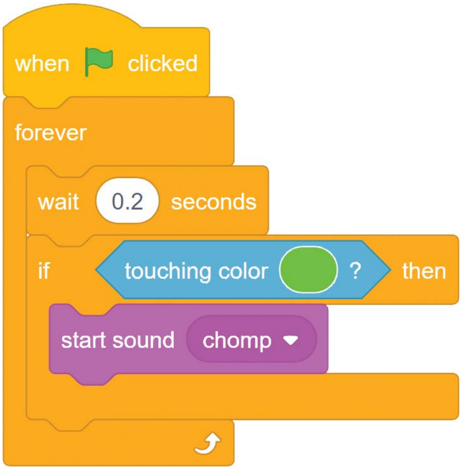
We'll use Scratch's simple, block-based programming language to code the interactive parts of our game. Scratch uses a system of blocks that represent different functions. They snap together like puzzle pieces into complete lines of code.



```
when ⚑ clicked
forever
  wait 0.2 seconds
  if touching color 🟩 ? then
    start sound chomp ▾
  
↻
```

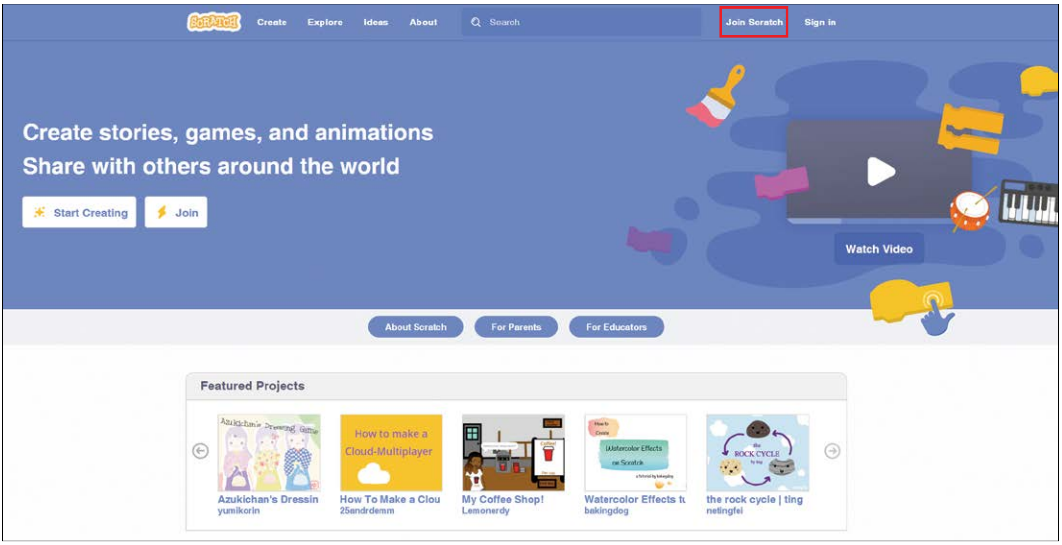
How Scratch code blocks fit together

These stacks of code blocks tell the objects in your games what to do. As you'll see shortly, using code blocks in Scratch is very easy.

Getting Started with Scratch

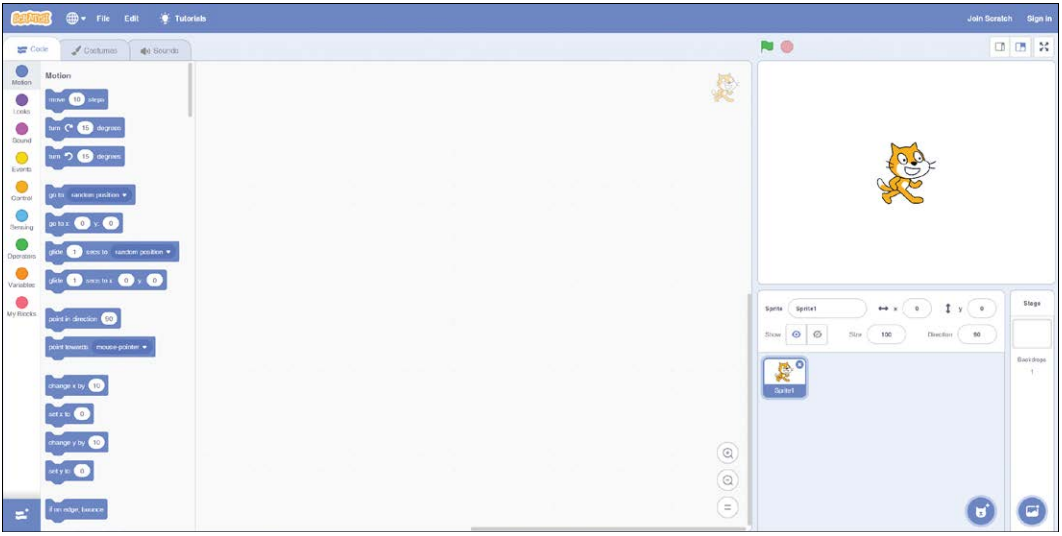
To begin, you need to sign up for your own Scratch account. It's free, and after you've registered, you can start saving and sharing your games. In your web browser, open the official Scratch website at *https://scratch.mit.edu/*.

On the Scratch website, click **Join Scratch** in the upper right. When asked, add your email address and some information about yourself. Then you'll get an email asking you to confirm your email address. Check your inbox and click the link in the email. Bam! You should now have your own Scratch account.



The Scratch home page

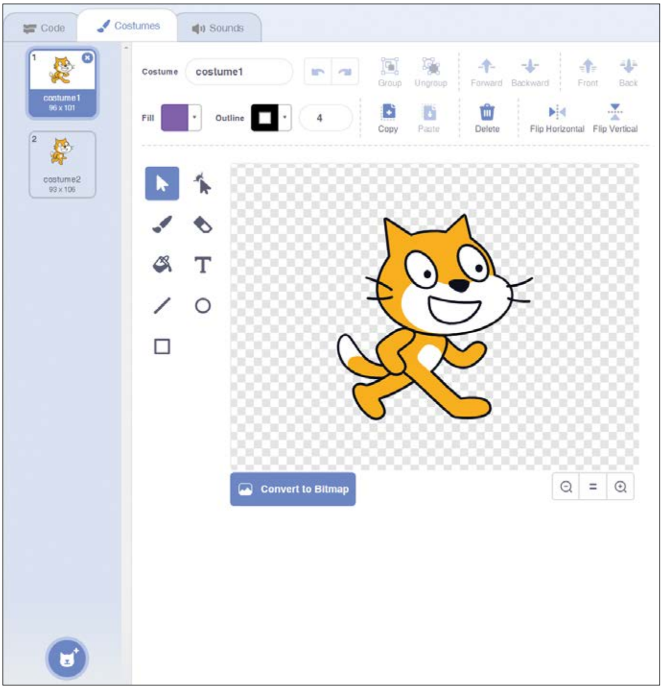
Click the **Create** button at the top of the Scratch home page if you're not already in Scratch's editor. You'll see a blank white box with a smiling orange cat, as shown in the following figure. This box is the game window, which is the only part of the editor your players will see. The cat is your first sprite. Let's talk about what a sprite is and what you can do with it.



Opening the Scratch editor

That orange cat is Scratchy, Scratch's mascot. With every new Scratch project, Scratchy is automatically included in a noble effort to encourage more people to make games about cats.

Does Scratchy love lasagna? Does Scratchy hate Mondays? No one knows. But we *do* know that Scratchy is a *sprite*, which is any object in a Scratch game that can do something, like change its appearance or play music. Each sprite can have scripts, costumes, and sounds attached to it. A *script* is a group of coding blocks that fit together, telling the sprite what to do. A *costume* is a snapshot or a frame of a sprite that you can use to create animation. For example, the pose Scratchy is in right now is one of two default costumes that come with Scratchy. Click the **Costumes** tab on the top left (between Code and Sounds) to see the Scratchy sprite's costumes.



Costume costume1

Fill

Outline 4

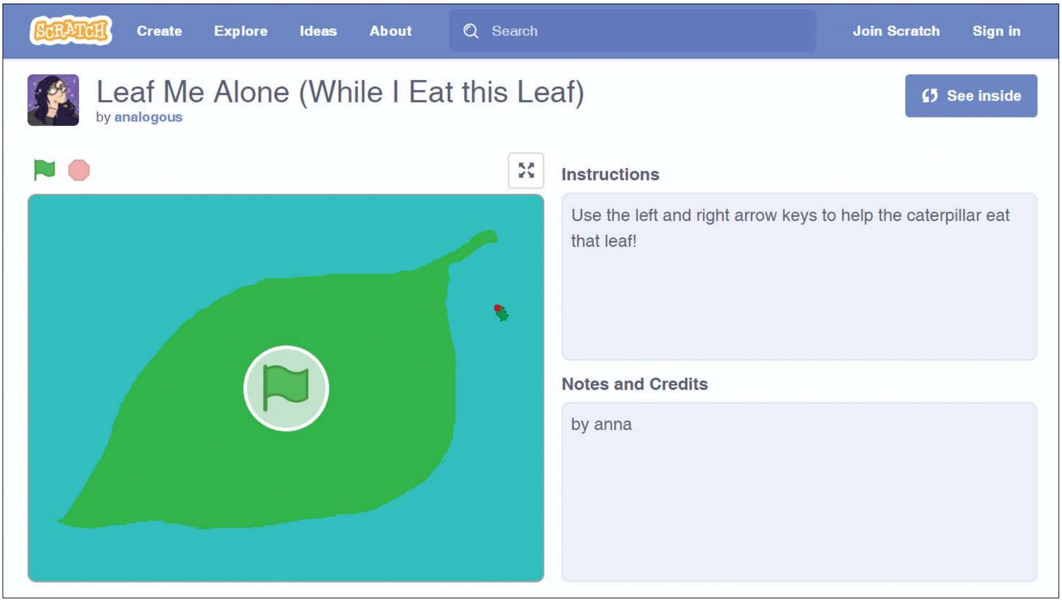
Scratchy comes with two costumes

When you flip between the two costumes by clicking one and then the other like a flip book, Scratchy should look they're walking. We'll use costumes to animate our sprites or change their appearance.

Sprites can also have sound effects that they can play. For example, Scratchy comes with a meow sound. Click the **Play** button in the Sounds tab to hear it.

Leaf Me Alone (While I Eat This Leaf)

We'll create a simple game in Scratch called *Leaf Me Alone (While I Eat This Leaf)*. It's about a hungry little bug who wants to eat a leaf and wants to do so in solitude. You can first play it at *https://scratch.mit.edu/projects/117199134/*, and then we'll create it.



Leaf Me Alone (While I Eat this Leaf)

by analogous

Instructions

Use the left and right arrow keys to help the caterpillar eat that leaf!

Notes and Credits

by anna

Exploring the Leaf Me Alone *game*

Click the green flag to start the game. The little bug starts moving forward automatically! Press the left or right arrows to make the bug change direction. As the bug eats the leaf, it leaves a trail of munches behind. This hungry little bug never stops munching. When you want to stop the game, click the red stop sign in the upper-right corner. If you click the green flag again, the game will start over.

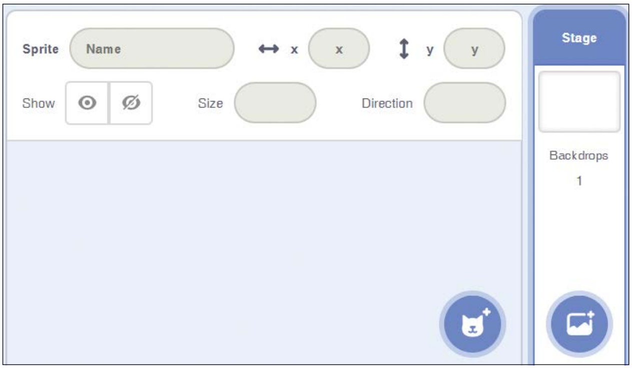
It seems pretty simple, right? Now, let's make this game from *scratch*!

Creating Your Own Sprites

You'll need to draw your *own* sprites for the *Leaf Me Alone* game. One of Scratch's cool features is that you can use the mouse to draw characters for your game. Then you can give them instructions to make them do different things. It's okay if they turn out messy or imperfect. Don't worry!

Right-click Scratchy and click **Delete**. *Zoink!* Scratchy disappears. (You won't really hear a *zoink* sound. I made that part up.)

Next, look at the Sprites list below the game window.



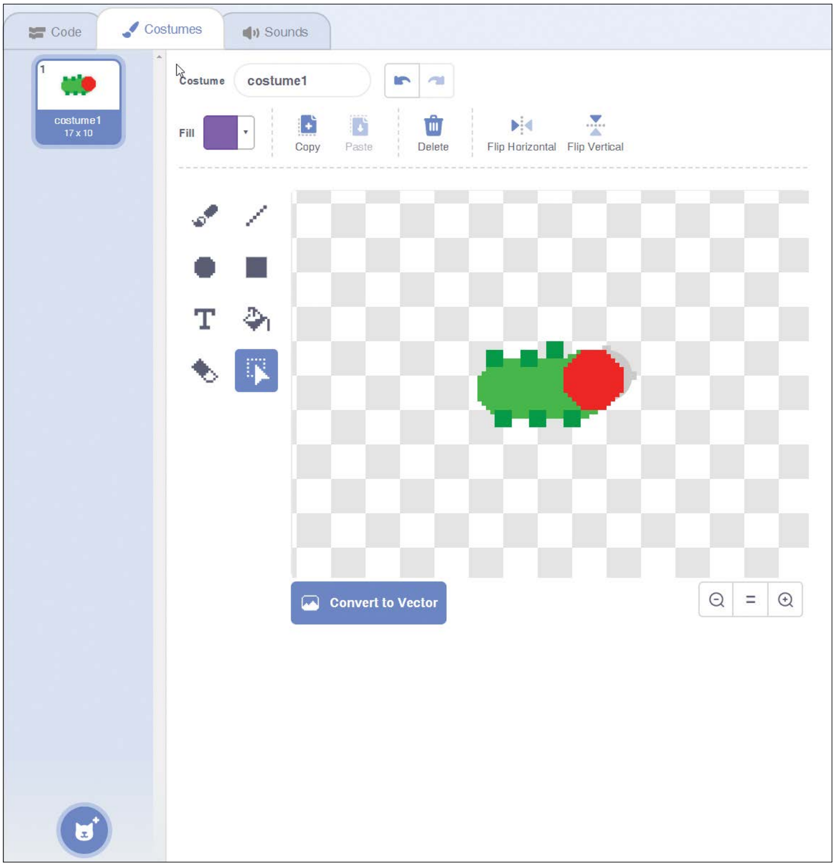
Opening the Sprites list

The *Stage*, which is currently a plain white rectangle, is the background for the game. When you click it, it should be surrounded by a blue highlight. Because we deleted Scratchy, no sprites are in our Stage right now. Let's fix that by adding a new sprite!

Hover over the cat icon below the Stage to display four more buttons. Click **Choose a Sprite** to select a sprite from the Sprites Library. The Paint button lets you create your own sprite. The Surprise button selects a random sprite, and the Upload button lets you use an image from your computer.

Drawing Weird Bugs

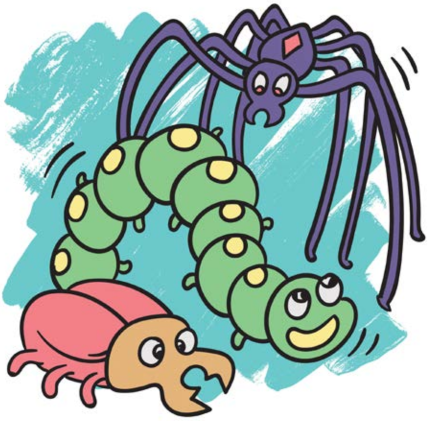
Let's click the **Paintbrush** to draw a sprite for our game. Then click the **Costumes** tab. Now you're in the costume window where you can draw a costume for your new sprite.



Drawing a new costume for your bug

On the left side of the drawing area is a menu of drawing tools. Make sure the paintbrush is selected. On the top of the drawing area is a color selector. To the right of the paintbrush icon is a number you can change to control the size of the paintbrush.

I drew a caterpillar, but you can draw any kind of bug you'd like. You should see a tiny plus sign (+) in the middle of the drawing area. That is the center point of the sprite: draw the bug's head there so its head is about the size of the plus sign (+).



That's a pretty small bug! To make drawing easier, use the magnifying glass buttons in the lower right to zoom in so you can see the bug better. The magnifying glass with the plus sign (+) zooms in, and the minus sign (–) zooms out. You can see what your sprite looks like at its normal size by looking over at the game window.

You can make your bug look however you want, but make sure it's facing toward the right. Scratch will automatically rotate your bug as it walks around the leaf, but sprites should always start by facing toward the right. So, if you want to give your bug a big weird butt, make sure its butt is pointing to the left side in the opposite direction of the head.

If you make a mistake while drawing your sprite, click the Undo button, which is the left arrow above your drawing. This button will undo the last change you made. If you decide you didn't want to undo the change, you can click the Redo button, which is the right arrow at the top of your drawing.

When you're happy with your bug, look at the Sprites list below the Stage again. Your bug should be there now, but it will be called empty. That's not the best name for a bug, but you can change it. Enter your new name into the Sprite field just below the stage. Change its name to something like weird bug or my perfect creation.

Drawing a Backdrop

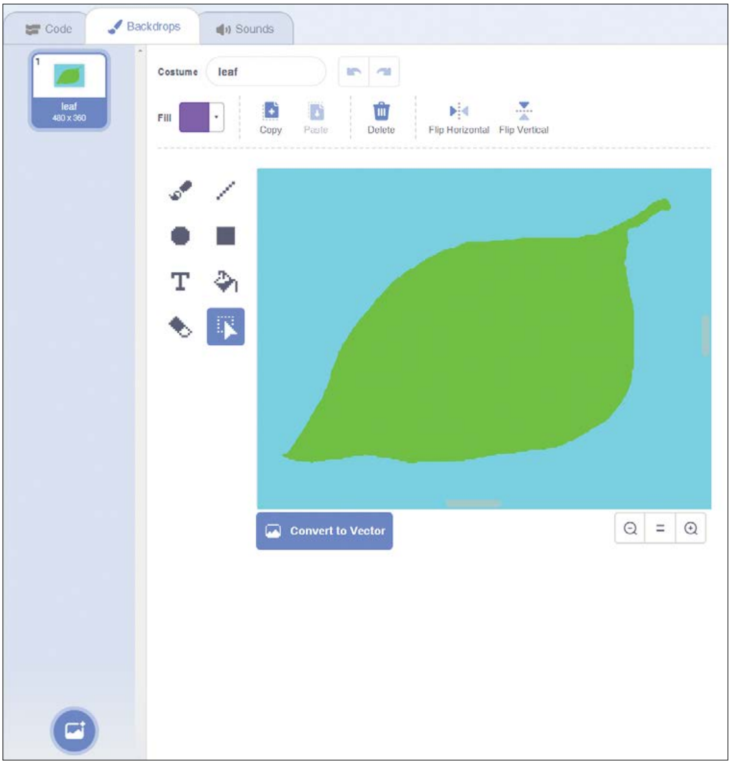
The Stage is a lot like a sprite in that it also has its own scripts, costumes, and sounds. Costumes for the Stage are called *backdrops*. Any image you create for the Stage becomes the backdrop for the entire game window.

You can make multiple backdrops and switch between them at different moments in your game. For example, the first level of your game could take place in a forest, and the second level could take place underwater. For *Leaf Me Alone*, we need only one backdrop because there is only one level. Let's draw it now.

Click the **Stage** (the blank white rectangle) to the right of the Sprites list. The Stage should show the text 1 backdrop below it. Below that is a blue button with a stage icon that works just like the Choose a Sprite button. Hover over it to show four buttons that do the same things as the Sprite buttons but for backdrops.

Instead of making a new backdrop, we'll just paint over the empty white one we started with. If you're not in the drawing window, click the **Backdrops** tab. Let's draw the background for the game: a big juicy leaf for the bug to eat.

If you zoomed in to draw the bug, you might still be zoomed in. Click the button in between the two magnifying glasses (=) to go back to normal view. You should see the entire backdrop. If you can see sliders around the edges to scroll the view, it means you're still zoomed in.



Costume leaf

Drawing the leaf backdrop

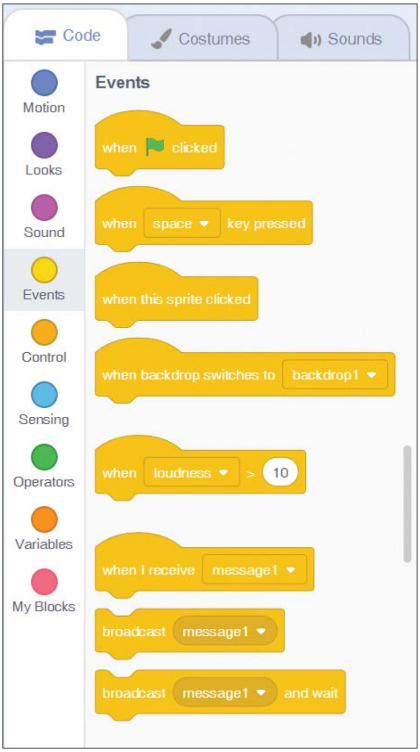
We'll use two colors for this backdrop: one for the background and one for the leaf. (I used sky blue for the background and leaf green for the leaf.) Using just two colors here will be important when you write your program later, so make sure not to add more colors right now. You can also rename the background by clicking the text backdrop1 in the text field and entering a new name.

Now that we have our bug and our leaf, it's time to get our bug moving.

Using Event Blocks

Click your bug in the Sprites list, and then make sure you're in the Code tab. The empty area on the right is where we'll snap our code blocks together to write directions for our sprites. On the left are several categories that hold the different blocks that we can choose. Click a few of the categories to see which code blocks they contain.

For example, click **Events** to view the **Events** blocks, which let you sense events and trigger other code blocks:

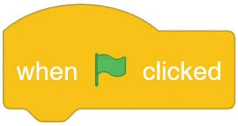


```
when 🏴 clicked
when space ▾ key pressed
when this sprite clicked
when backdrop switches to backdrop1 ▾
when loudness ▾ > 10
when I receive message1 ▾
broadcast message1 ▾
broadcast message1 ▾ and wait
```

*Exploring the **Events** blocks*

Most categories are pretty self-explanatory. **Motion** blocks control movement. **Looks** blocks change the sprite's appearance. **Sound** blocks play sounds.

Every stack of programming blocks you'll write begins with an **Events** block. Click **Events** and you'll see that all the **Events** blocks have big round tops. Like a chef following directions to bake a cake, Scratch always reads directions from top to bottom. Keep in mind that an **Events** block *always* goes first; all other blocks can only attach underneath it.



```
when ⚑ clicked
```

***Events** blocks always go first (at the top).*

Events answer the important question *When do I do this?* The **Events** block we want to use right now is when green flag clicked. Click and drag it into the area on the right.

Recall that you started the example game by clicking the green flag. The game had the when green flag clicked event, which tells the program to run the code blocks when the player clicks the green flag. Our game will start with this **Events** block too. Next, we'll write code that tells our bug to move.

Moving the Bug

Think of a bug crawling along a wall and the weird, winding path it takes. That kind of movement is very easy to replicate. The bug can keep moving forward until the player makes it turn to the left or right. To make our bug move, we'll use the blocks Scratch has for moving forward and turning. The player will have to do their best to guide the bug as it moves nonstop, making the game fun to play.

Let's start by making our bug move forward. The block that moves a sprite forward moves the sprite only one step forward. Because we don't want the bug to stop after just one step, we need to add a loop. A *loop* is a block that makes something happen more than once. To make our bug keep moving, we'll use a loop block to repeat a *move* instruction over and over for as long as the green flag is on.