

From ScratchJr to Scratch: a taster

Lesson at a glance

Frame Scratch as the word-block step up from ScratchJr, then play the short intro video on the IWB. Guide pairs together to scratch.mit.edu → Create, model green-flag + move, then turn, wrap both in forever, and tune the numbers so the cat steers toward a stage goal. Add a backdrop for purpose, free-explore briefly if time, and close with display-only Code talk: one bug fixed, one thing same as ScratchJr, one thing new.

Learning objectives

- Make a Scratch sprite move to a goal with a short block sequence
- Say one thing that is the same as ScratchJr and one thing that is new

Before the bell – prep

Check school devices open scratch.mit.edu and Create without blocked pop-ups. Decide the no-account finish routine (leave open, screenshot, or teacher save) and say it early. Have the Scratch intro video ready on the IWB; bookmark Create yourself for modelling.

Materials

Item	Qty	Per	Source	Low-cost substitute
tablet or laptop with Scratch	1	pair	school devices	run the lesson on the IWB with pupils taking turns

Safety watch-point

Use school-approved browsers and paths only. Skip pupil accounts if that is school policy; set the save/screenshot routine up front and keep screens on task.



Teaching moves

- **What we're building:** Settle devices face-down and eyes on the IWB. State the clear goal: a short block sequence that moves the cat toward something on the stage. Ask what they already know from ScratchJr so comparison language is ready before anyone logs on.
- **Introducing Scratch:** Before the video, cue pupils to watch how sprites move. Afterward harvest one movement idea they noticed, then link: we will make our cat move with blocks the same way — plan, stack, run.
- **Open the Scratch website:** Walk the class in lockstep: browser → scratch.mit.edu → Create. On the IWB show closing any tutorial pop-up. Circulate until every pair is on a fresh editor; redirect anyone stuck on Explore/home to the Create link at the top.
- **Make the cat move:** Model orange Control → when green flag clicked, then blue Motion → move (10) steps snapped under it. Run with the green flag and name a goal (e.g. toward the side of the stage). Watch for blocks left floating unconnected; struggling pairs only need flag + move working before you move on.
- **Turn the Cat:** Model attaching turn cw (15) degrees under move so one green-flag click does both. Ask which way the cat faced and whether it is closer to the goal. Common bug: turn sitting beside the stack — cue snap-under and several runs described in plain words.
- **Add a loop:** Show the forever block's C-shape on the IWB and ask what the gap is for before anyone uses it. Demonstrate wrapping move and turn inside so both repeat. Show the stop sign; watch for empty forever blocks or panic-clicking to halt.
- **Change the numbers:** Name the numbers as parameters programmers tune. Invite pairs to try larger steps and a different turn aimed at a stage feature. If the sprite flies off-stage, revoice smaller steps; support pairs change only one number at a time.
- **Add a backdrop:** Point to Choose a Backdrop at the bottom-right of the stage. Let pairs pick a scene that gives the path a purpose. Remind them the backdrop does not replace the code — they may need to retune move/turn after the scene changes.
- **Explore Scratch:** If time remains, allow brief free explore under the rule change-run-notice: one new block or a second sprite only. Use an early-stop signal so Code talk still gets protected time.
- **Code talk — what worked and what we'd change:** Display-only talk — nothing typed. Surface one real bug a pair hit (unconnected blocks, empty forever, sprite off-stage) and how they fixed it. Prompt one thing like ScratchJr, one thing new, then a class echo that debugging is what programmers do.

What it should show

A working stack is green flag → move and turn inside forever, so the cat walks a repeating curved path the pair can steer toward a stage goal by tuning step and degree values. Surprising results are usually unconnected blocks, an empty forever, or steps so large the sprite leaves the stage — re-snap, show stop, then try a smaller number.



Misconceptions & interventions

- **“I’m on Scratch but nothing happens when I press the green flag.”** — Check they clicked Create (not Explore) and that blocks are snapped under the green-flag hat, not floating loose. Have them drag the stack until it clicks, then run again.
- **“The forever block just sits there — or I can’t stop the cat.”** — Show the C-shaped gap must wrap the move and turn blocks inside it. Point to the red stop sign beside the green flag so they halt on purpose, not by mashing the keyboard.
- **“Bigger numbers are always better.”** — When the cat shoots off-stage, ask them to halve the steps and run again. Link the number to aiming at the goal, not to speed for its own sake.

Differentiation

Emerging	Developing	Proficient
• Stay with flag + move only until that runs reliably, then add turn with the teacher table before forever. • Change only the move number; leave turn at 15 until the path is steady.	• Tune both step and degree values to reach a chosen spot on the backdrop, then describe the path in plain words. • If exploring, try one new Motion block, run, and say what changed.	• Challenge a more precise route to a backdrop landmark; critique whether the test of each number change was fair (one change, then run). • In Code talk, name a concrete debug fix and one Scratch feature they would use in a rebuild.

Cross-curricular hook

Link turn degrees and step sizes to Maths angle and length — changing a number is measuring a move on screen.