

■ Scratch Coding Unit

A Complete 4-Week Beginner Curriculum

Grades 3 – 6 | 20 Lessons | Browser-Based (No Install Required)



- ◆ 20 Detailed Lesson Plans
- ◆ 40-Page Student Workbook
- ◆ Coding Concept Reference Cards
- ◆ 15 Debugging Challenge Cards
- ◆ Project Showcase Rubric
- ◆ Teacher Screencast Links

UM Printables

Table of Contents

Unit Overview & Standards Alignment

WEEK 1: Getting Started with Scratch

Lesson 1: Setting Up Accounts & Exploring the Interface

Lesson 2: Sprites & Costumes — Meet Your Characters

Lesson 3: Backdrops & Stages — Setting the Scene

Lesson 4: Motion & Looks — Making Sprites Move

Lesson 5: Week 1 Project — My First Animation

WEEK 2: Loops & Conditionals

Lesson 6: Sequences — Ordering Your Steps

Lesson 7: Forever Loops — Never-Ending Fun

Lesson 8: Repeat Loops — Doing Things a Set Number of Times

Lesson 9: If/Then — Making Decisions

Lesson 10: Week 2 Project — Interactive Story Scene

WEEK 3: Animation & Interaction

Lesson 11: Broadcasting Messages — Sprites Talking to Each Other

Lesson 12: Variables — Keeping Score & Tracking Data

Lesson 13: Sensing — Detecting Touch, Color & Distance

Lesson 14: Sound & Music — Adding Audio

Lesson 15: Week 3 Project — Animated Short Film

WEEK 4: Final Game Project

Lesson 16: Game Design Planning

Lesson 17: Building Game Mechanics

Lesson 18: Adding Levels & Challenges

Lesson 19: Polishing & Debugging Your Game

Lesson 20: Project Showcase & Celebration

Student Workbook (40 pages)

Coding Concept Reference Cards

Debugging Challenge Cards (15)

Project Showcase Rubric

Teacher Screencasts Link List

Unit Overview

This **4-week Scratch Coding Unit** introduces students in Grades 3–6 to the fundamentals of computer programming through the Scratch visual coding platform. All lessons are **100% browser-based** — no software installation is required. Students progress from exploring the Scratch interface to creating their own fully functional game by the end of the unit.

Component	Details
Grade Level	Grades 3–6 (ages 8–12)
Duration	4 weeks, 5 lessons per week (45 min each)
Total Lessons	20 structured lesson plans
Platform	Scratch (scratch.mit.edu) — browser-based, no install
Student Workbook	40-page step-by-step workbook with guided activities
Supplementary Materials	Concept reference cards, 15 debugging challenges, rubric, screencast links
Prior Experience	None required — designed for absolute beginners
Devices Needed	Laptops, Chromebooks, or desktops with internet + browser

Week-at-a-Glance

Week	Theme	Key Concepts	Project
1	Getting Started	Interface, Sprites, Costumes, Backdrops, Motion, Looks	My First Animation
2	Loops & Conditionals	Sequences, Forever, Repeat, If/Then, Operators	Interactive Story Scene
3	Animation & Interaction	Broadcasting, Variables, Sensing, Sound	Animated Short Film
4	Final Game Project	Game Design, Mechanics, Levels, Debugging, Polish	Original Game

Standards Alignment

This unit aligns with the following national standards:

- **CSTA K–2:** 1A-CS-01, 1A-AP-08, 1A-AP-09, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14
- **CSTA 3–5:** 1B-CS-01, 1B-CS-02, 1B-AP-08, 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-13, 1B-AP-15, 1B-AP-17
- **ISTE:** 1.1 Empowered Learner, 1.4 Innovative Designer, 1.5 Computational Thinker, 1.6 Creative Communicator

- **CCSS-ELA:** W.3-6.2 (Informative Writing), SL.3-6.1 (Collaborative Discussions), SL.3-6.5 (Multimedia Presentations)
- **CCSS-Math Practice:** MP1 (Problem Solving), MP2 (Abstract Reasoning), MP6 (Precision), MP7 (Structure)

Materials Checklist

- Computers with internet access (1 per student or pair)
- Modern web browser (Chrome, Firefox, Edge, or Safari)
- Student Scratch accounts (set up in Lesson 1)
- Printed student workbooks (1 per student)
- Printed coding concept reference cards (1 set per student or pair)
- Printed debugging challenge cards (1 set per small group)
- Teacher projector / interactive whiteboard for demonstrations
- Headphones (optional, for Lessons 14 and 20)
- Project showcase rubric (1 per student, for Lesson 20)

WEEK 1: GETTING STARTED WITH SCRATCH

Explore the Scratch interface, create sprites, design backdrops, and build your first animation!

This week, students set up their Scratch accounts and become familiar with the coding environment. They learn to add and customize sprites, create backdrops, use motion and looks blocks, and combine these skills to produce their very first animation project.

■ Teacher Note

Before Week 1 begins, ensure all computers have internet access and a modern browser. Consider creating a class Scratch account or having IT set up individual student accounts in advance. Scratch requires users to be 13+ or have parent/teacher consent — check your school's policy.

Lesson 1: Setting Up Accounts & Exploring the Interface

■ Duration: 45 min

Learning Objectives

- ✓ Create and log into a Scratch account
- ✓ Identify the five main areas of the Scratch editor (Stage, Sprite Pane, Block Palette, Scripts Area, Menu Bar)
- ✓ Open, close, and switch between projects
- ✓ Explain what Scratch is and why it is used for coding

Materials Needed

- Computers with internet access
- Projector/whiteboard for demo
- Student Workbook pp. 1–4

Lesson Procedure

Introduction (10 min)

Ask students: *"Have you ever played a video game or used an app and wondered how it works?"* Explain that today they will begin learning to code using Scratch — a free, visual programming platform created at MIT where they can create their own games, animations, and stories by snapping code blocks together like puzzle pieces.

Key talking points:

- Scratch is a **visual** programming language — you drag and drop blocks; no typing code
- Over 100 million projects have been created on Scratch worldwide

- It runs entirely in your web browser — nothing to install!

Account Setup (10 min)

1. Open a browser and navigate to **scratch.mit.edu**
2. Click the **"Join Scratch"** button in the top-right corner
3. Choose a username (no real names — e.g., **CoolCoder123**) and a secure password
4. Provide birth month/year and country (use school email if available)
5. Confirm the account via email if required
6. Click **"Create"** to start a new project

■ Tip

If setting up accounts takes too long, have students use the Scratch editor without saving (they can create accounts later). Alternatively, create a shared class account in advance.

Explore the Interface (15 min)

Project the Scratch editor on the board. Walk students through each area using the **5-Zone Scavenger Hunt**:

Zone	Location	What It Does
Stage	Top-right	Shows your project running — the “stage” where things happen
Sprite Pane	Bottom-right	Lists all sprites (characters) in your project
Block Palette	Left side	Contains all coding blocks organized by category (color-coded)
Scripts Area	Center	Where you drag and snap blocks together to create code
Menu Bar	Very top	File menu, tutorials, project name, share button

Give students 5 minutes to freely explore. Encourage them to click on different block categories (Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables) and notice the color coding.

Closure (10 min)

Have students close their laptops. Ask these review questions:

- Which zone do you drag blocks into to write code? (*Scripts Area*)
- Which zone shows your project running? (*Stage*)
- What color are the Motion blocks? (*Blue*)
- What color are the Control blocks? (*Orange/Gold*)

Assign **Student Workbook pp. 1–4** for homework or in-class completion: labeling the Scratch interface zones and reflecting on what they are most excited to create.

Assessment

- **Formative:** Observe students during the Scavenger Hunt — can they locate each zone?
- **Exit Ticket:** Students write one thing they learned and one question they have on a sticky note.

Lesson 2: Sprites & Costumes — Meet Your Characters

■ Duration: 45 min

Learning Objectives

- ✓ Define what a sprite is in Scratch
- ✓ Add a new sprite from the library and delete the default Cat sprite
- ✓ Explain the difference between a sprite and a costume
- ✓ Switch between costumes to create a simple animation effect

Materials Needed

- Computers with Scratch accounts • Projector for demo • Student Workbook pp. 5–8

Lesson Procedure

Introduction (8 min)

Show students a Scratch project with multiple sprites. Explain: *"In Scratch, every character, object, or thing that appears on the Stage is called a **sprite**. The Scratch Cat is our default sprite, but we can add hundreds of others — animals, people, vehicles, and more!"*

Explain the concept of **costumes**: *"A costume is like an outfit for your sprite. Each sprite can have multiple costumes, and when you switch between them quickly, it looks like the sprite is moving — this is how animation works!"*

Guided Practice: Adding & Deleting Sprites (12 min)

1. Log in to Scratch and click **"Create"** to start a new project
2. Notice the Scratch Cat sprite already on the Stage
3. To **delete** the Cat: right-click (or long-press) the Cat in the Sprite Pane → select **"Delete"**
4. To **add a new sprite**: click the **"Choose a Sprite"** icon (cat with + sign) at the bottom-right
5. Browse the sprite library by category (Animals, People, Fantasy, etc.)
6. Select a sprite (e.g., **Pico** from the People category) and click to add it
7. Repeat to add a second sprite (e.g., **Butterfly1**)
8. Practice deleting a sprite and adding a different one

Guided Practice: Exploring Costumes (12 min)

1. Click on your sprite in the Sprite Pane to select it
2. Click the **"Costumes"** tab at the top of the Scripts Area
3. If the Cat is still present, notice it has two costumes: "cat-a" and "cat-b"
4. Switch between costumes by clicking each one — observe how the sprite changes
5. Add a sprite with many costumes (e.g., **Dani** or **Nano**) and explore all costumes

6. Click the **"Paint"** icon to create your own costume using the drawing tools
7. Try renaming a costume by clicking the name and typing a new one

■ **Tip**

Students love the drawing editor! Allow a few extra minutes for them to create a custom costume. This builds ownership and excitement for their projects.

Independent Practice (8 min)

■ **CHALLENGE:** Create a project with at least 3 different sprites. Give one of them a custom costume you drew yourself!

Closure (5 min)

- What is the difference between a sprite and a costume? (*A sprite is a character on the Stage; a costume is one of its appearances/outfits*)
- How many sprites can a Scratch project have? (*As many as you want!*)
- What are two ways to add a sprite? (*From the library, or paint your own*)

Assign **Student Workbook pp. 5–8**.

Lesson 3: Backdrops & Stages — Setting the Scene

■ Duration: 45 min

Learning Objectives

- ✓ Define what a backdrop is and how it differs from a sprite
- ✓ Add and switch between multiple backdrops from the library
- ✓ Paint a custom backdrop using the drawing editor
- ✓ Use the "switch backdrop to" block to change the scene in code

Materials Needed

- Computers with Scratch accounts
- Projector for demo
- Student Workbook pp. 9–12

Lesson Procedure

Introduction (8 min)

Explain: *"If sprites are the actors in a play, then the **backdrop** is the stage scenery — the background behind everything. A good backdrop sets the mood and tells the audience where the story takes place."*

Show examples: a forest backdrop for a nature story, a space backdrop for a space game, a room backdrop for a conversation.

Key distinction: **Sprites move; backdrops stay still.** A backdrop is the entire background of the Stage, while a sprite is a character or object placed on top of it.

Guided Practice: Adding Backdrops (12 min)

1. Open a new Scratch project
2. Look at the bottom-right area — find the **"Choose a Backdrop"** icon (mountain image with +)
3. Click it and browse the backdrop library by category
4. Select **"Forest"** from the Nature category
5. Notice the Stage thumbnail in the Sprite Pane now shows the forest backdrop
6. Add a second backdrop: **"Space"** from the Space category
7. Click between the two backdrops in the Stage's **Backdrops** tab to switch them

Guided Practice: Paint Your Own Backdrop (10 min)

1. Click the **"Paint"** icon next to "Choose a Backdrop"
2. Use the **Rectangle** tool to draw a green ground at the bottom
3. Use the **Circle** tool to draw a yellow sun in the top-right corner
4. Use the **Line** tool to add some grass or tree trunks
5. Use the **Fill** tool to color in shapes
6. Rename your backdrop to something descriptive like **"My Park"**

Guided Practice: Switching Backdrops with Code (10 min)

Now let's make the backdrop change when we click the green flag:

1. Select the **Stage** (click the Stage thumbnail in the Sprite Pane)
2. Go to the **Scripts** tab (make sure you are coding the Stage, not a sprite!)
3. Drag an **Events** → **When green flag clicked** block
4. Drag a **Looks** → **Switch backdrop to** block and snap it underneath
5. Select "Forest" from the dropdown
6. Add a **Control** → **Wait 2 seconds** block
7. Add another **Looks** → **Switch backdrop to** block — select "Space"
8. Click the green flag and watch the backdrop change!

■ Teacher Note

A common student mistake is adding backdrop-switching code to a sprite instead of the Stage. Remind them: **Backdrop code goes on the Stage**, not on individual sprites. The Stage has its own scripts tab — click the Stage thumbnail first!

Closure (5 min)

- What is the difference between a sprite and a backdrop? (*Sprites move; backdrops are the background*)
- Where does backdrop-switching code go? (*On the Stage, not on a sprite*)
- How do you create your own backdrop? (*Click the Paint icon and use the drawing tools*)

Assign **Student Workbook pp. 9–12**.

STUDENT WORKBOOK

Step-by-Step Activities, Planning Templates & Reflections

■ My Scratch Coding Workbook

A 4-Week Journey into Coding

Name: _____

Class / Period: _____

Date Started: _____

The Scratch Interface

Lesson 1: Getting Started

Scavenger Hunt: Label the 5 zones of the Scratch Editor using the word bank.

Word Bank: Block Palette | Scripts Area | Stage | Sprite Pane | Menu Bar

1. _____

(Top-right: where your project plays)

2. _____

(Left side: where you find all the coding blocks)

3. _____

(Center: where you drag and snap blocks)

4. _____

(Bottom-right: shows all your characters)

5. _____

(Very top: file, save, and share buttons)

Block Category Colors: Match the category to its color.

Category	Color
Motion	_____ (Hint: Like the ocean)
Looks	_____ (Hint: Like a grape)
Sound	_____ (Hint: Like a flamingo)
Events	_____ (Hint: Like the sun)
Control	_____ (Hint: Like an orange)

My Scratch Account

Lesson 1: Getting Started

Record your Scratch login information here so you don't forget it!

My Username: _____

My Password Hint: _____

Reflection: What are you most excited to create in Scratch? Why?

Sprites & Costumes

Lesson 2: Meet Your Characters

Key Vocabulary:

Word	Definition	In My Own Words
Sprite	A character or object on the Stage	
Costume	A specific appearance or outfit for a sprite	

Steps to Add a Sprite: Put these steps in the correct order (1-4).

- _____ Decide what kind of character your project needs.
- _____ Click the "Choose a Sprite" icon at the bottom-right.
- _____ Browse the categories (Animals, People, Fantasy, etc.).
- _____ Select a sprite from the library and click to add it.

True or False?

- A sprite can only have one costume. (True / False)
- You can draw your own sprite using the Paint editor. (True / False)
- The Scratch Cat cannot be deleted. (True / False)

■ Conditionals

What is it?

A decision in your code. It checks if something is true, and if it is, it runs the code inside.

If / Then:

Runs code ONLY if the condition is true. If false, it skips the code.

If / Then / Else:

Runs one block of code if true, and a different block of code if false.

Tip:

Put your If/Then blocks inside a **forever** loop so the computer keeps checking!

■ Variables

What is it?

A labeled box that holds information (like a number or word) that can change while the program runs.

Set to:

Replaces the value. Example: Set Score to 0.

Change by:

Adds to or subtracts from the value. Example: Change Score by 1.

Pro Tip:

Always reset your variables at the start using **When green flag clicked!**

■ Events

What is it?

Blocks that tell the computer WHEN to run your code. All scripts must start with an Event block (a hat block).

Common Events:

When green flag clicked: Runs when you press Go.

When key pressed: Runs when you press a keyboard key.

When this sprite clicked: Runs when you click the sprite.

When I receive: Runs when another sprite sends a broadcast.

■ Motion

What is it?

Blocks that control how a sprite moves around the Stage using the coordinate system (x and y).

Key Blocks:

Move (10) steps: Walks forward in the direction the sprite faces.

Change x/y by: Slides the sprite without turning it (best for keyboard controls!).

Glide: Smoothly slides to a position over time.

Go to: Instantly teleports to a position.

■ Looks

What is it?

Blocks that change how a sprite looks or sounds, including speech bubbles, size, and special effects.

Key Blocks:

Say / Think: Shows a speech or thought bubble.

Show / Hide: Makes sprite visible or invisible.

Next costume: Switches to the next outfit (great for animation!).

Change size / Set effect: Modifies scale and visual filters.

■ Sensing

What is it?

Blocks that let your sprite "feel" its surroundings. They check for touches, colors, mouse clicks, and distances.

Key Blocks:

Touching (sprite)? Is it overlapping another sprite?

Touching (color)? Is it standing on a specific color?

Key pressed? Is the user pressing a key?

Mouse down? Is the mouse button clicked?

■ Broadcasting

What is it?

A way for sprites to send messages to each other. One sprite broadcasts (shouts) a message, and any sprite listening can react.

How to use it:

1. Sprite A uses **Broadcast (message)**.
2. Sprite B uses **When I receive (message)** to start its code.
3. Great for scene changes, conversations, and triggering Game Over screens!

■ Pen Extension

What is it?

An extension that lets your sprite draw lines on the Stage as it moves, like a pen on paper.

How to use it:

1. Click **Add Extension** and select **Pen**.
2. Use **Pen Down** to start drawing.
3. Move the sprite to draw a line.
4. Use **Pen Up** to stop drawing.
5. Use **Erase All** to clear the screen.

Debugging Challenge Cards

15 common bugs and how to fix them! Use these as desk references or early-finisher challenges.

Debug Card 1

Challenge: The Frozen Sprite

■ **The Bug:** I attached a **move (10) steps** block, but when I click the green flag, my sprite doesn't move at all!

■ **The Fix:** You are missing an **Event** hat block! Code only runs if it starts with an event. Add a **When green flag clicked** block to the very top of your script.

Debug Card 2

Challenge: The Instant Teleport

■ **The Bug:** I want my sprite to glide smoothly across the screen, but it instantly jumps to the other side instead!

■ **The Fix:** You probably used the **go to x: y:** block, which is instant. Replace it with the **glide (1) secs to x: y:** block to see a smooth animation.

Debug Card 3

Challenge: The Infinite Score Glitch

■ **The Bug:** When my sprite touches the apple, the score goes up by hundreds of points in one second!

■ **The Fix:** The sprite stays touching the apple for several frames, so the score keeps going up. Fix this by adding a **wait (1) seconds** block after changing the score, or by moving the apple to a random position immediately.

Debug Card 4

Challenge: The Broken Restart

■ **The Bug:** When I click the green flag to play again, my score and lives aren't resetting! It keeps my old score from the last game.

■ **The Fix:** You need to reset your variables at the start of the game. Under your **When green flag clicked** block, add **set Score to (0)** and **set Lives to (3)**.

Project Showcase Rubric

Use this rubric to assess the final game projects. Share with students before they begin Week 4 so they know the expectations!

Criteria	4 – Exceeds	3 – Meets	2 – Approaching	1 – Below
Code Functionality	Game runs without any errors. All mechanics (movement, scoring, lives) work perfectly. Win/lose conditions trigger correctly.	Game runs with minor bugs that do not disrupt gameplay. Most mechanics work as intended.	Game has noticeable bugs that disrupt gameplay. Some mechanics do not work as intended.	Game does not run properly or crashes. Major mechanics are missing or broken.
Coding Concepts (Loops, Conditionals, Variables)	Uses loops, conditionals, and variables effectively and creatively. Code is efficient (e.g., uses loops instead of repeating blocks).	Uses loops, conditionals, and variables correctly to create a functioning game.	Uses some coding concepts, but misses others (e.g., uses loops but no conditionals) or uses them incorrectly.	Rarely or never uses the coding concepts taught in the unit.
Creativity & Design	Game has a unique theme or creative twist. Custom sprites/backdrops are used. Highly engaging.	Game has a clear theme. Mix of library and custom assets. Fun to play.	Game uses only default sprites/backdrops. Theme is unclear or unoriginal.	Little effort put into design. Game is a blank stage with a single sprite.
User Experience (Title, Instructions, Difficulty)	Has a title screen, clear instructions, and smooth difficulty progression. Very easy to understand how to play.	Has instructions and a way to start. Gameplay is understandable.	Missing instructions or a title screen. Hard to figure out how to play without help.	No instructions or title screen. Game is confusing to play.
Presentation & Explanation	Clearly explains how the code works using correct vocabulary. Answers questions confidently.	Explains the basic logic of the game. Uses some coding vocabulary.	Struggles to explain how the game works. Rarely uses coding vocabulary.	Cannot explain how the game works or refuses to present.

Total Score	_____ / 20
Grade	4 = A, 3 = B, 2 = C, 1 = D

Teacher Comments:

Teacher Screencasts & Video Resources

Below is a curated list of recommended screencast topics and video resources to support the instruction of this unit. Teachers can use these to flip the classroom, provide absent students with make-up material, or as a visual guide for their own instruction. Record your own or link to trusted Scratch tutorials!

Lesson	Video Topic / Description	Link / URL
1	Setting up a Scratch Account & Interface Tour	https://scratch.mit.edu/ideas
2	Adding & Deleting Sprites; Exploring Costumes	[Insert Link Here]
3	Adding Backdrops & Using the Paint Editor	[Insert Link Here]
4	Understanding the Coordinate System (x & y)	[Insert Link Here]
5	Building Your First Animation (Green Flag to End)	[Insert Link Here]
6	What is a Sequence? Ordering Your Blocks	[Insert Link Here]
7	The Forever Loop: Continuous Movement & Animation	[Insert Link Here]
8	The Repeat Loop & Pen Extension: Drawing Shapes	[Insert Link Here]
9	If/Then Conditionals: Making Sprites React	[Insert Link Here]
10	Putting it Together: Keyboard Controls & Interactive Stories	[Insert Link Here]
11	Broadcasting Messages: Sprites Communicating	[Insert Link Here]
12	Variables 101: Creating Scoreboards & Counters	[Insert Link Here]
13	Sensing Blocks: Touching Colors & Mouse Input	[Insert Link Here]
14	Adding Sound Effects & Music to Your Project	[Insert Link Here]
15	Creating a Multi-Scene Animated Film	[Insert Link Here]
16	Planning Your Game: The Game Design Document	[Insert Link Here]
17	Coding Core Game Mechanics (Movement & Collecting)	[Insert Link Here]
18	Adding Lives, Win/Lose Conditions & Difficulty	[Insert Link Here]

19	Debugging 101: Finding and Fixing 5 Common Bugs	[Insert Link Here]
20	Adding a Title Screen & Polishing Your Game	[Insert Link Here]

■ Tip

To create your own screencasts, use free tools like Loom, Screencastify, or OBS Studio. Record your screen while explaining the steps, then paste the links in the table above before printing or sharing this guide with students!

■ Thank You!

Thank you so much for purchasing and using the **Scratch Coding Unit: Beginners (Gr 3–6)**! I poured my heart into creating a resource that is rigorous, engaging, and easy to implement in your classroom.

Your feedback is incredibly valuable!

If you found this resource helpful, I would be so grateful if you would take a minute to leave a **fair review on Teachers Pay Teachers**. Your reviews help other teachers find quality resources and allow me to continue creating premium materials for your classroom!

How to Leave a Review:

1. Go to your **"My Purchases"** page on TPT.
2. Find this resource and click **"Leave a Review"**.
3. Share how you used it in your classroom and earn TPT credits!

Happy Coding!

— UM Printables