**Scratch and Scratch Jr Lesson Plans**

<https://code.org/educate/curriculum/teacher-led>

As of Summer 202 this link is not active but all the individual links to the lessons below are still active



## Elementary School

[Scratch Animate Your Name](https://scratch.mit.edu/scratchr2/static/pdfs/help/AnimateYourNameGuide.pdf)
**Ages 8 to 16. Programming, Creativity, Web-based.** Students will animate the letters of their name, initials, or favorite word using Scratch!

[Scratch Hide and Seek Game](https://scratch.mit.edu/scratchr2/static/pdfs/help/Hide-and-Seek-Guide.pdf)
**Ages 8 to 16. Programming, Creativity, Web-based.** Students will gain experience with coding as they make a hide-and-seek game.

[Scratch Dance, Dance, Dance](https://scratch.mit.edu/scratchr2/static/pdfs/help/DanceGuide.pdf)
**Ages 8 to 16. Programming, Creativity, Web-based.** Participants will create and code an animated dance scene.

[ScratchJr: Can I Make a Spooky Forest?](https://drive.google.com/file/d/0B_kqMacZctDuWDRtLXQwRWp4ZGc/view)
**Ages 5-9. Art, Storytelling, Sequence, Unplugged.** Students will learn more features of ScratchJr by creating a spooky forest with multiple characters!

[ScratchJr: Can I Make My Characters Greet Each Other?](https://drive.google.com/file/d/0B_kqMacZctDuNVg0QTJaWUR6WVE/view)
**Ages 5-9. Art, Storytelling, Events, Unplugged.** Students will learn advanced features of the ScratchJr app when they make a dog and kitten meet each other and exchange hellos!

[ScratchJr: Can I Make the Sun Set?](https://drive.google.com/file/d/0B_kqMacZctDuTG5CdEhoR2o3M0E/view?usp=sharing)
**Ages 5-9. Programming, Storytelling, Mathematics, Creativity, Unplugged.** Students will get an introduction to programming by making a sun set over a city landscape using ScratchJr!

[My Robotic Friend](https://csedweek.org/unplugged/thinkersmith)
**Ages 5-18. Sequence, Algorithms, Programming, Unplugged.** Students use paper and pencils to create programs to teach their "Robotic Friends" how to stack plastic cups into a specific design.

[Conditionals with Cards](https://code.org/files/ConditionalsHoC.pdf)
**Ages 8-12. Sequence, Algorithms, Conditionals, Unplugged.** Use playing cards (or dice, or sheets of paper...anything with the ability to randomize) to create a program on paper for the class to follow. This program will utilize the randomization, along with "if" and "else" statements, to add uncertainty to your game's outcome.

[Binary Baubles](https://code.org/files/CSEDbinary.pdf)
**Ages 8-18. Sequence, Algorithms, Programming, Unplugged.** Make fun take-home items that "store" students' initials using only two colors.

[Bomberbot Hour of Code Activities](http://www.bomberbot.com/hourofcode)
**Ages 8-12. Sequence, Algorithms, Programming, Unplugged or Web-based.** Students will learn a series of programming concepts and apply them to programming a robot.

[STEM Projects](https://www.tynker.com/hour-of-code/tynker-stem-teacher-guide.pdf)
**Ages 5-18. Programming, Science (Ecology), Science (Space), Web-based.** Code and animate a Solar System simulation, an interactive ecological pyramid, a working analog clock, and more.

[Rock, Paper, Scissors](https://code.org/files/PGUTSRockPaperScissors.pdf)
**Ages 10-15. Science (Biology), Unplugged.** This activity builds off of the classic game of Rock/Paper/Scissors, known to most students, and relates it to a phenomenon seen in biology.

[Kodable fuzzFamily Frenzy](https://www.kodable.com/hour-of-code/lessons)
**Ages 5-9. Seqence, Algorithms, Programming, Unplugged.** Students will learn basic programing language and use math concepts to move a “robot” forward, spin, and jump.

[Google CS First](http://cs-first.com/hocteachers)
**Ages 9-14 Sequence, Programming, Art, Storytelling, Web-based.** Students use Scratch to code a story about being lost at sea.

[Bitsbox Coding + Math](https://bitsbox.com/hocprint.html)
**Ages 5-13. Programming, Math (General, Geometry), Web-based.**

[Bitsbox Coding + Science](https://bitsbox.com/hocprint.html)
**Ages 5-9. Programming, Science (General, Biology, Chemistry, Physics), Web-based.**

[Bitsbox Coding + Art](https://bitsbox.com/hocprint.html)
**Ages 5-9, Programming. Language Arts, Visual Arts, Music, Web-based.**

[Flocabulary On One Condition](https://www.flocabulary.com/lesson-plan-coding-conditionals/#introlesson)
**Ages 7-13. Programming, Language arts, Storytelling, Web-based.** Students will learn how to write conditional statements and complete an activity sequence where they generate their own conditionals, evaluate given conditionals to determine the outcome of a program.

## Middle School

[Scratch Animate Your Name](https://scratch.mit.edu/scratchr2/static/pdfs/help/AnimateYourNameGuide.pdf)
**Ages 8 to 16. Programming, Creativity, Web-based.** Students will animate the letters of their name, initials, or favorite word using Scratch!

[Scratch Hide and Seek Game](https://scratch.mit.edu/scratchr2/static/pdfs/help/Hide-and-Seek-Guide.pdf)
**Ages 8 to 16. Programming, Creativity, Web-based.** Students will gain experience with coding as they make a hide-and-seek game.

[Scratch Dance, Dance, Dance](https://scratch.mit.edu/scratchr2/static/pdfs/help/DanceGuide.pdf)
**Ages 8 to 16. Programming, Creativity, Web-based.** Participants will create and code an animated dance scene.

[STEM Projects](https://www.tynker.com/hour-of-code/tynker-stem-teacher-guide.pdf)
**Ages 5-18. Programming, Science (Ecology), Science (Space), Web-based.** Code and animate a Solar System simulation, an interactive ecological pyramid, a working analog clock, and more.

[Bomberbot Hour of Code Activities](http://www.bomberbot.com/hourofcode)
**Ages 8-12. Sequence, Algorithms, Programming, Unplugged or Web-based.** Students will learn a series of programming concepts and apply them to programming a robot.

[Input and Output, Math Activity](https://app.vidcode.io/doc/unplugged-activity-math.pdf)
**Ages 12-16. Math (Algebra), Math (Functions), Unplugged.** Connect JavaScript functions to both math and real world problems.

[Climate Science](https://app.vidcode.io/hourofcode/science-teacher-guide)
**Ages 12-16. Science (Climate), Science (Environment), Unplugged.** Students draw a picture, and take turns giving the class steps to recreate their drawing.

[Google CS First](http://cs-first.com/hocteachers)
**Ages 9-14 Sequence, Programming, Art, Storytelling, Web-based.** Students use Scratch to code a story about being lost at sea.

[Arduino Activity](https://csedweek.org/csteacher/arduino.pdf)
**Ages 12-15. Programming, Arduino** Ever wonder how toys make noises and blink lights when you push buttons? Microcontrollers and circuits are used in all sorts of everyday objects. From remote controlled cars to robots and drones.

[Secret Codes Activity](https://csedweek.org/csteacher/secretcodes.pdf)
**Ages 12-15. Ciphers, Math (Cryptography), Scratch, Web-based.** Turing has done many things for computer science (often called the father of computer science) but today we will focus on one very important one that helped with the invention of computers.

[Computer History Activity](https://csedweek.org/csteacher/computerhistory.pdf)
**Ages 12-15. History, Storytelling, Scratch, Google Docs, Unplugged** Your class will be be creating a 'history of computers' web page/Scratch project/video that we can share with the world. To make this web page, you and your partner will do research and write about one important event or person in computer history

[Grace Hopper Debugging Activity](https://csedweek.org/csteacher/gracehopperdebugging.pdf)
**Ages 12-18. History, Language Arts, Storytelling, Unplugged** Students will research Grace Hopper and learn the story of the first "bug".

[Flocabulary On One Condition](https://www.flocabulary.com/lesson-plan-coding-conditionals/#introlesson)
**Ages 7-13. Programming, Language arts, Storytelling, Web-based.** Students will learn how to write conditional statements and complete an activity sequence where they generate their own conditionals, evaluate given conditionals to determine the outcome of a program.