Scratch 3 Walkthrough and Demo

Created by Isaac Wellish

https://learn.adafruit.com/guide-to-scratch-3

Last updated on 2022-12-01 03:27:52 PM EST

Table of Contents

Overview

In this guide we'll delve into the popular programming language Scratch ().

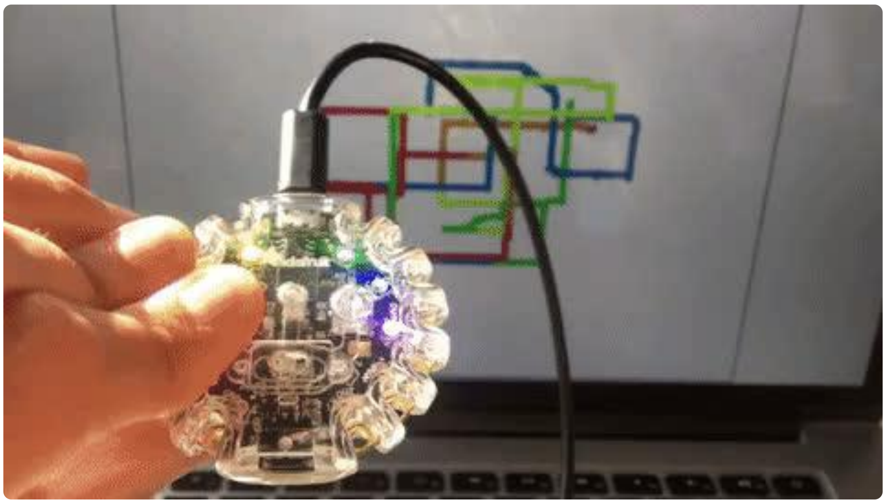
Scratch is a block-based language geared for beginners that lets you create your own interactive stories, animations, games, music, and art.

Scratch is easy to use and requires no prior coding experience!

First we'll do a walkthrough of the basics.

Next we'll cover the updates in the 3rd and newest edition of the language which was released January 2, 2019.

Lastly we'll do some digital painting inside Scratch using a Circuit Playground Express as our "paintbrush" and "palette". See below!



Adafruit Parts

Circuit Playground Express is a great introduction to electronics and programming

1 x Circuit Playground Express (CPX) https://www.adafruit.com/product/3333
Circuit Playground Express is a great introduction to electronics and programming

1 x USB cable https://www.adafruit.com/product/592
USB A to Micro-B (Plugs into computer to program the CPX)

Optional

1 x Adafruit Circuit Playground Express Enclosure https://www.adafruit.com/product/3915
A plastic case for the Circuit Playground Express that makes it look pretty.

Walkthrough

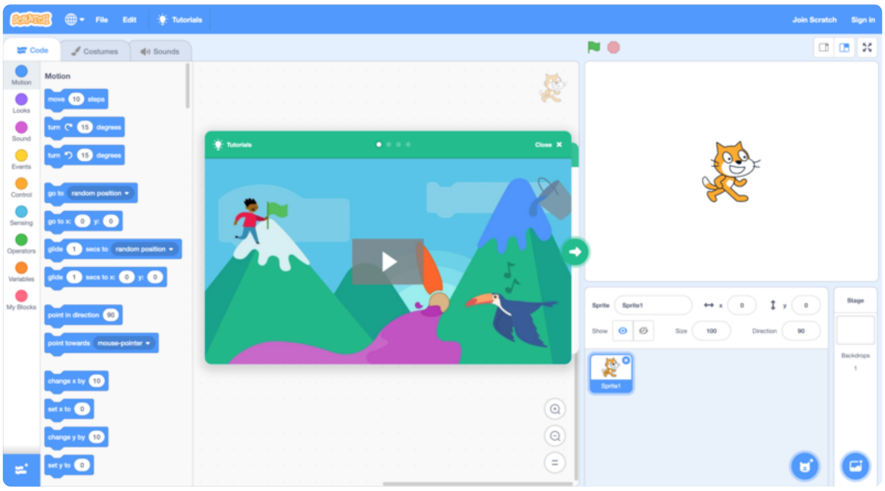
The quickest way to get started with Scratch is to jump right in! The programming language is web-based which means there is no downloading necessary. Just head to the website below.

Click here to get started with Scratch!

After entering the Scratch site, click the orange "Start Creating" button.

You will then be taken to the Scratch project editor. This is where all the coding happens!

Follow the short tutorial to learn the basics of how "sprites ()" work in Scratch.

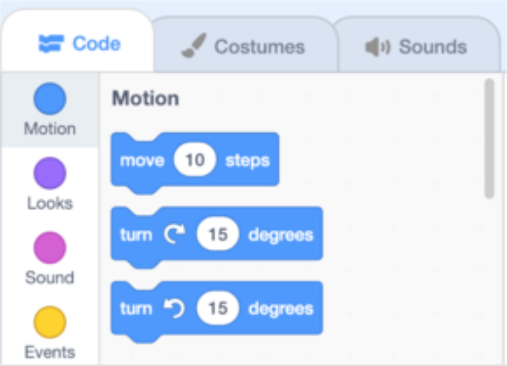


Block types

Scratch is a "block-based ()" programming language. This means instead of writing lines of code to program, we use "blocks ()". Each block has a specific category that describes what its purpose is. Sprites () are the objects that perform actions in a project and these blocks tell the sprites what to do. Code blocks can also tell a stage or backdrop how to behave.

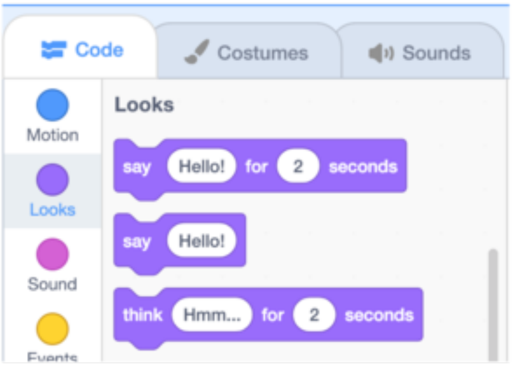
The different types of code blocks:

Motion ()



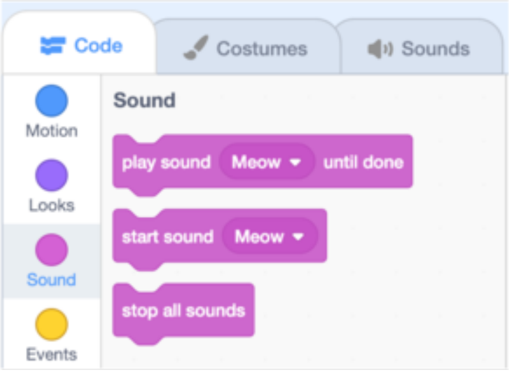
Define the movement of sprites

Looks ()



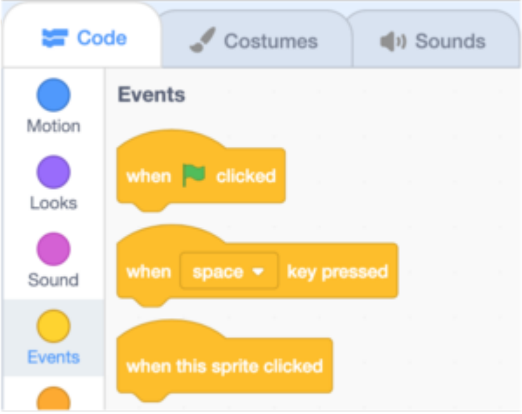
Change the appearance of sprites. Gives the ability to change costumes and apply graphic effects.

Sound ()



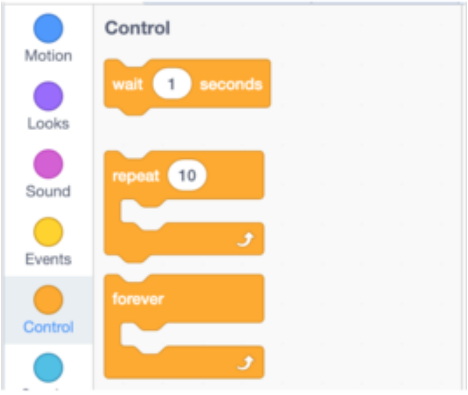
Play audio by recording sounds, using Midi notes () and more.

Events ()



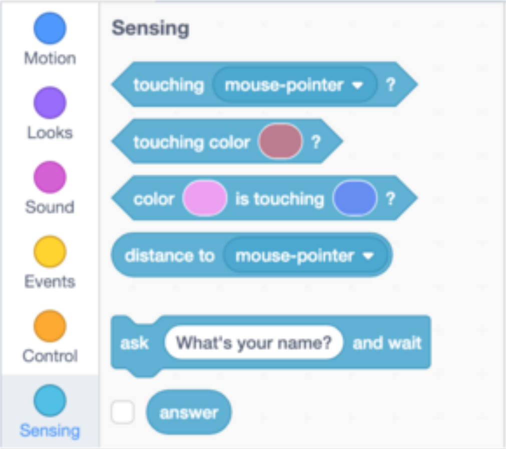
Trigger scripts or series of code blocks. Consists mainly of hat blocks () which start a script.

Control ()



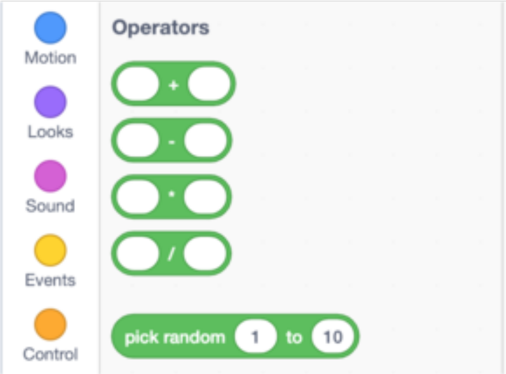
Dictate the flow of a project. Provide mechanism for "looping" different blocks and scripts.

Sensing ()



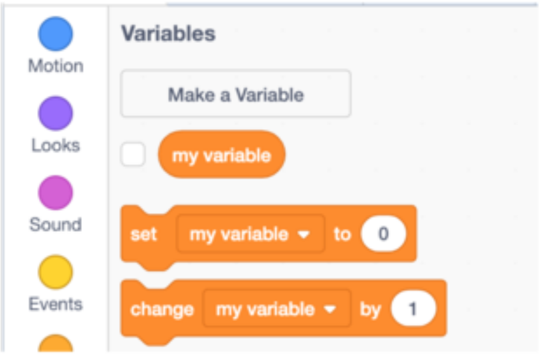
Let multiple sprites interact with themselves and/or the stage. Can determine how two sprites collide for example.

Operators ()



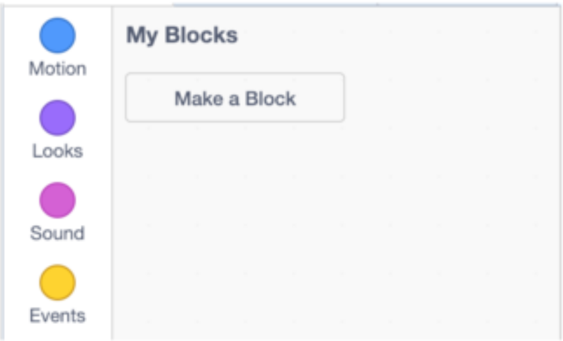
Allow various mathematical functions within a project. Can be used to modify variables and strings ().

Variables ()



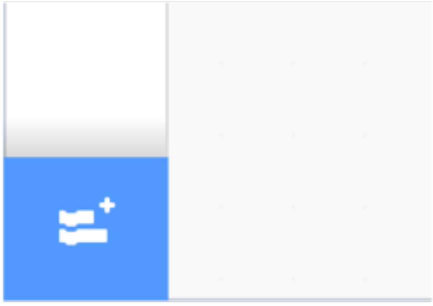
Used to store data in projects, like a name or a score.

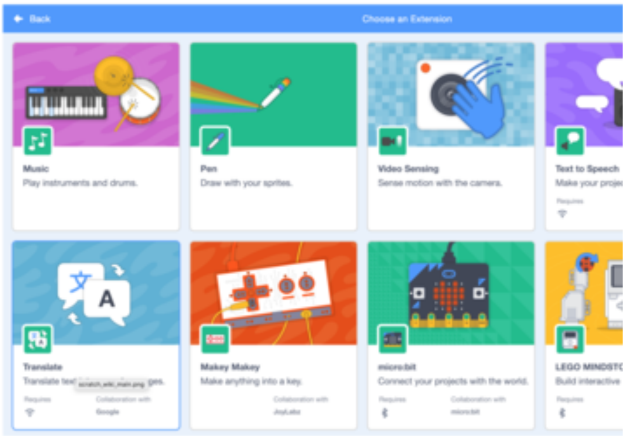
My Blocks



You can create your own blocks!

Extension Blocks

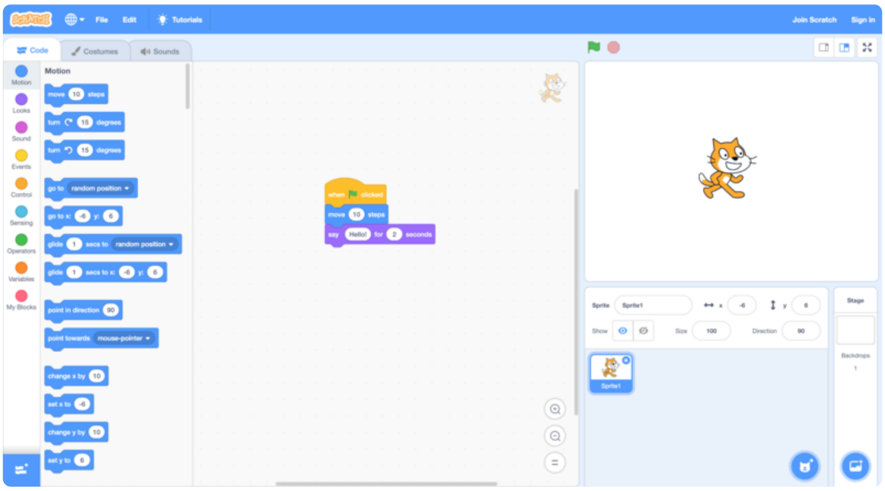




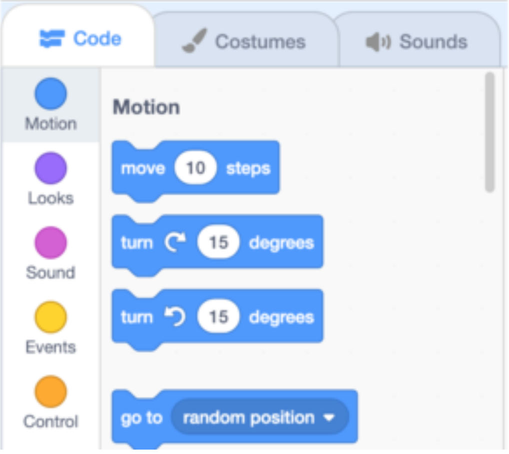
At the bottom left of the screen. These blocks let you connect to hardware or access a number of other block types like the pen, or music blocks.

User Interface

The Scratch user interface () is the area on your screen where the Scratch program exists. The screen is divided into multiple sections or "panes ()" which have different functions from picking blocks to code with, to coding, to seeing your code in action. We will go through the different elements of the interface.

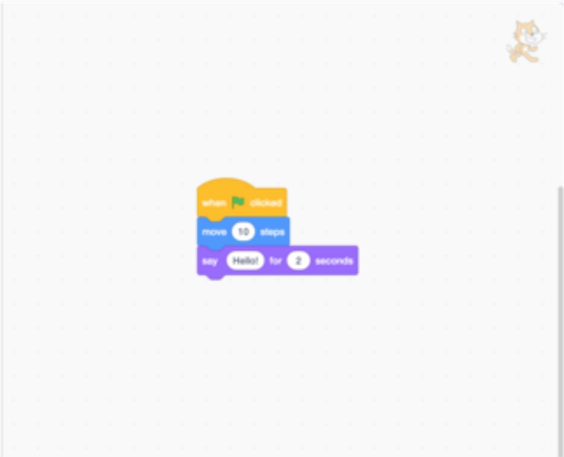


The Block Palette



The block palette is where all the code blocks exist. Blocks are color coded by category. These blocks can be dragged to the scripting area to either code sprites or the stage.

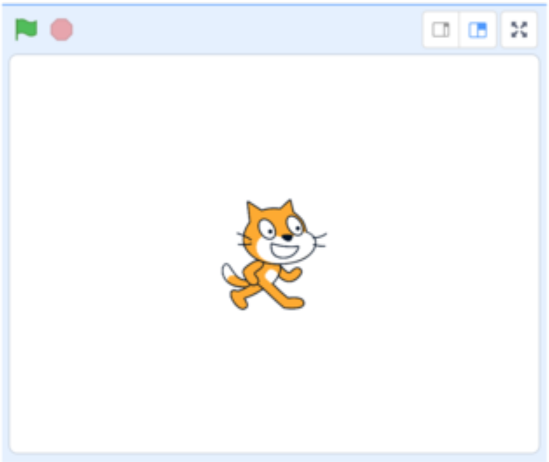
The Script Area

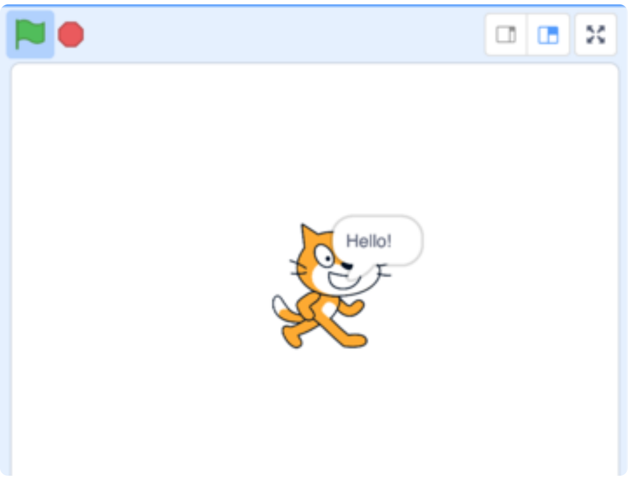


```
when 🏁 clicked
move 10 steps
say Hello! for 2 seconds
```

The scripts area is where our code is dragged to and assembled.

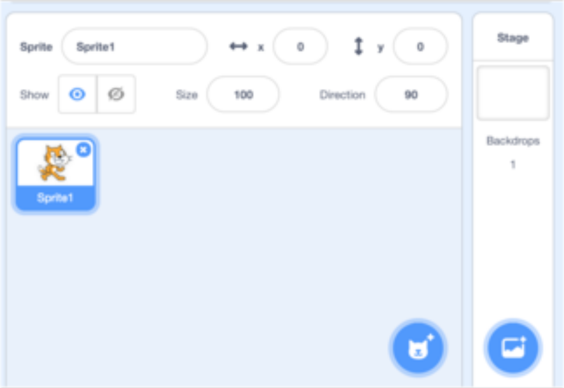
The Stage Area





The stage is where we can see our code come to life! For example if we have a "when green flag clicked" event block at the start of our code, we can click the green flag to trigger the code we wrote.

The Sprite Info Pane

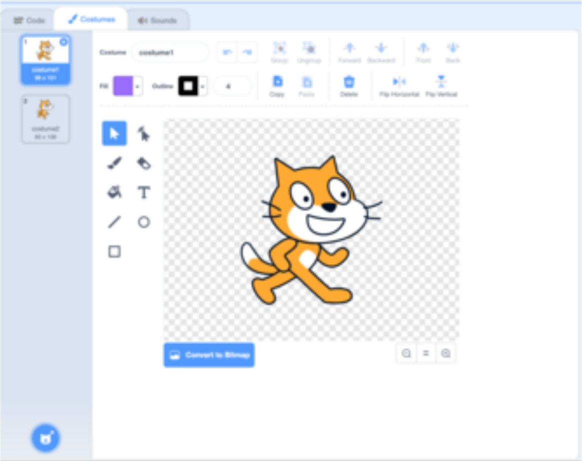


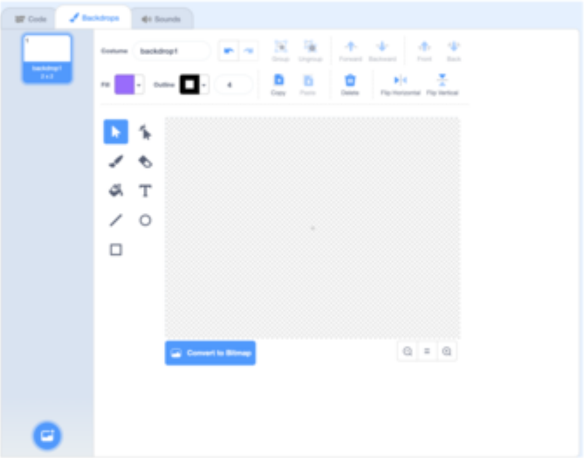
Sprite Sprite1 x 0 y 0

Show Size 100 Direction 90

The sprite info pane is where we can find info about our sprites as well as manipulate them. Access this pane by clicking on the thumbnail of the desired sprite. This area also allows us to delete and create new sprites.

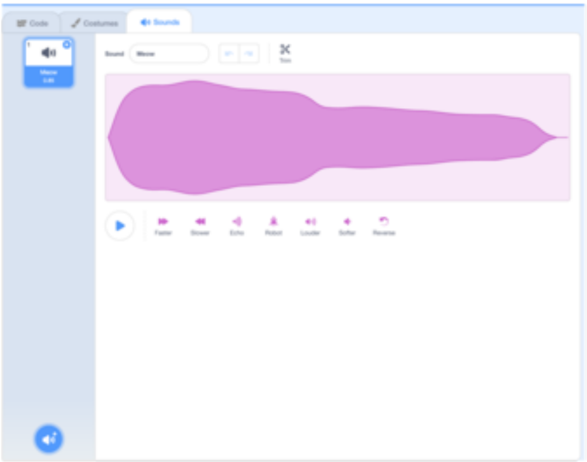
The Costume/Backdrop Pane





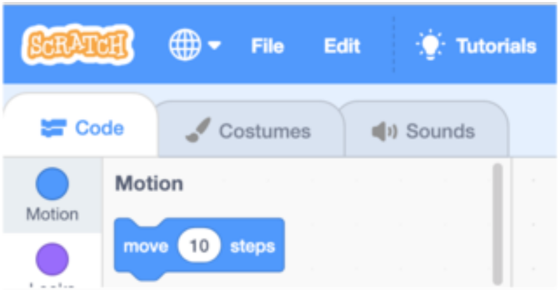
The costume and backdrop panes can be accessed by clicking the middle tab in between the "code" tab and the "sounds" tab. This is where we can create and manipulate sprites and backdrops. To switch between costume and backdrop panes, select the desired thumbnail in the sprite info pane.

The Sounds Pane

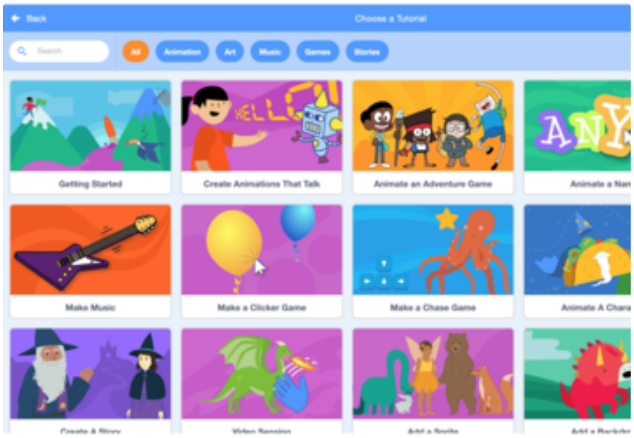


The last tab in the top left area of the interface allows us to crate and manipulate sounds. You can even record your own!

Toolbar and Tutorials



The toolbar on the top of the screen allows us to load and save projects as well as undo the deletion of sprites.



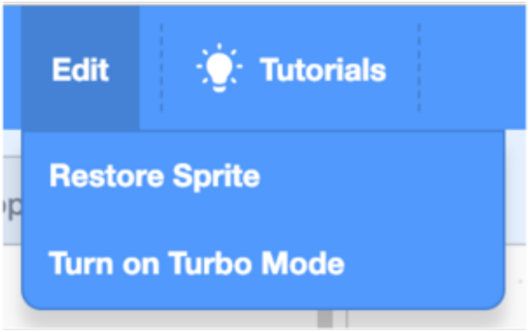
Find a variety of tutorials on how to do just about anything in Scratch by clicking the "tutorials" button!

Tips and Tricks

Workflow and mistakes

- Copy and paste code blocks to save time.
- Be careful when creating and deleting sprites! The code blocks you create tell the associated sprite what to do. Thus when you delete a sprite you are deleting the code that tells it what to do.

Deleting a sprite will delete all the code associated with that sprite!



If you accidentally delete a sprite, click edit on the tool bar then click "Restore Sprite" to bring your code back to life!

Troubleshooting

- Right click a block for help.
- Visit the Scratch wiki () for more help.
- There's a whole community of Scratchers () that have come across a multitude of issues and would also love to help.
- To join the community make an account!

What's New in Scratch 3?

Scratch 3.0 includes a new sound editor, new sprites, new programming blocks, and much more.

A new build

Scratch was originally built with Adobe Flash which is going away soon so the new update is built in HTML5 () and a Javascript library called React ().

What does that mean for us?

Scratch can now be used on tablets and almost all hardware, like Raspberry Pi for example!

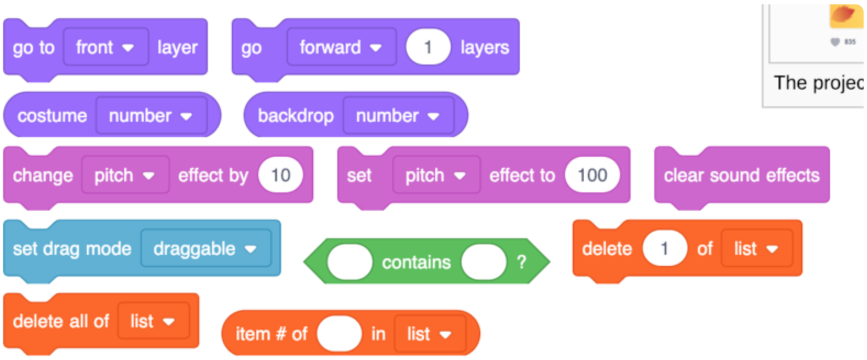
Scratch Extensions

Scratch extensions is a new feature that allows for Scratch programs to talk to external hardware and online resources. Micro:bit () even has its own extension!

Other updates per the Scratch Wiki ():

- Stage is moved to the top right.
- New modern look and design.
- Some color changes to blocks. Events Blocks are the color of the Scratch 2.0 Control Blocks, and control blocks are a bit lighter than the old events blocks
- More Blocks has been renamed My Blocks (), and is a shade of light red
- Some default inputs (the inputs that show up in the Block Palette () ar e different)
- New Paint Editor () and Sound Editor ()
- Pen (), Video Sensing (), and Music () blocks are now extensions
- You can now make and play projects on tablets and play them on phones, but currently you cannot edit projects on mobile phones
- Several extra options for the “touching ()?” boolean, including “any edge”, “mouse-position”, and “left edge”
- Different fonts: Sans Serif (thin Helvetica), Serif (Times), Handwriting, Marker, Curly, Pixel, Chinese, Japanese, and Korean
- The vector paint editor is now the default editor
- The Video Sensing blocks are in an extension
- New extensions: Text to Speech (), Translate (), Micro:bit (), and Lego Mindstorms EV3.
- Internet Explorer will not be a supported browser

- PicoBoard () and LEGO WeDo () 1.0 are not supported
- The addition of a pause button like in Phosphorus () and Sulfurous ()
- New blocks!! (below) Click here () to learn more about the functionality of these blocks.



```
go to front ▾ layer
go forward ▾ 1 layers
costume number ▾
backdrop number ▾
change pitch ▾ effect by 10
set pitch ▾ effect to 100
clear sound effects
set drag mode draggable ▾
contains ?
delete 1 of list ▾
delete all of list ▾
item # of in list ▾
```

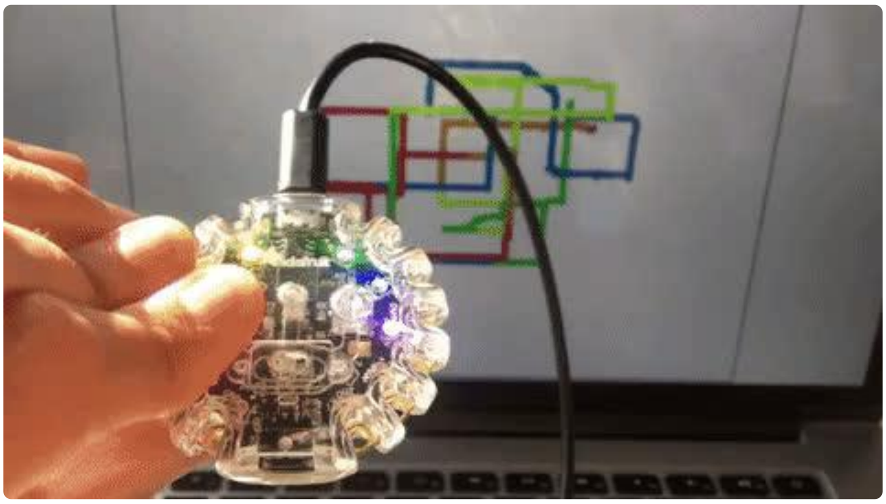
The projec

Draw Demo!

In this demo we will create a simple draw program in Scratch.

Next we will create a program in MakeCode () that will allow a Circuit Playground Express (CPX) to control the mouse and keyboard functions that we define in Scratch.

Lastly we create a masterpiece!



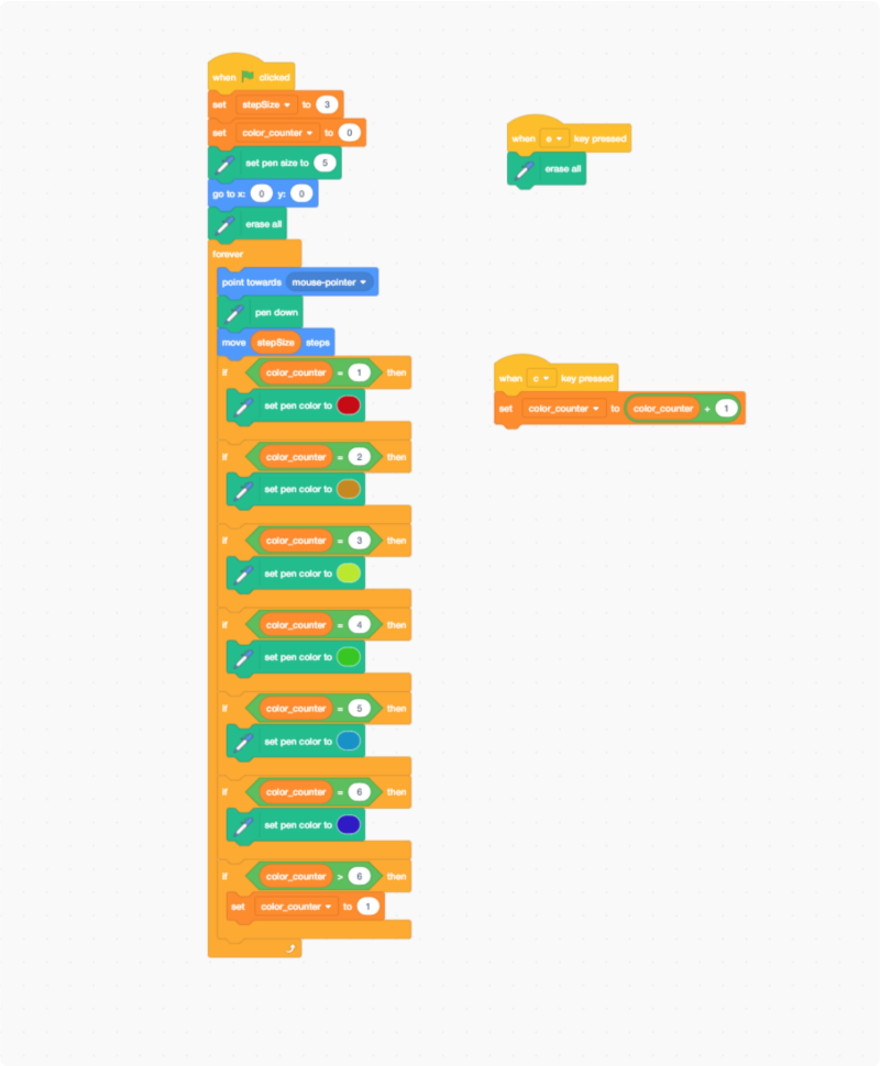
The best way to learn a new language is to start making with it, so let's go!

Scratch Code

Find the Scratch code to this project below. Feel free to follow along or access the code for yourself!

Click here to access the Scratch code for this project

After clicking the link above, you'll be taken to the project page. Once there, click see inside to access the code.



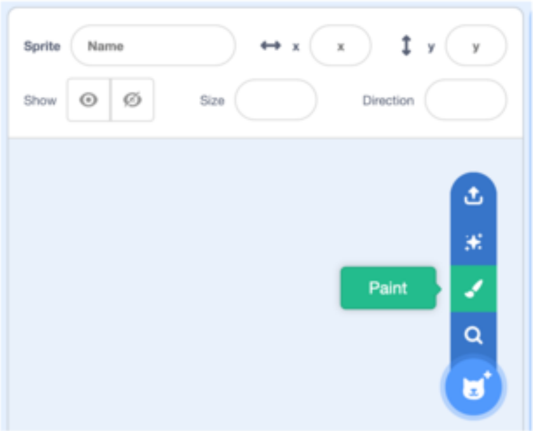
```
when 🏁 clicked
set stepSize ▾ to 3
set color_counter ▾ to 0
set pen size to 5
go to x: 0 y: 0
erase all
forever
    point towards mouse-pointer ▾
    pen down
    move stepSize steps
    if color_counter = 1 then
        set pen color to 🔴
    if color_counter = 2 then
        set pen color to 🟤
    if color_counter = 3 then
        set pen color to 🟡
    if color_counter = 4 then
        set pen color to 🟢
    if color_counter = 5 then
        set pen color to 🔵
    if color_counter = 6 then
        set pen color to 🟣
    if color_counter > 6 then
        set color_counter ▾ to 1
```

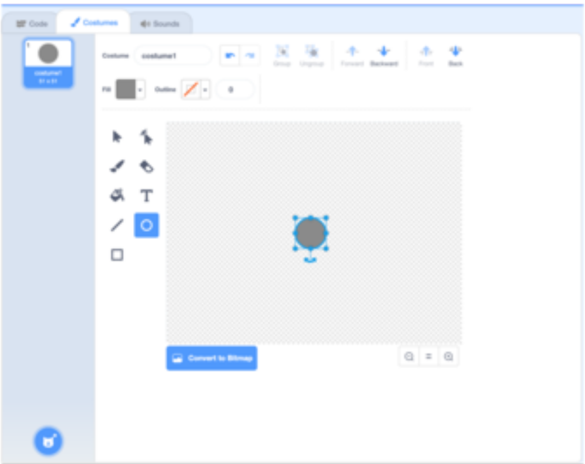
```
when e ▾ key pressed
erase all
```

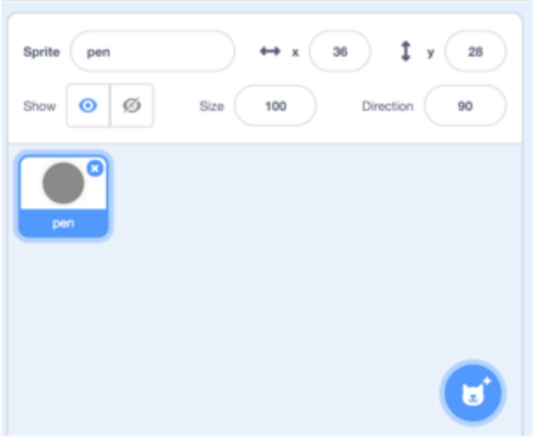
```
when c ▾ key pressed
set color_counter ▾ to color_counter + 1
```

Create a sprite for the pen

The first thing we want to do is make a new sprite for our pen. This sprite will not only be the visual representation of our painting tool but it will also hold all the associated code.



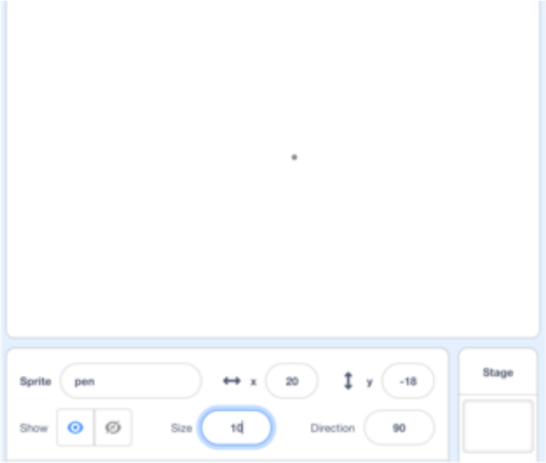




Sprite pen x 36 y 28

Show Size 100 Direction 90

pen



Sprite pen x 20 y -18

Show Size 10 Direction 90

In the sprite info pane, delete any current sprites.

Next, to create a new sprite, hover the mouse over the cat with a plus sign. Now find and click the paint brush icon.

This will bring us to the sprite costume pane where we can create our very own sprite!

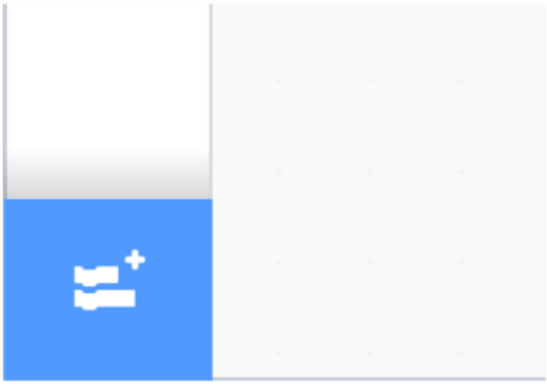
Our pen sprite is simple but feel free to make it extra fun.

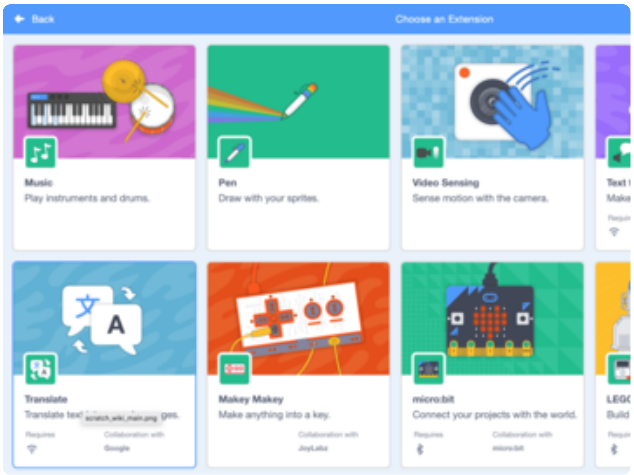
Click the circle shape in the toolbar on the left, next disable the outline, lastly we choose the color we want for the sprite in the fill drop down. Then click and drag a small circle or whatever shape we want in the canvas.

Now we see our new sprite in the sprite info pane, so let's go ahead and name it.

The last step is to change the size of the sprite by typing in the desired size in the sprite info pane.

Adding the pen extension blocks

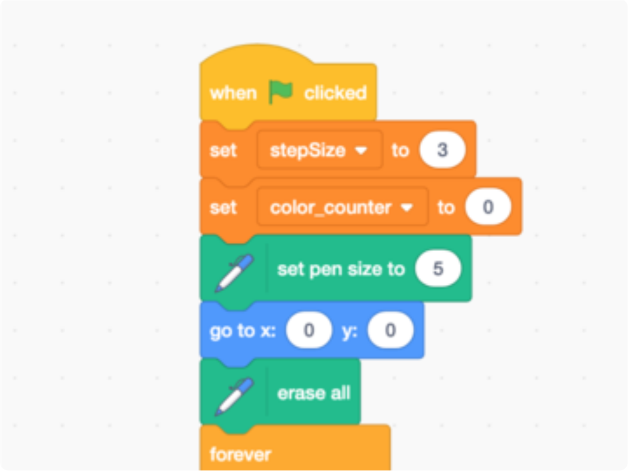




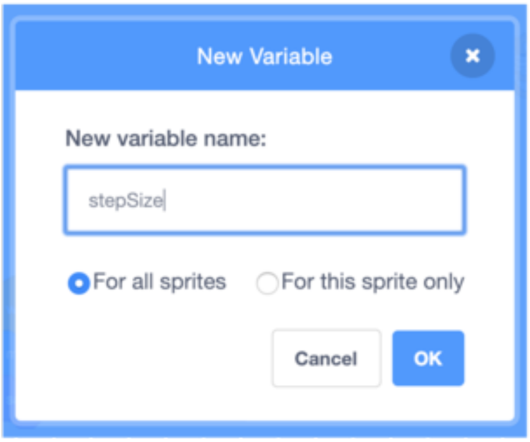
In order to access the pen blocks in scratch, we need to grab the extension. Click the extension icon on the bottom left of the blocks palette.

Next click the pen extension to add it.

Setting up the program



```
when 🏁 clicked
set stepSize to 3
set color_counter to 0
set pen size to 5
go to x: 0 y: 0
erase all
forever
```



New Variable

New variable name: stepSize

● For all sprites ○ For this sprite only

The first thing we want to do is tell the program to start after the green flag is clicked.

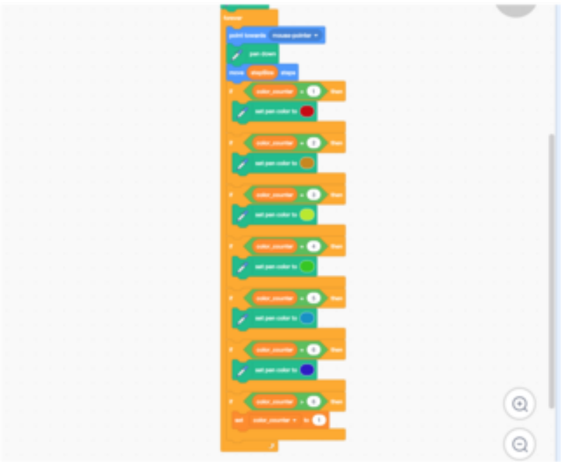
Next we create a variable called `stepSize` and set it to `3`. This variable will determine how fast we want the pen to follow the mouse. Then create another variable called `color_counter` and set this to `0`. This will help us switch colors more easily.

Next set the pen size to `5`. This is the thickness of the pen.

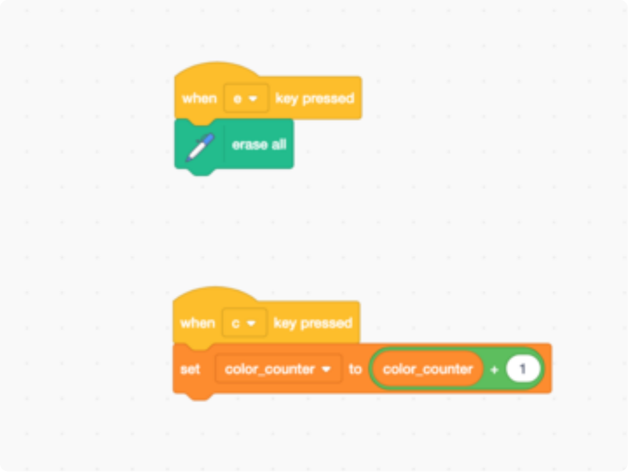
Then we tell the pen to start in the middle of the screen at `0, 0`.

Lastly we want to erase everything that's currently on the stage before the forever loop begins.

Forever loop



In a forever loop, we add a `point towards mouse-pointer block`. Next, goes a `pen down` block. Then drag a `move x steps` block, dragging in the variable `stepSize` from the variables blocks in place of the number. Now we create a series of if statements that check for the `color-counter` value which will tell the program when to switch colors and what color to switch to. Each time the character c is typed on the keyboard, the value of `color-counter` increases by 1 (we will get to this next). However, with the last if statement, when the value reaches above 6, it is reset back to 1.

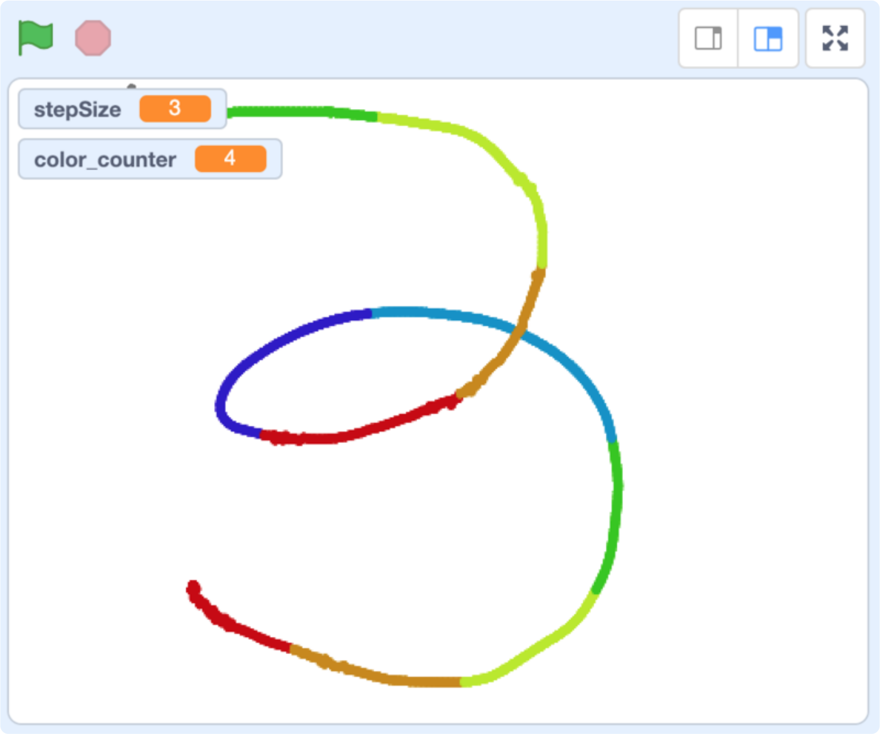


```
when e ▾ key pressed
erase all

when c ▾ key pressed
set color_counter ▾ to (color_counter + 1)
```

The last few blocks of code define functions for pressing keys e (erase) and c (color). When e is pressed, we erase the board. In the next section, when we code the CPX, we tell it to send the character e when it is shaken. When c is pressed we increase the value of `color-counter` by `1`. Later this will get translated into: when button A is pressed on the CPX, send the character c.

Test it out!

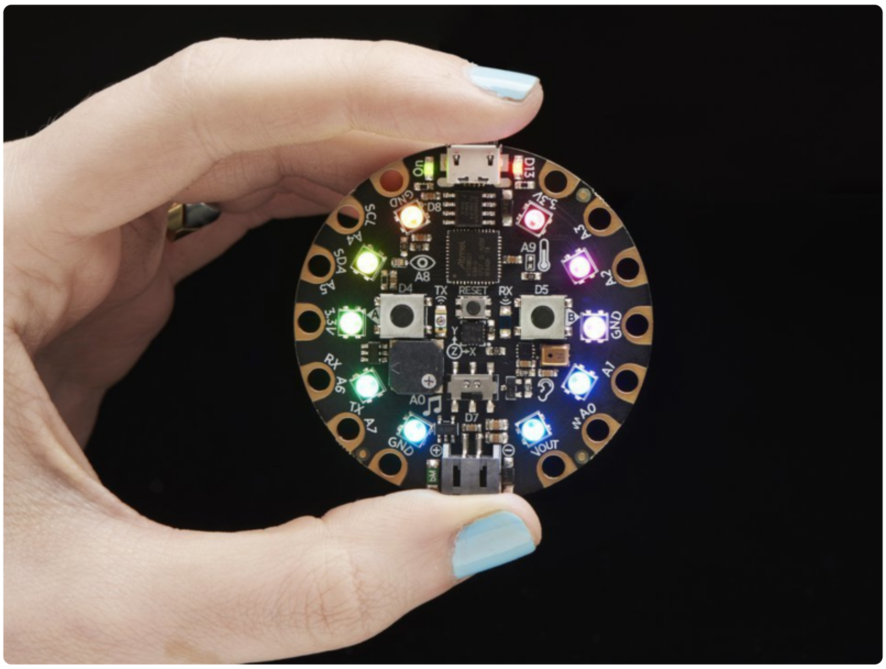
Now that the Scratch program is finished, test out the code in the stage by clicking the green flag to start. Move the mouse around to guide the pen. Press c to change colors and e to erase the board.



CPX and MakeCode

What is Circuit Playground Express (CPX)?

The Circuit Playground Express is an easy way to get started learning electronics and coding. It's filled with tons of fun lights and sensors! The CPX is what we'll use to control the mouse and keyboard to draw in our new Scratch program. Head here () to learn more about the CPX.



What is MakeCode?

MakeCode is a web-based code editor for physical computing made by Microsoft.

It is almost identical to Scratch in design, however it is mainly for programming hardware like the CPX rather than a sprite on a computer screen.

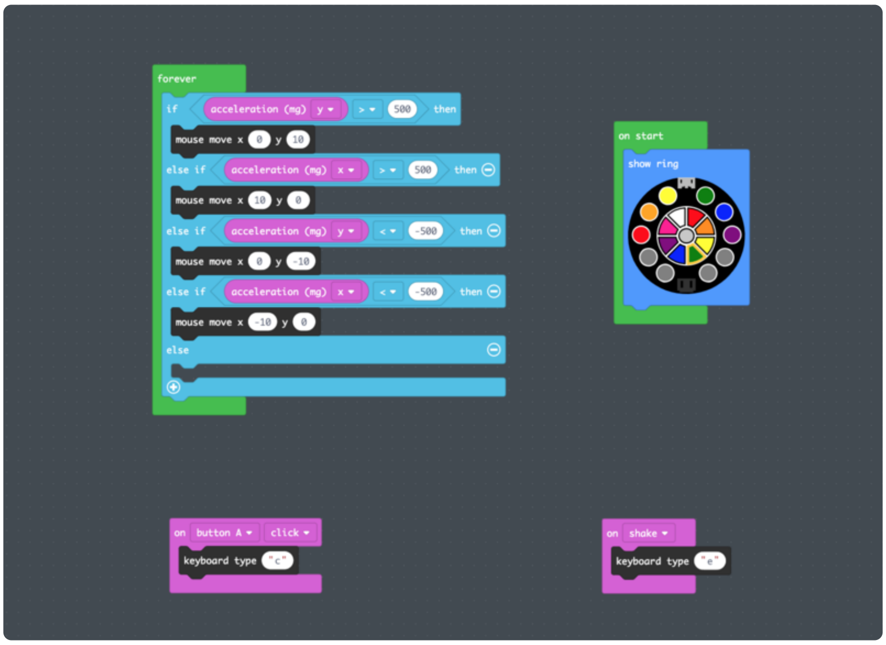
More on MakeCode for Adafruit boards here ().

Create a New Project with MakeCode

Head over to https://makecode.adafruit.com/ () and create a new project.

Click the button below to access the code for this project.

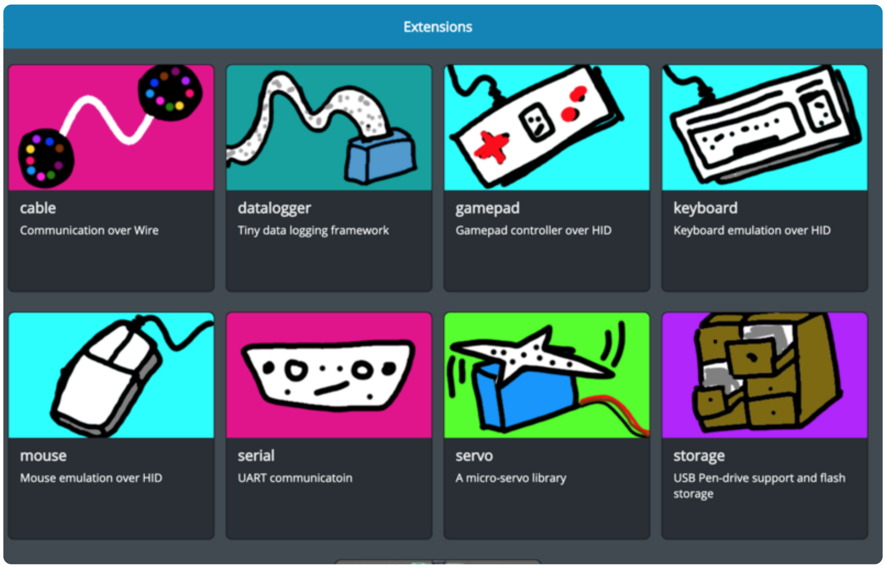
Click here to open the project up in MakeCode



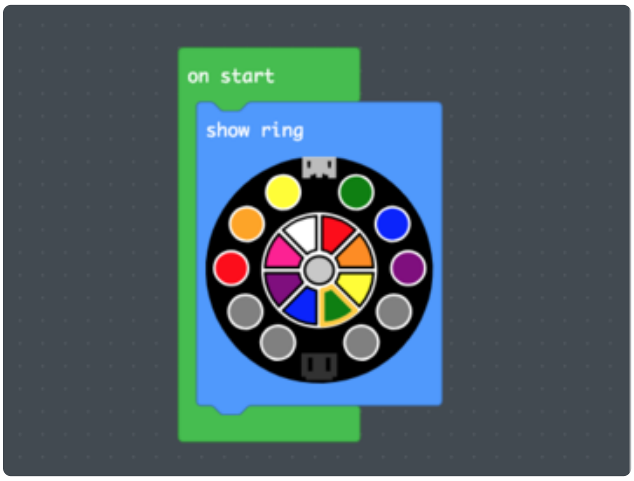
```
forever
  if acceleration (mg) y > 500 then
    mouse move x 0 y 10
  else if acceleration (mg) x > 500 then
    mouse move x 10 y 0
  else if acceleration (mg) y < -500 then
    mouse move x 0 y -10
  else if acceleration (mg) x < -500 then
    mouse move x -10 y 0
  else

on start
  show ring

on button A click
  keyboard type "c"

on shake
  keyboard type "e"
```

Add the Keyboard and Mouse Extensions in MakeCode

Follow this guide () on how to add these extensions to MakeCode. They will allow us to use the CPX to act as a mouse and a keyboard!



The NeoPixel Palette



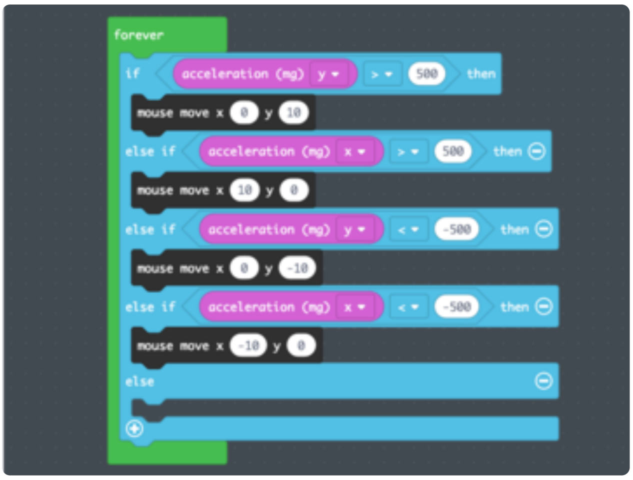
In an `on start` loop, set the colors of the NeoPixels on the CPX to the according colors. This is so the user knows which colors they have to choose from.

Coding with the accelerometer

1 G (one gee) is equivalent to Earth's gravitational pull at a resting point.

The CPX's accelerometer senses the gravity it feels in any given direction on the x, y or z axis. The value it senses is anywhere from 0 to 1023 mg (miligravities). 1G = 1000 miligravities.

Learn more about the accelerometer on the Circuit Playground Express here ().

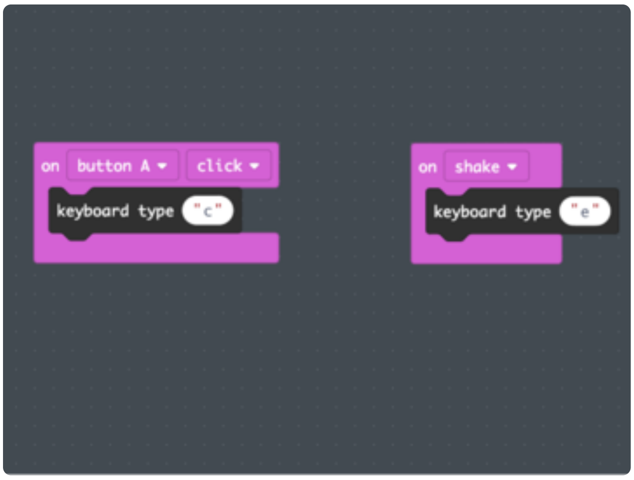


```
forever
  if acceleration (mg) y > 500 then
    mouse move x 0 y 10
  else if acceleration (mg) x > 500 then
    mouse move x 10 y 0
  else if acceleration (mg) y < -500 then
    mouse move x 0 y -10
  else if acceleration (mg) x < -500 then
    mouse move x -10 y 0
  else
```

In a `forever` loop, we create an `if else` statement that describes how the mouse should move when the CPX is tilted forwards, to the right, backwards, or to the left.

If the accelerometer senses any more that 500mg in any direction, the mouse will move in that direction.

Sending keyboard info



```
on button A click
  keyboard type "c"

on shake
  keyboard type "e"
```

When button A is clicked, the CPX sends the character c for color change.

When the CPX is shaken, the CPX sends the character e for erase.

Uploading the Code

Now that we have the code for the project, we need to upload it to the CPX.

Let's name our file and download it.

- Choose a name at the bottom of the page.
- Then click the pink Download button.

Follow the directions and connect your CPX to your computer via the USB cable. Click the CPX Reset button once to go into programming mode (all the Circuit Playground Expresses' LED light NeoPixels will turn green).

In some cases, you may need to press the reset button twice to get into programming mode.

Next, look for the file in your downloads folder and drag it onto your CPLAYBOOT driv e that should have showed up in your file manager/finder when you plugged in your board and entered programming mode.

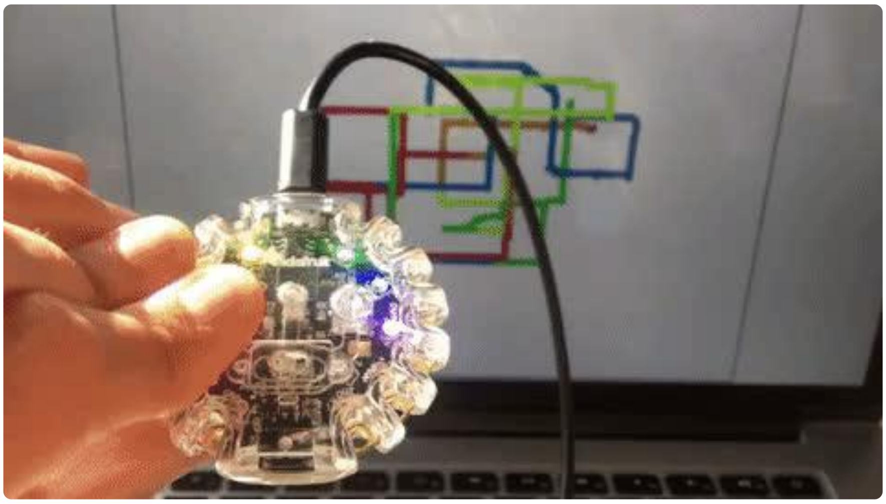
Once your code file is placed on the CPLAYBOOT drive should see the CPLAYBOOT drive disappear indicating the code is running.

Putting it All Together

The code should now be running! Try tilting around the CPX and see if the mouse moves.

Head to our Scratch program, enter full screen and test it out!

- Tilting the CPX moves around the pen.
- Clicking button A on the CPX changes colors.
- Shaking the CPX erases the picture.



Troubleshooting

- If the pen in scratch is acting glitchy as you tilt the CPX around, try decreasing the value of `stepSize`.
- Head here () to troubleshoot MakeCode issues.
- For CPX related issues head here ().
- Head to the Scratch Wiki () to troubleshoot Scratch related issues.

What will you create next?

How can you modify what we created to make something even cooler? What other sensors on the CPX could we use to add more effects with the pen in Scratch?

Happy Hacking!