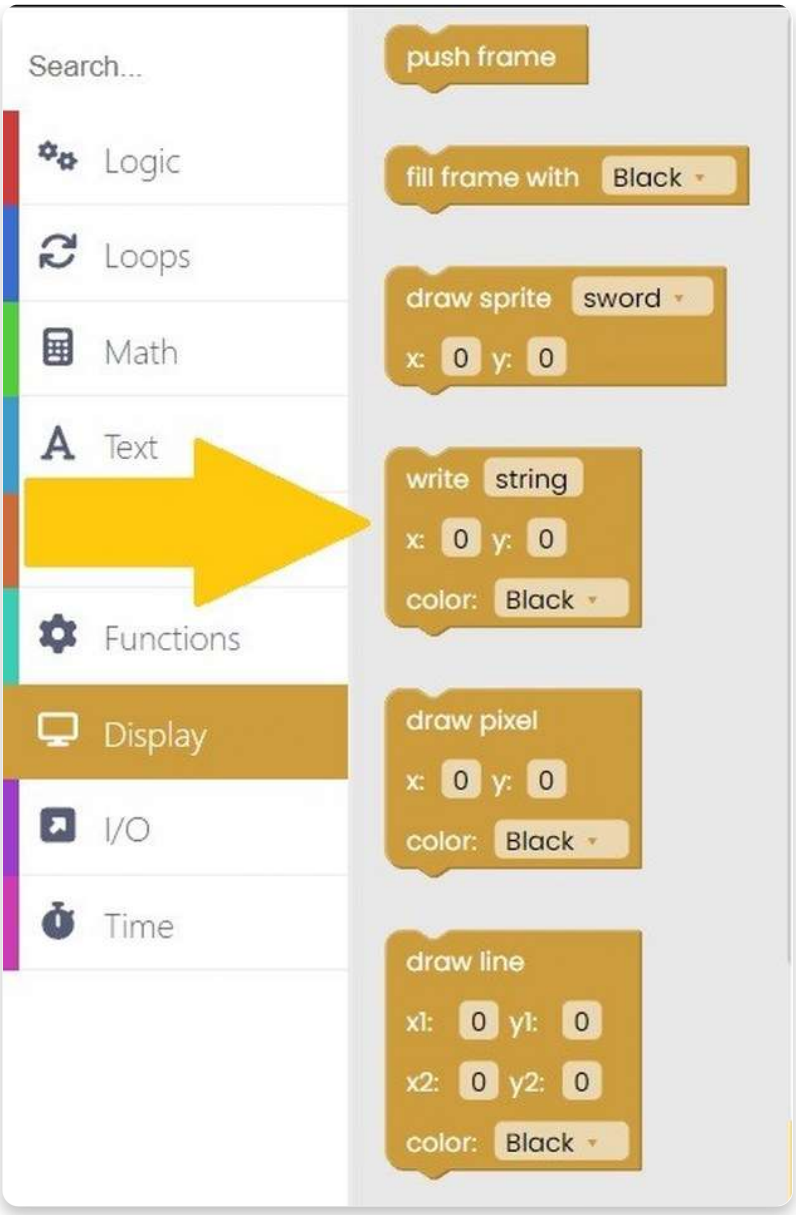
Let's start!

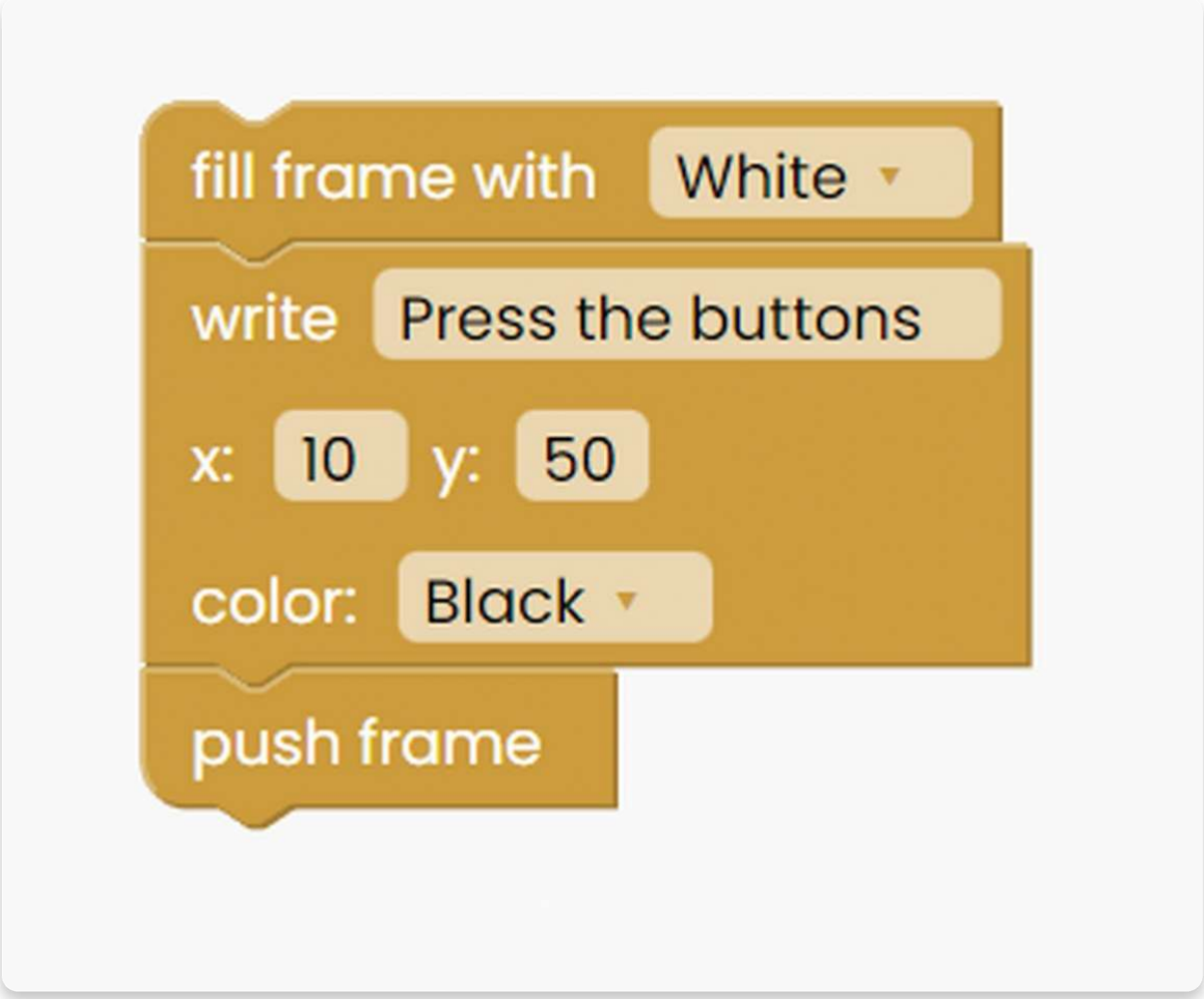
Go to the display block section and choose the "**fill frame with**" block.



Now we'd like to write the instructions that are shown as soon as we press the Run button.

We'll need the "**write string**" block for that. You can find it here:


Search...
Logic
Loops
Math
Text
Functions
Display
I/O
Time
push frame
fill frame with Black
draw sprite sword
x: 0 y: 0
write string
x: 0 y: 0
color: Black
draw pixel
x: 0 y: 0
color: Black
draw line
x1: 0 y1: 0
x2: 0 y2: 0
color: Black



We'll make the background white and write "**Press the buttons**" in black, as shown in the image above.

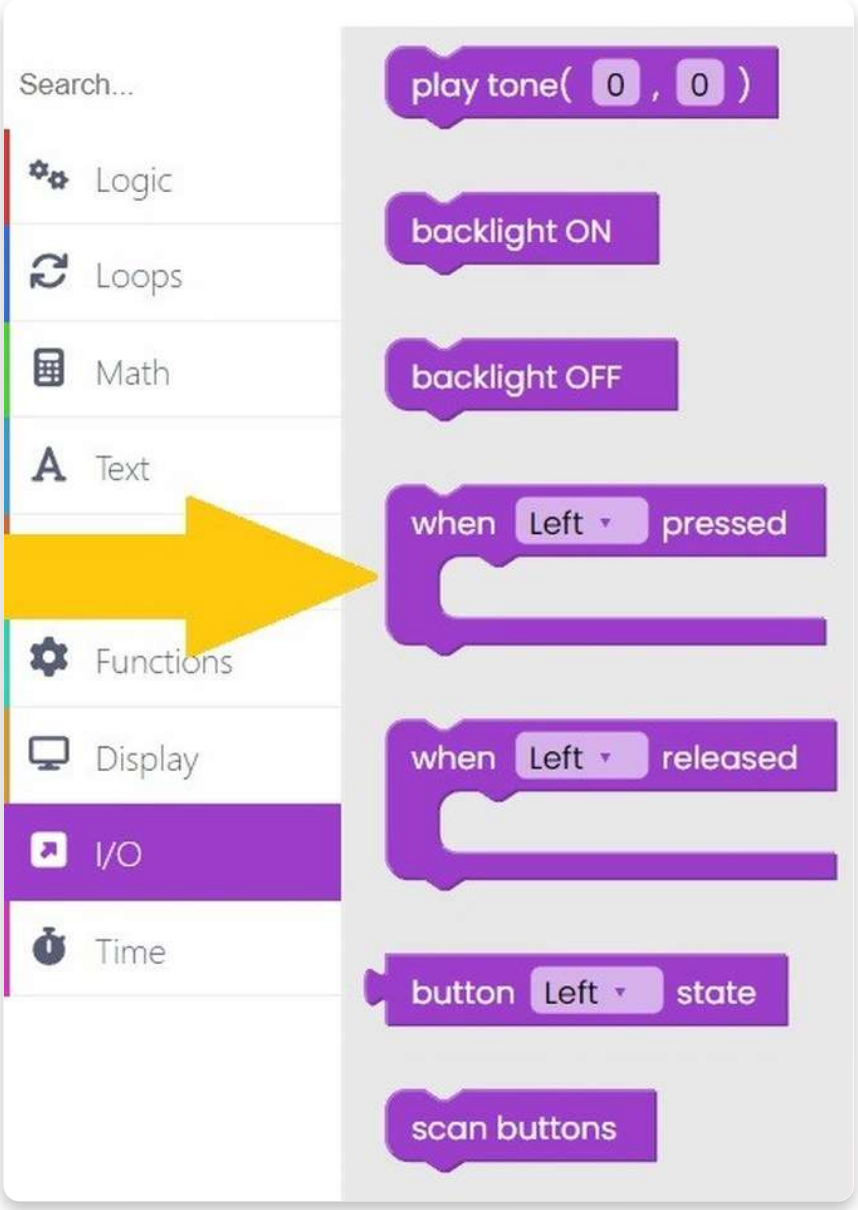
We don't want to forget about the "**push frame**" block.

Amazing! **You can test your code by clicking Run,** and then proceed with the sketch and change what happens when a particular event is triggered.

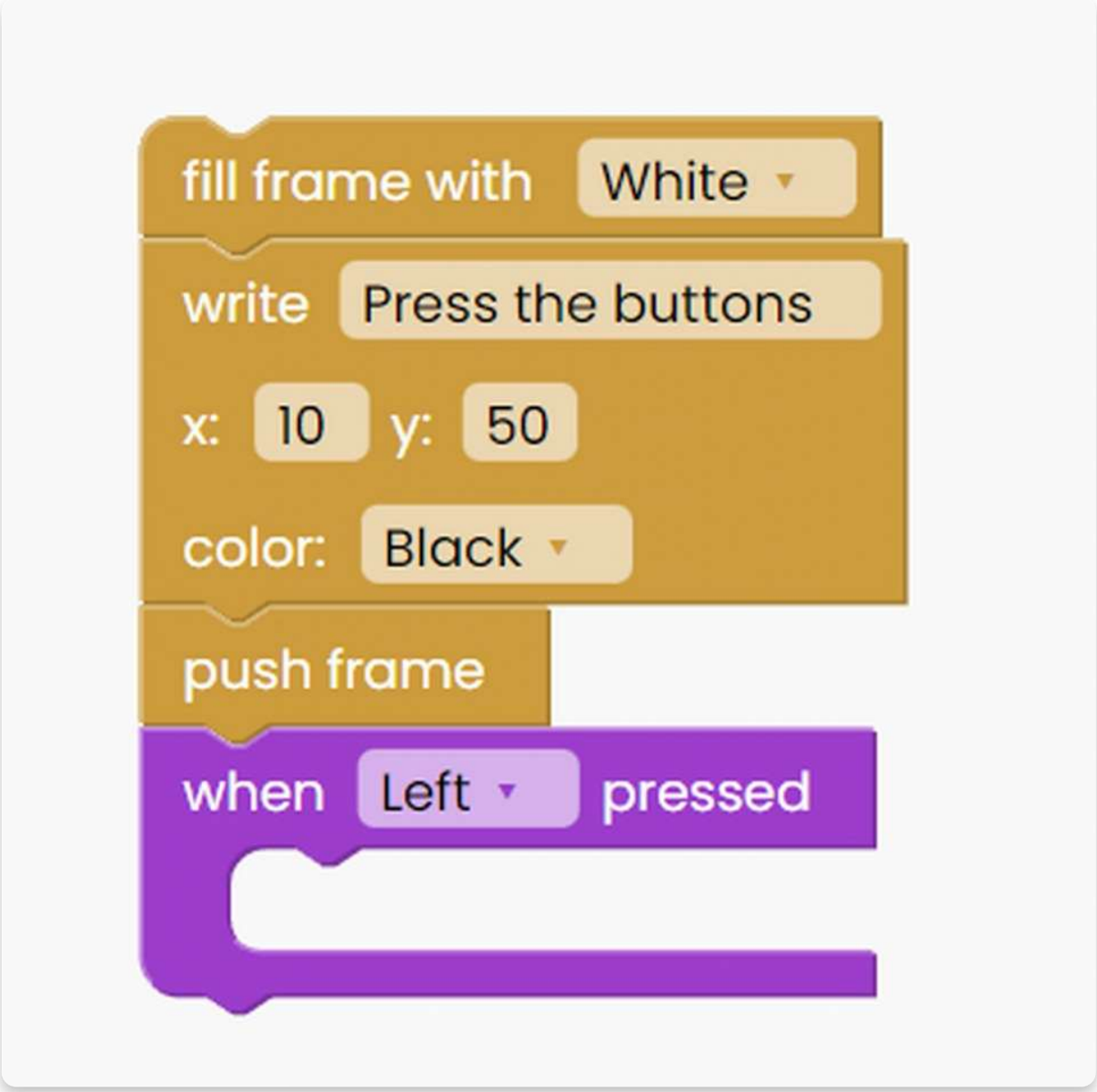
When you press Chatter's buttons, we trigger a particular event.

As we learned in the previous chapter, a specific block defined for that can be found under the **I/O** section.

The first block from the I/O section we'll use is "**when left pressed**".



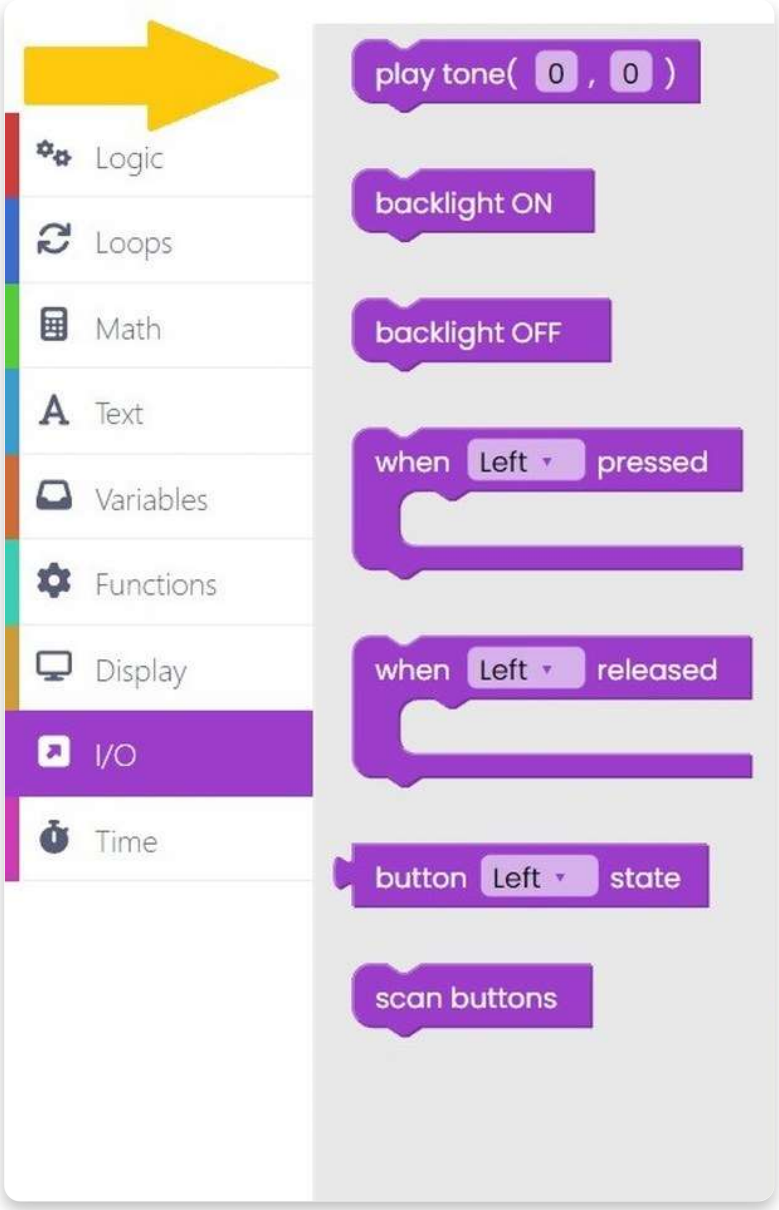
Drag and drop it under the **instructions** part of the code.



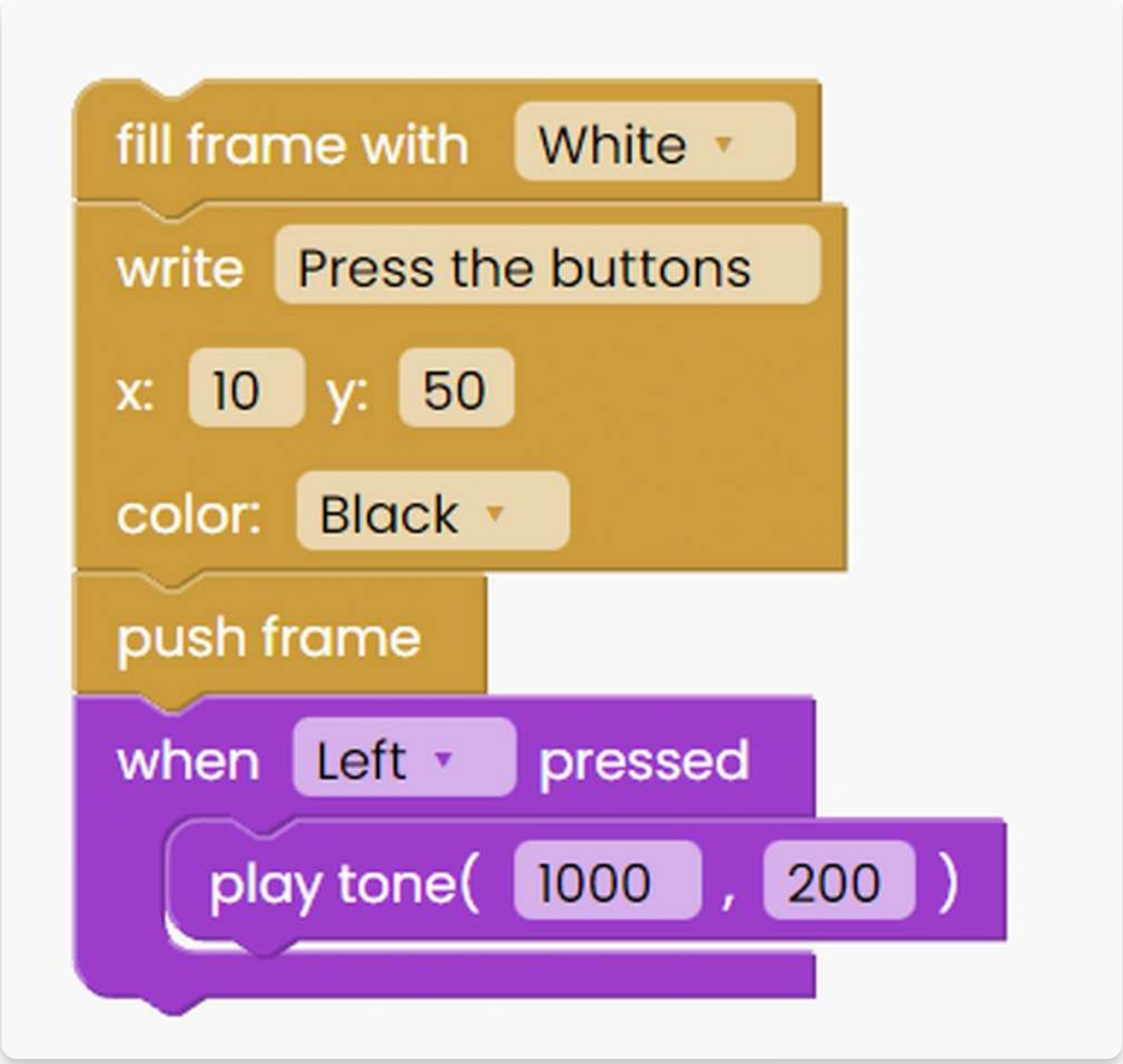
This time, we want a **buzzer** to make a **specific sound** and **draw one of the shapes** from the first chapter.

To do so, we'll need to begin with another **I/O block**, "**play tone (0,0)**".

This block will be used to specify the **frequency** and **duration** of our sound.



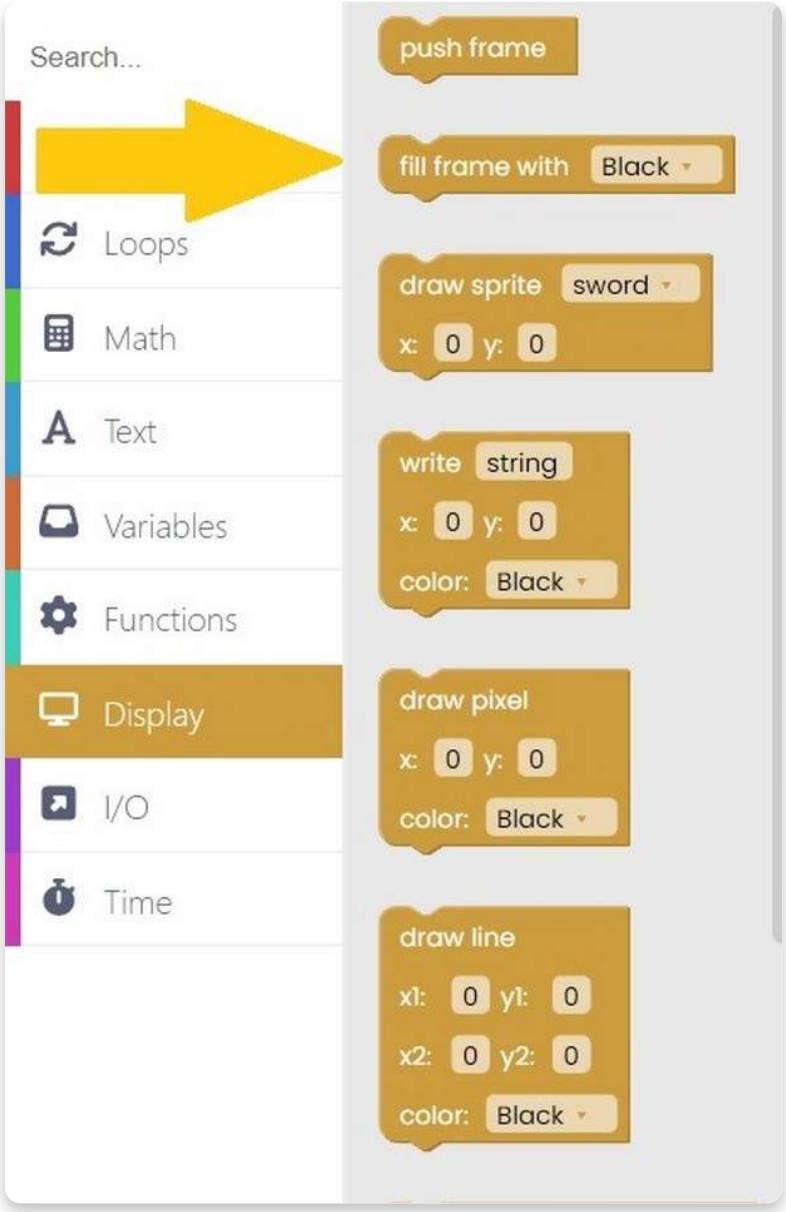
Drag it into the **I/O block,** change the **frequency** to **1000 Hz**, and play it for **200 milliseconds**.

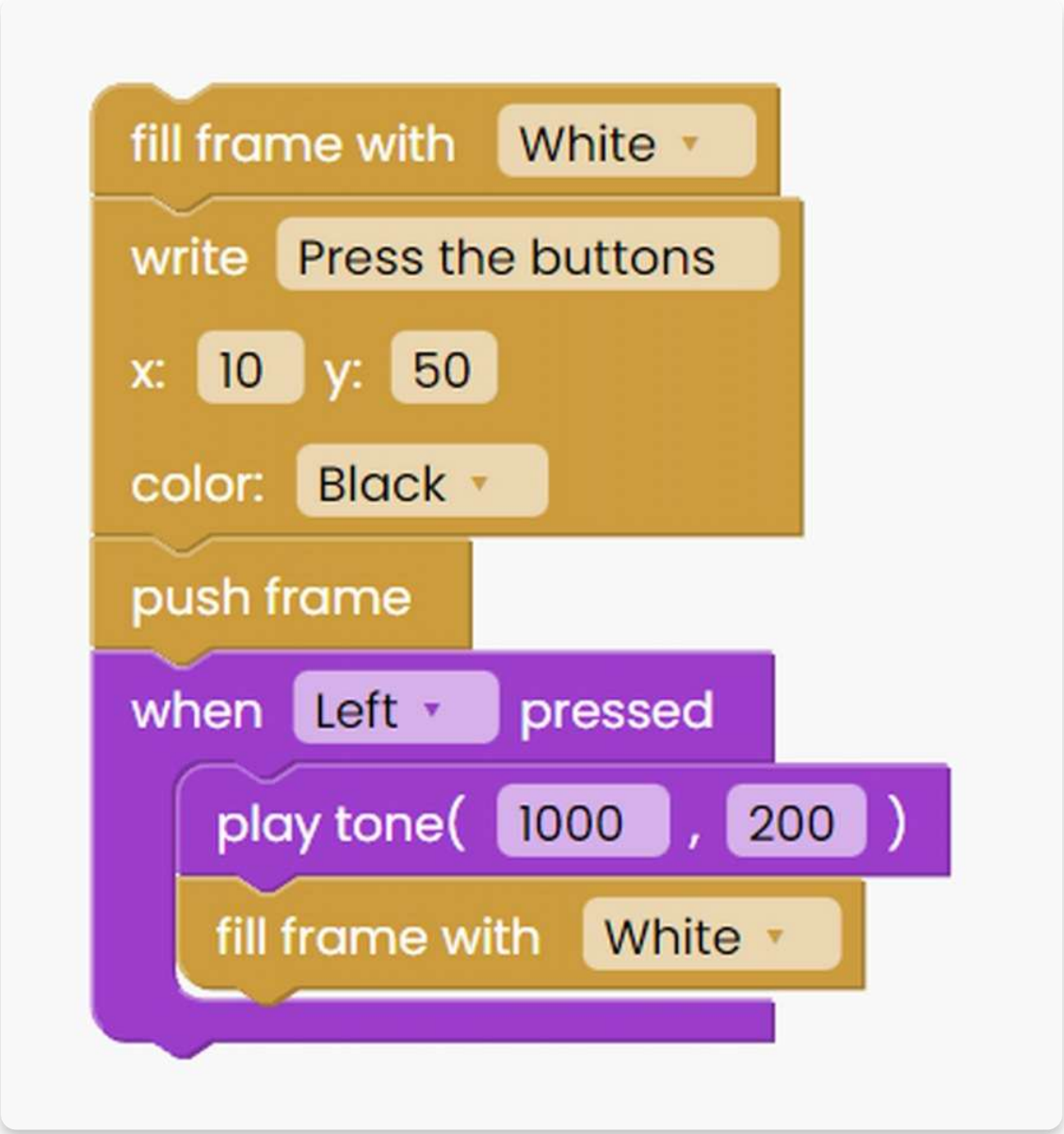


Amazing!

Let's try something similar to what we did in the previous example.

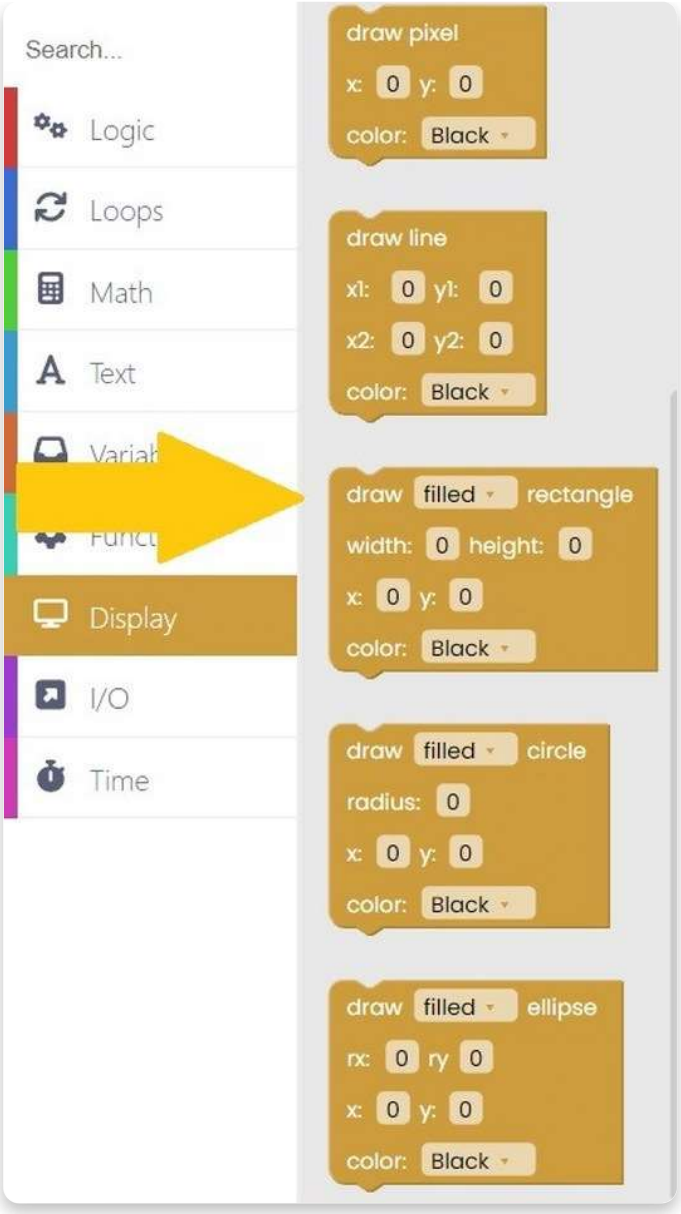
Because we'll be drawing something on the screen, we'll need to fill the screen with the particular color, which you can do by locating the "**fill frame**" block.

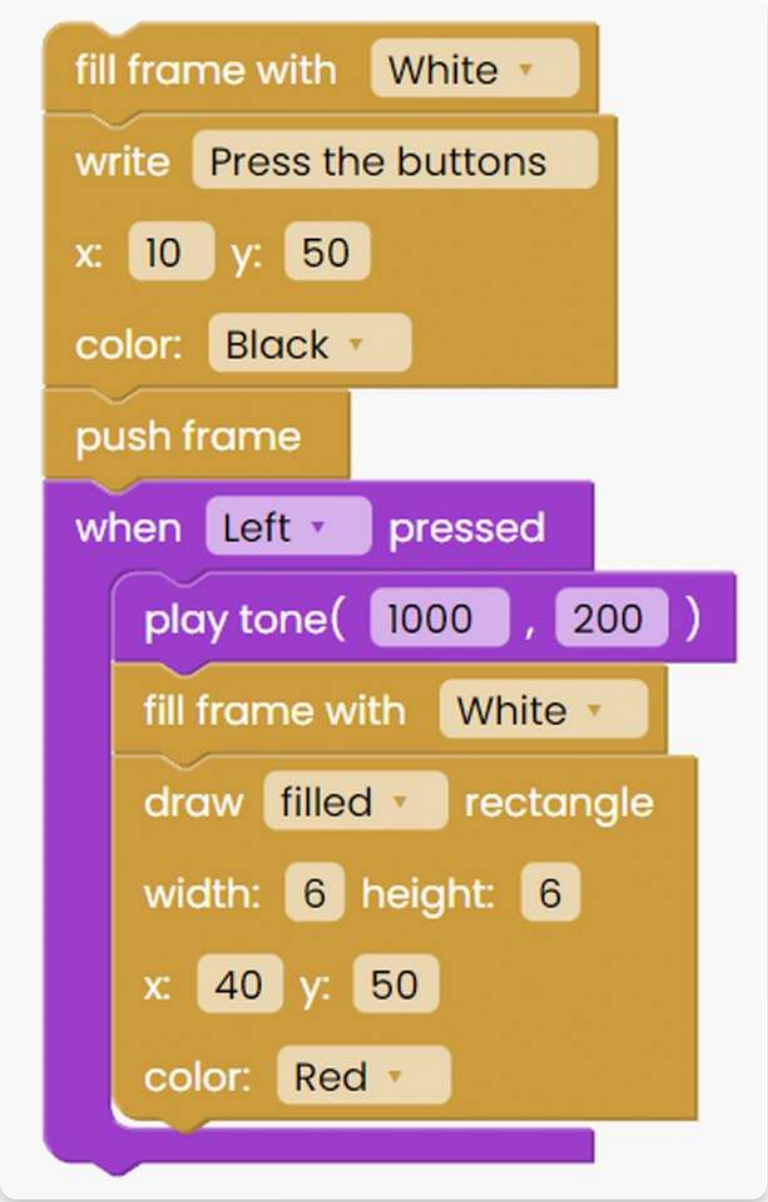

Search...
Loops
Math
Text
Variables
Functions
Display
I/O
Time
push frame
fill frame with Black
draw sprite sword
x: 0 y: 0
write string
x: 0 y: 0
color: Black
draw pixel
x: 0 y: 0
color: Black
draw line
x1: 0 y1: 0
x2: 0 y2: 0
color: Black



We must now decide what we want to draw, whether it will be filled or outlined, and where we want it to be drawn.

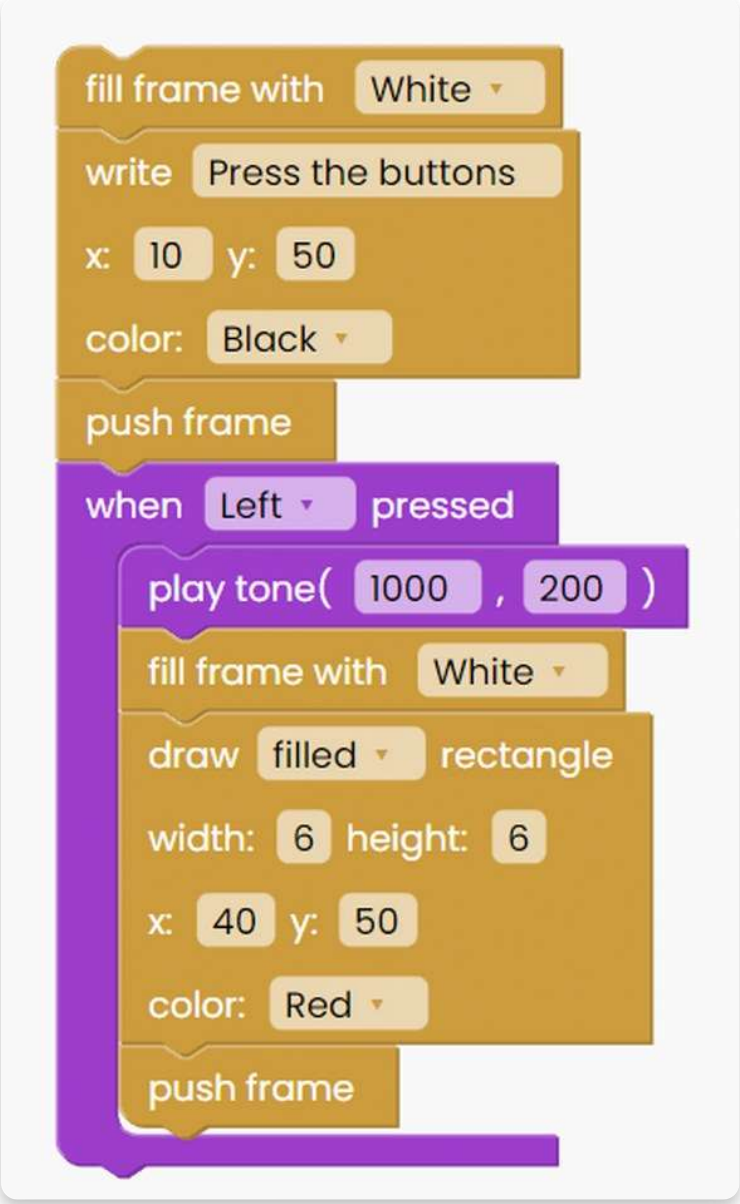
We can do that using this block:


Search...
Logic
Loops
Math
Text
Display
I/O
Time
draw pixel
x: 0 y: 0
color: Black
draw line
x1: 0 y1: 0
x2: 0 y2: 0
color: Black
draw filled rectangle
width: 0 height: 0
x: 0 y: 0
color: Black
draw filled circle
radius: 0
x: 0 y: 0
color: Black
draw filled ellipse
rx: 0 ry 0
x: 0 y: 0
color: Black



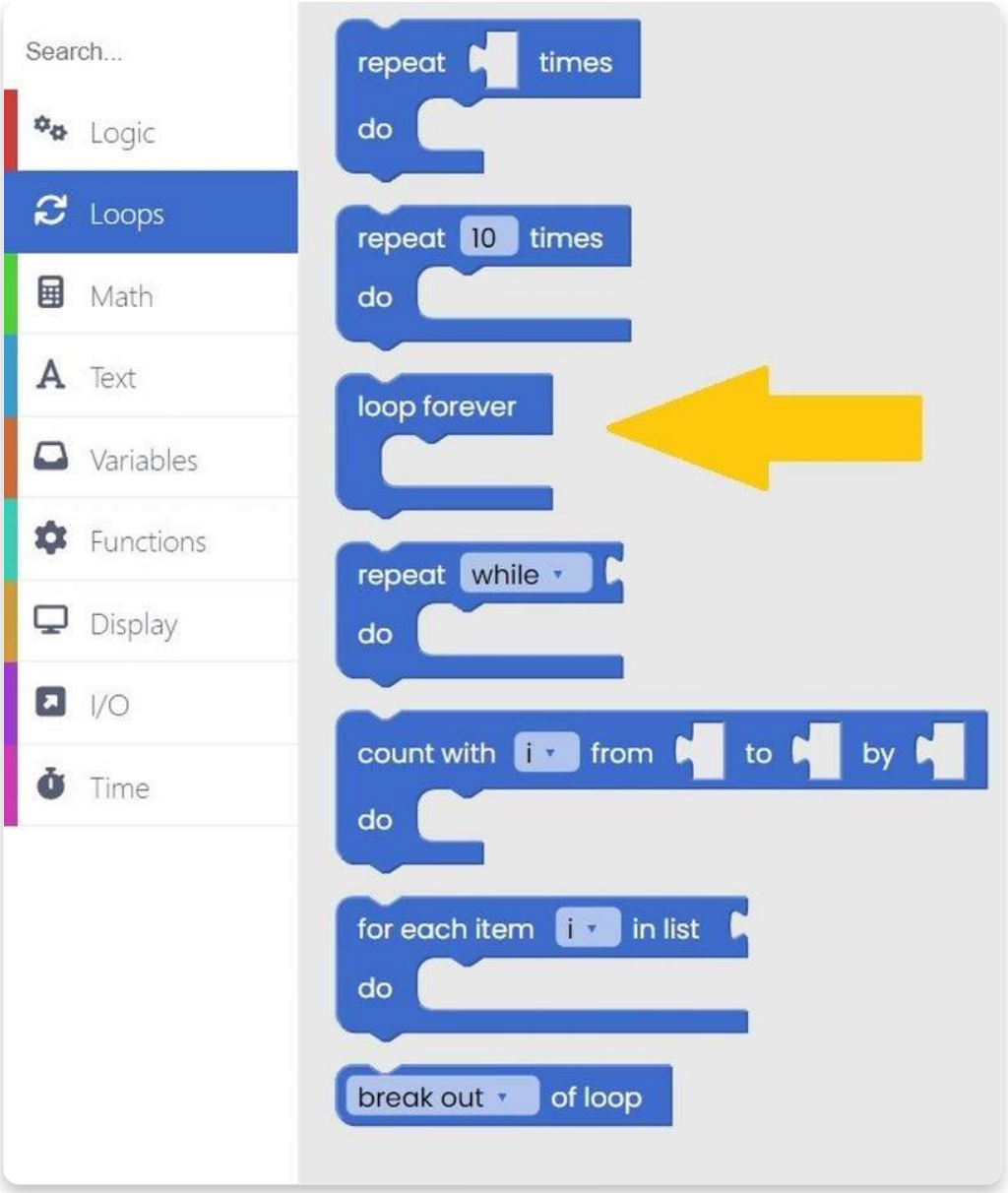
We chose to draw a **6x6 filled red rectangle** at the **coordinates 40x50**.

Finally, at the end of this block, insert the "**push frame**" block.

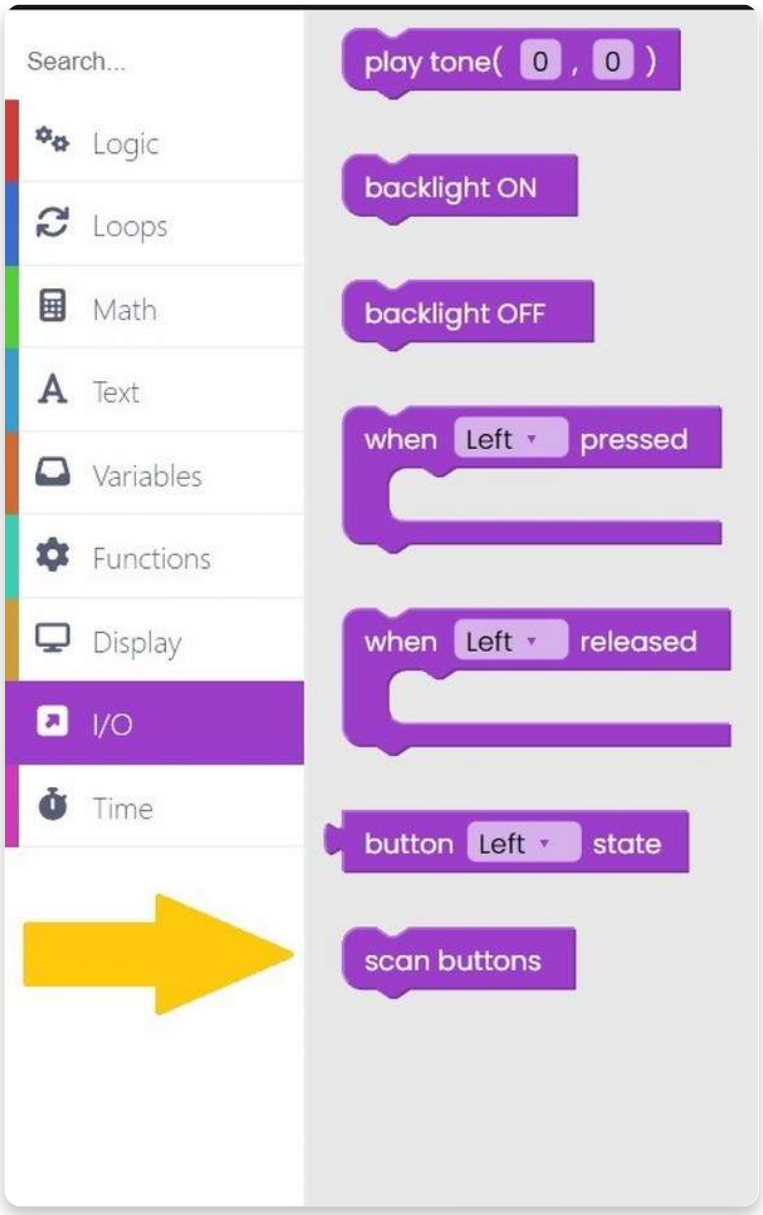


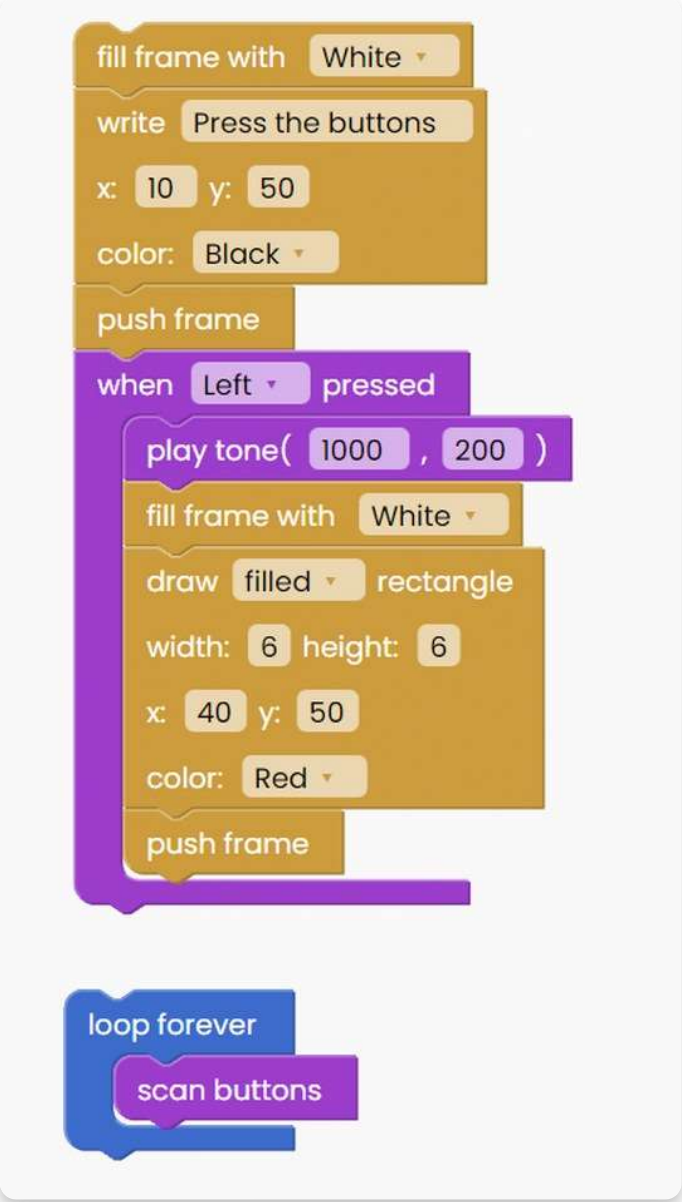
Before we can test whether pressing this specific button causes the desired event, we must first find the "**loop forever**" block to ensure that your buttons are being scanned at all times. **Please remember to use this every time you code something with the buttons, or it will not work.**

The "**loop forever**" button can be found here:



Drag and drop it on the drawing area and look for the **I/O** "**scan buttons**" block:


Search...
Logic
Loops
Math
Text
Variables
Functions
Display
I/O
Time
play tone(0 , 0)
backlight ON
backlight OFF
when Left pressed
when Left released
button Left state
scan buttons

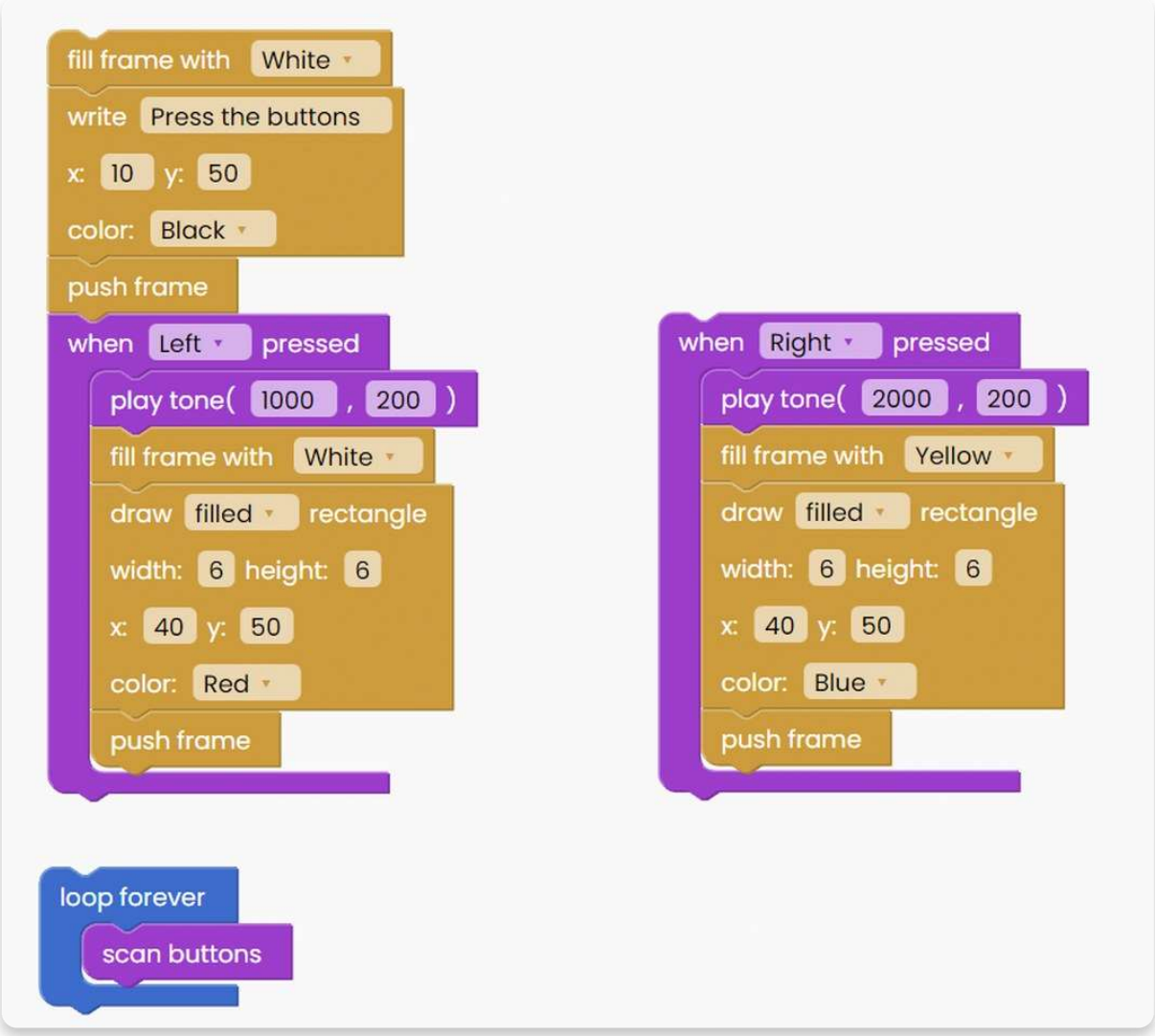


You can now test everything by pressing the **Run button** and then clicking on left button.

When you press that button, your screen should look like this, and you should hear a sound from your Chatter:

Now, just like in the previous sketch, we'll duplicate this for the three more buttons and change them slightly.

To begin, double-click the **I/O block** that says "**when left pressed**" to **duplicate** everything inside it.



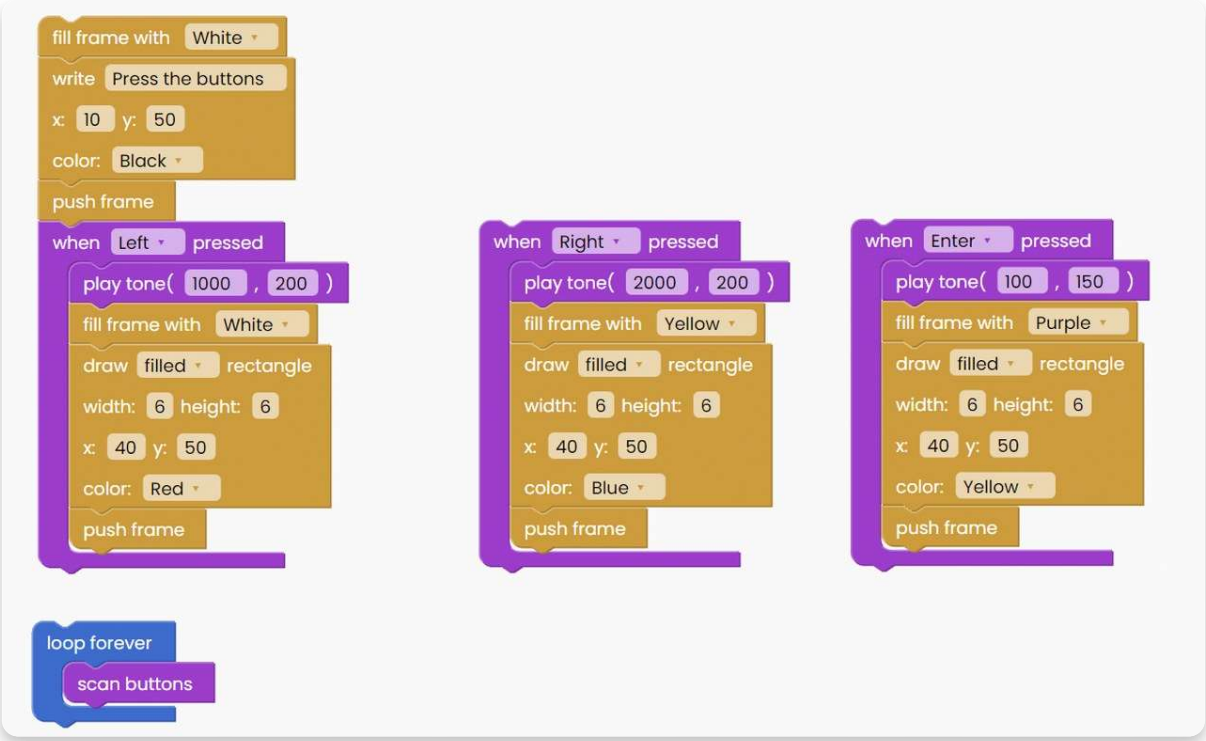
This new **I/O block** will alter what happens when we **press the right button**.

Now, we'll **play the sound at 2000 Hz for 200 milliseconds**.

At the same time, we'll draw a **6x6 blue-filled rectangle** with **coordinates 40x50** on a **yellow background**.

To see what happens, click the Run button and then the left button on your Chatter:

Let's do it again with another button.



This time, we'll trigger the event pressing the **enter button**.

The **sound will play at 100 Hz for 150 milliseconds**, and we will draw a **6x6 yellow-filled rectangle** on a **purple background** with **coordinates 40x50**.

To see what happens, click the **Run button** and then **enter**:

Finally, we'll **repeat this process for the back button**.

The **sound will play for 50 milliseconds at 3000 Hz**, and we will draw a **white filled rectangle 6x6** in the **coordinates 40x50** on the **marron background**.

Now that you know how to make sounds and draw different things, you can experiment with the buzzer and the display.

Using sprites

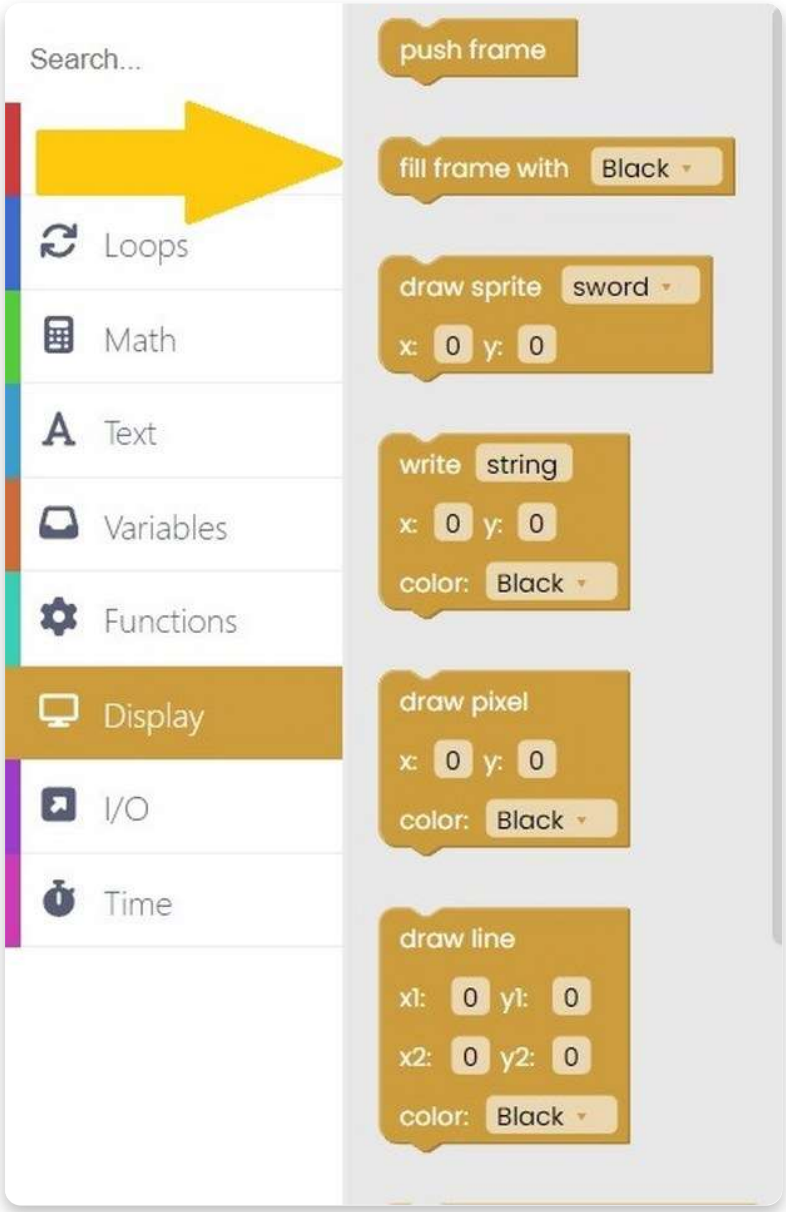
We've already covered how to create a new sprite and modify existing ones.

We'll now learn how to make various sprites appear on Chatter's display.

Let's start!

All of the sprites can be found in the display block section and are very simple to use.

To begin, we'd need to fill the frame with a specific color, which we know we can do by going to the display block section and selecting this specific block:



After filling the frame with a specific color, we must find the block we use to draw sprites.

In this block, we can select which of the many available sprites to draw and its coordinates.

We decided to draw a **sword** on a **white background at 50x50 coordinates**.

Finally, **don't forget** about the "**push frame**" button.

You can now press the Run button, and a sword should appear on Chatter's display.

Try drawing your sprite and putting it on Chatter's display now.

Restore Chatter's base firmware

Restore Chatter's firmware

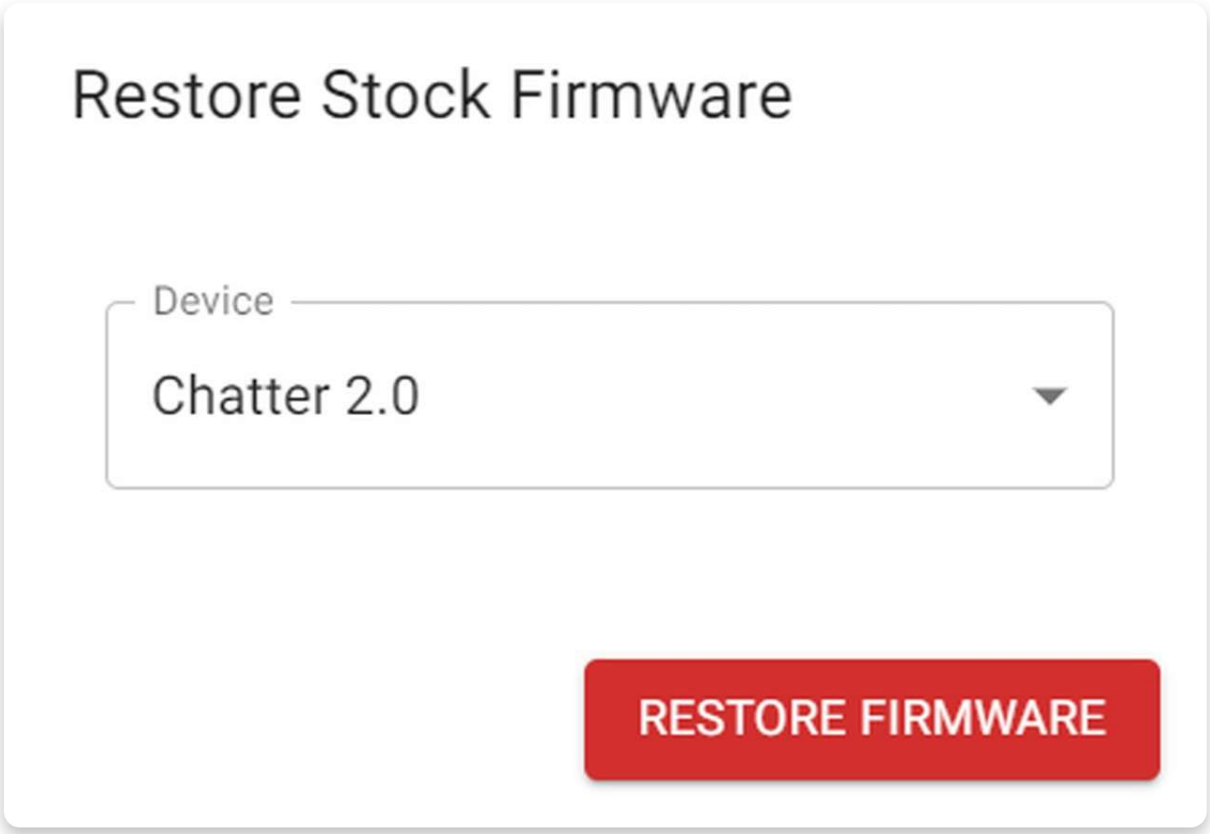
If you want to **return Chatter to its original state**, you can use the **restore firmware** option.

In MicroPython, you don't have to do this after running every new code.

To restore Chatter's firmware, follow these steps:

1. Navigate to the main CircuitBlocks page and look for the Restore firmware button in the upper right corner of the screen.

2. Click on it and choose the red Restore firmware button.



3. The process will begin and last for a few seconds.

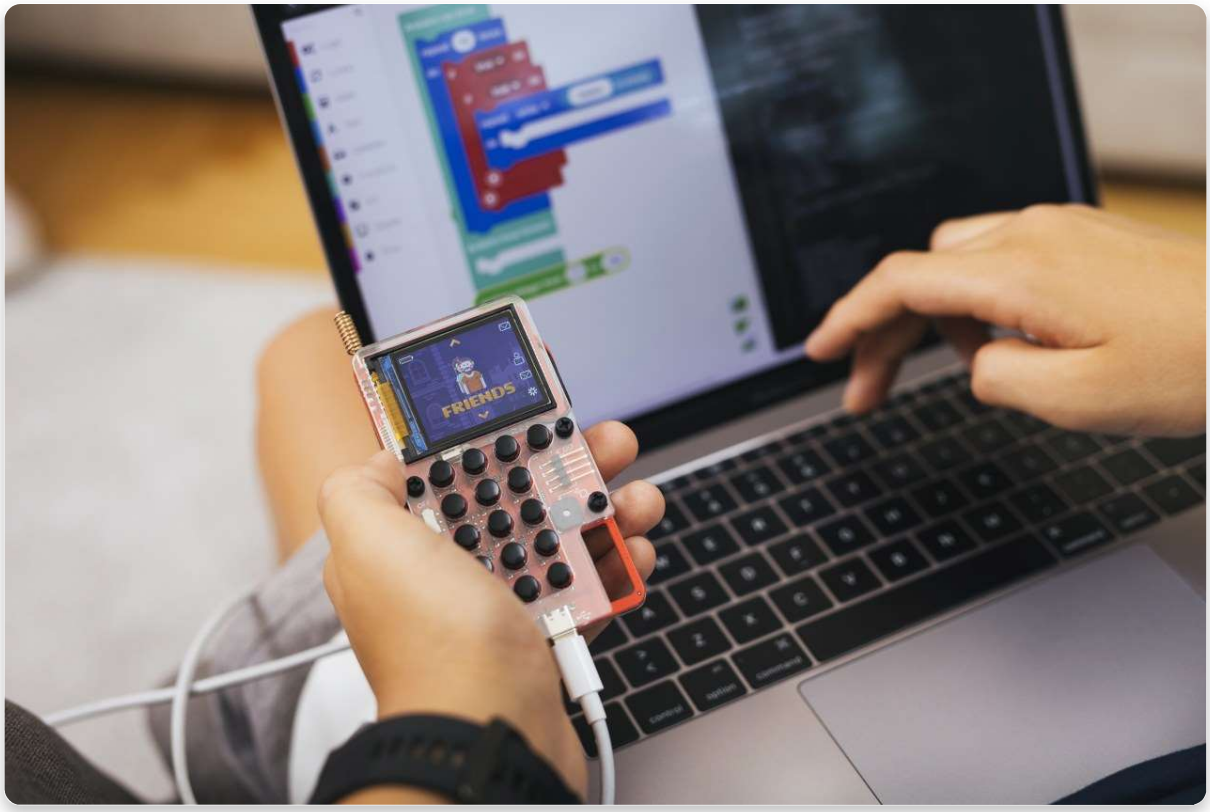
When it is finished, your device will restart and you will have to repeat the hardware test.

That's all!

Your Chatter has been restored to its original state, and you can use it once more.

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 the 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!