

1

GETTING STARTED

Have you ever wanted to create your own computer game, animated story, tutorial, or science simulation? *Scratch* is a graphical programming language that makes it easy to create those applications and more.

In this introductory chapter, you will:

- Explore Scratch's programming environment
- Learn about different types of command blocks
- Create your first game in Scratch

When you make a Scratch application, you can save it on your computer or upload it to the Scratch website, where others can comment on it and remix it into new projects.

Excited? Then let's get started!

What Is Scratch?

A computer program is just a set of instructions that tell a computer what to do. You write these instructions using a *programming language*, and that's where Scratch comes in.

Most programming languages are *text based*, which means you have to give the computer commands in what looks like a cryptic form of English. For example, to display “Hello!” on the screen, you might write:

```
print('Hello!')           (in the Python language)
std::cout << "Hello!" << std::endl; (in the C++ language)
System.out.print("Hello!"); (in the Java language)
```

Learning these languages and understanding their syntax rules can be challenging for beginners. Scratch, on the other hand, is a *visual* programming language. It was developed in the Massachusetts Institute of Technology (MIT) Media Lab to make programming easier and more fun to learn.

In Scratch, you won't type any complicated commands. Instead, you'll connect graphical blocks together to create programs. Confused? Look at the simple program in Figure 1-1, and I'll explain.

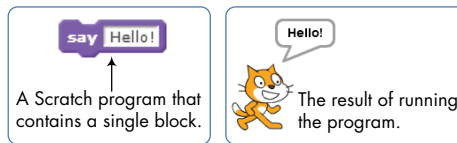


Figure 1-1: When you run this Scratch block, the cat says “Hello!” in a speech bubble.

The cat that you see in Figure 1-1 is called a *sprite*. Sprites understand and obey sets of instructions that you give them. The purple block on the left tells the cat to display “Hello!” in a speech bubble. Many of the applications you'll create in this book will contain multiple sprites, and you'll use blocks to make sprites move, turn, say things, play music, do math, and so on.

You can program in Scratch by snapping those color-coded blocks together as you would puzzle pieces or LEGO bricks. The stacks of blocks that you create are called *scripts*. For example, Figure 1-2 shows a script that changes a sprite's color four times.

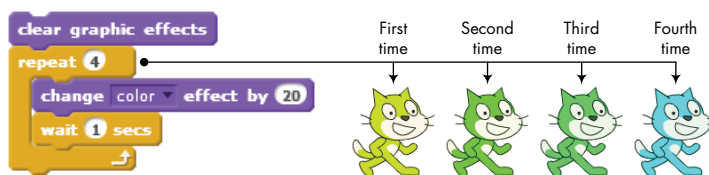


Figure 1-2: Using a script to change the Cat sprite's color

This script waits for one second between color changes, and the four cats you see here show the sprite's new color after each change.

TRY IT OUT 1-1

Though we haven't discussed the blocks in Figure 1-2 yet, read them, look at their shapes, and try to figure out the steps the script took to make the cat teal. (Hint: The first purple block returns the cat to its original color.) What do you think would happen if we removed the **wait** block from the script?

This book covers Scratch 2, which was released in May 2013. This version allows you to create projects directly in your web browser so you don't have to install any software on your computer, and we'll rely on Scratch's web interface for the material in this book.

Now, that you know a little about this language, it's time to kick off our programming journey and learn how to use it.

Scratch Programming Environment

To start Scratch, go to the Scratch website (<http://scratch.mit.edu/>) and click the *TRY IT OUT* link. This should take you to Scratch's project editor interface, shown in Figure 1-3.

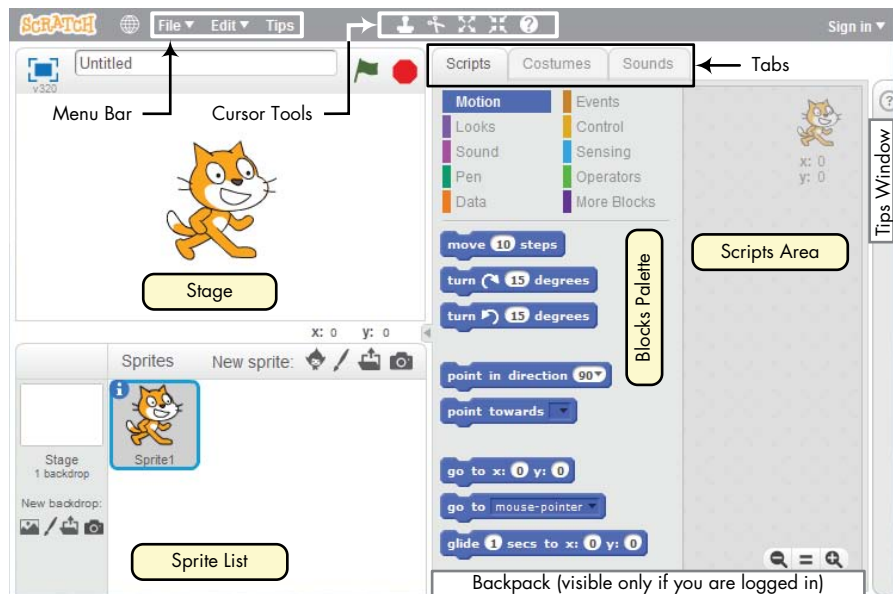


Figure 1-3: The Scratch user interface, where you'll build your programs

You should see a single window with at least the following three panes: the Stage (top left), the Sprite List (bottom left), and the Scripts tab (right), which contains the Blocks tab and the Scripts Area. The right pane also contains two additional tabs, Costumes and Sounds, which will be discussed later in this section. If you're logged into an account on the Scratch website, you should also see the Backpack (bottom right), which has buttons that let you share your project and use sprites and scripts from existing projects.

Let's take a quick look at the three main panes.

The Stage

The *Stage* is where your sprites move, draw, and interact. The Stage is 480 steps wide and 360 steps tall, as illustrated in Figure 1-4. The center of the Stage has an x -coordinate of 0 and a y -coordinate of 0.

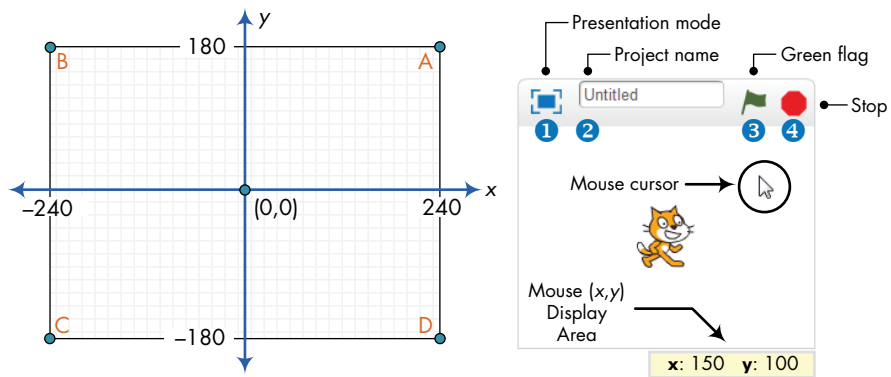



Figure 1-4: The Stage is like a coordinate plane with point $(0,0)$ in the center.

You can find the (x,y) coordinates of any point on the Stage by moving the mouse cursor to that point and watching the numbers in the Mouse (x,y) Display Area, located directly below the Stage.

The small bar located above the Stage has several controls. The Presentation mode icon 1 hides all scripts and programming tools and makes the Stage area take up almost your entire monitor. The edit box 2 shows the name of the current project. The green flag 3 and stop 4 icons let you start and end your program.

TRY IT OUT 1-2

Move the mouse around the Stage and watch the Mouse Display Area. What happens when you move the mouse outside the Stage area? Now, switch to Presentation mode and watch how the screen changes. Click the  icon in the top left of the screen or press ESC on your keyboard to exit Presentation mode.

Sprite List

The Sprite List displays names and thumbnails for all the sprites in your project. New projects begin with a white Stage and a single cat-costumed sprite, as illustrated in Figure 1-5.

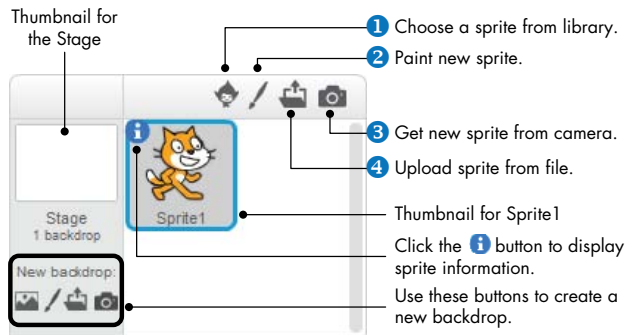


Figure 1-5: The Sprite List for a new project

The buttons above the Sprite List let you add new sprites to your project from one of four places: Scratch’s sprite library ❶, the built-in Paint Editor ❷ (where you can draw your own costume), a camera connected to your computer ❸, or your computer ❹.

TRY IT OUT 1-3

Add new sprites to your project using some of the buttons located above the Sprite List. Rearrange the sprites in the Sprite List by dragging their corresponding thumbnails.

Each sprite in your project has its own scripts, costumes, and sounds. You can select any sprite to see its belongings. Either (1) click the sprite’s thumbnail in the Sprite List or (2) double-click the sprite itself on the Stage. The currently selected sprite thumbnail is always highlighted and outlined with a blue border. When you select a sprite, you can access its scripts, costumes, and sounds by clicking one of the three tabs located above the Scripts Area. We’ll look at the contents of these tabs later. For now, right-click (or CTRL-click if you’re using a Mac) the Cat sprite’s thumbnail to see the pop-up menu shown in Figure 1-6.

The duplicate option ❶ copies the sprite and gives the copy a different name. You can remove a sprite from your project with delete ❷, and you can export a sprite to a `.sprite2` file on your computer using the save to local file option ❸. (To import an exported sprite into another project, just click the Upload sprite from file button shown in Figure 1-5.) The hide/show option ❹ allows you to change whether a sprite on the Stage is visible or not.