

Welcome to Scratch

Chapter 1: Introduction

Downloading scratch:

1. Go to: <https://scratch.mit.edu/scratch2download/>
2. Follow the instructions to download scratch on your Mac, Windows, or Linux machine.

To make an online account and upload projects to the cloud:

1. Go to: <https://scratch.mit.edu/>
2. Click on the “Join Scratch” button in the top right corner of the page. Follow the instructions to create a scratch account.



This is a picture of what a new project screen may look like in scratch.

**1.1 Scratch Layout**



The top of the scratch application contains 4 tabs that are very useful.

File tab: lets you open an existing project, create a new project, save your project, and upload your project to your online account.

Edit tab: lets you undo a “delete” action, move to a smaller screen layout so you can see all of your code easier, or enter “turbo mode,” a mode that runs all of your programs at the fastest rate possible.

Tips tab: opens a tip side-bar with instructions on how to use different tools.

About tab: opens a web browser to the “about” section on scratch’s website.



The right side of scratch shows the program's’ output screen. The results of your code can be seen here. The red stop sign can be used to stop the program at any time. The green flag is used to start the program. The x and y coordinates give the coordinates of the mouse pointer. If the pointer is off the screen, the coordinate grid ends at (-240, 240) for x and (-180, 180) for y.

The blue square in the top left corner makes the screen go full-sized. The cat is just an example sprite.



Below the output screen is the sprite selection screen. The new sprite buttons allow you to create a new sprite from (left to right): a scratch example, the scratch drawing tool, an image file, or your computer’s camera.

The stage (background) tools do the same. Use this part of the scratch layout to select sprites for modification, make new sprites, or select the stage.



The right half of the Scratch screen is devoted entirely to coding. Use the script tabs to add blocks of code. Use the costumes tab to edit the sprites’ costumes. Use the sounds tab to edit the sprites’ sound.

Block Info:

<http://wiki.scratch.mit.edu/wiki/Blocks>

Video Tutorials:

[https://scratch.mit.edu/help/videos/#](https://scratch.mit.edu/help/videos/)

Project examples:

<https://scratch.mit.edu/starter_projects/>

Chapter 2: Motion

**2.1 Working with motion**

1. Drag a “Green Flag” block (from the event script tab) into the program to start
2. Experiment with different motion scripts (from the motion script tab)

**2.2 XY coordinate system**

1. Explain how the scratch coordinate system works (-180, 180 on the y-axis, -240, 240 on the x-axis)
2. Explain positioning and center location (the sprite’s location is a point, rather than an area)
3. Use the “change x” and “change y” blocks to move a sprite
4. right-click the sprite and check its info to see its exact coordinate

**2.3 Control tab**

1. Practice using the repeat and forever loops to make movements continuous. Explain the difference between repeat and forever.
2. Leave the if and then statements for later

**2.4 Lab: Bounce**

1. Start a new project. Name it “Lab 1: Bounce”
2. Draw a new sprite and make it do the following things:
	1. Move forward once
	2. Move for an infinite amount of time
	3. Move infinitely, then turn when sensing a wall
	4. Move in a circle
3. Using the motion blocks, make the sprite “bounce” around the screen (turning whenever it hits a wall)

Chapter 3: Events

**3.1 Working with events**

1. Up until now we have been using the “green flag” block to start the program. Now, try making a sprite move using the “key pressed” block and “sprite clicked” block.
2. Explain the “loudness” and “background switches” trigger blocks.
3. Additionally, sprites and backgrounds can be programmed to generate and react to “messages.” Make a second sprite and use the “broadcast” block to communicate between the two sprites. (i.e when one is clicked, the other one moves forward)

**3.2 Variables**

1. Variables are found under the “data” tab
2. Explain variables, and how they are different from the variables that you solve for in
3. math class
4. Global variables save to the cloud. Only use those for things like
5. Use a variable to vary the speed that a sprite moves at based on what key is pressed
6. You can use the “operator” tab to modify and set variables
7. Operators can also be used on their own
8. Set the move speed of a sprite to a “random” number between 1 and 10
9. Play around with variables, operators, and events

**3.3. Lists**

1. Explain lists
2. Demonstrate how to add and remove items from lists. Give some examples of where this could be useful. <https://scratch.mit.edu/projects/260208/#fullscreen>

**3.4 Control and sensing and operators**

1. Make a sprite that moves forward infinitely
2. Use the if/then blocks in control in combination with the sensing blocks. Use this project to explain the concept of if/then: <https://scratch.mit.edu/projects/97246/>
3. Try using the operators and, or, and not to edit sensing blocks

**3.5 Lab: Quiz board game**

1. Make three sprites - two player sprites and one question sprite.
2. Using the “ask” block under sensing, have the question sprite ask a question.
3. Using if statements and messages, have the two player sprites take turns answering the questions. If a player gets the question correct, he/she moves forward. The first player to reach a certain point wins.

**3.6 Lab: Chase game**

1. Make 3 sprites. One of the sprites is the “player” sprite. This sprite is controlled by the player. One sprite is the “bad guy.” This sprite is programmed to chase the player. One sprite is the “goal.” Touching the “goal” sprite should give the player points. Make variables to keep track of lives, time, and points. Add a message that triggers an ending to the game, either points-based or lives-based.

Chapter 4: Looks, Sound and Pen

**4.1 Costume editor**

1. Select a new sprite. Next to the scripts tab, click costumes.
2. Explain the difference between bitmapped and vector graphics. Use this project: <https://scratch.mit.edu/projects/10568936/>
3. Draw 5 more costumes for your sprite
4. Return to the scripts page. Drag blocks from the “looks” tab to experiment with different “looks” blocks.

**4.2 Sound editor**

1. Select a new sprite. Next to the scripts tab, click sounds. Try editing/adding your own sounds
2. Return to the scripts page. Drag blocks from the “sound” tab to experiment with different “sound” blocks.

**4.3 Pen tab**

1. Go to the “pen” section in the scripts page. Experiment with the different pen scripts.

**4.4 Lab: Pizza maker**

1. Draw a blank pizza outline on the background. Make a sprite, and give it multiple costumes. Each costume should be of a different pizza topping. The sprite should be mobile and controlled by the user. Add code to make the pizza toppings stamp whenever an event occurs. Add another event to make the costume (topping) change to the next costume. Add appropriate sound effects and other finishing touches. The final product should be a customizable pizza maker.

Chapter 5: Clones and More blocks

**5.1 Clones**

1. Give a sprite a set of command blocks
2. Use a clone block to create clones of the same sprite. Use a “when I start as clone” block to give the clones instructions
3. Make sure to add a “delete clone” instruction at the end of the script so your screen doesn’t fill with clones.

**5.2 Definition blocks**

1. Open up the “More blocks” tab
2. This section lets you create your own functions. You can use this in many ways. Here are a few examples: <https://scratch.mit.edu/projects/10006686/> <https://scratch.mit.edu/projects/14736729/> <https://scratch.mit.edu/projects/25835662/>

**5.3 Lab: Asteroids**

1. Create the game asteroids in scratch. Use clones and define blocks in your game.

Chapter 6: Advanced Scratch Concepts

**6.1 Gravity simulation**

 Gravity is a hard physical concept to understand, none the less recreate in Scratch. Look inside this project: <https://scratch.mit.edu/projects/610765/> to understand how a simple loop can create a realistic “gravity” jump effect.

**6.2 Clone tracking**

 Clones are relatively easy to create. However, in order to track and modify individual clones, a list is required. Learn more at the wiki page: <http://wiki.scratch.mit.edu/wiki/Advanced_Clone_Usage>

**6.3 Calculator functions**

 While scratch has many built in mathematical functions, several more must be programmed. Check out this example of a graphing calculator: <https://scratch.mit.edu/projects/2911404/>

**6.4 Side scrolling adventures**

 There are several methods of creating “side-scroller” games. This example uses a define function and costume rotation to make a scolling soccer field: <https://scratch.mit.edu/projects/26083374/>