


Computational Thinking...

- you are already doing it!
- problem solving...taking a big idea and breaking it down.
- IS NOT CODING...coding is a way to demonstrate computational thinking.
- **Programming (CODING): taking a solution and breaking it down into steps so simple that even a computer can do it!**

Let's do an Hour of Code Activities...



Hour of Code Activities

Try a one-hour tutorial designed for all ages in over 45 languages. Join millions of students and teachers in over 180 countries starting with an Hour of Code.

Want to keep learning? [Go beyond an hour](#)

Teachers: [Host an hour](#) or [read the How-To Guide](#)

All grades | Pre-reader | Grades 2-5 | **Grades 6-8** | Grades 9+ | **Beginner** | Comfortable

Sort by: Recommended

Created by: All

Classroom technology:

- Computers
- Android
- iPad/iPhone
- Poor or no internet
- No computers or devices

MINECRAFT Hour of Code
Grades 2+ | Blocks

Create your own Google logo
Grades 2+ | Blocks, Scratch

Mobile CS Principles
Play That Tune App
Grades 6+ | Blocks

<https://hourofcode.com/ca/learn>

Hour of Code Activities... Play That Tune

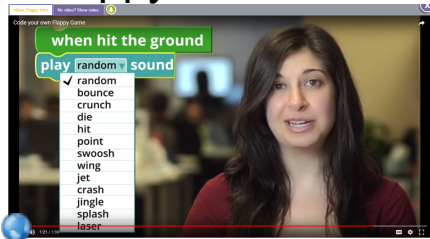


Welcome to the Mobile CS Principles Hour of Code! In this activity, you will build code to play the tunes given in the puzzles. After you complete all the levels, click on Create to code your own tune and then package it as an Android app. You can also try MIT App Inventor, which has similar block programming to create mobile apps. App Inventor is used in the AP high-school Mobile CSP curriculum.

To start the puzzles, click the button below and listen and watch for the red dots on the piano keys on the left.

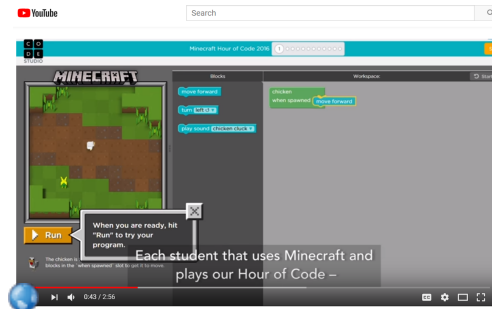
[Play First Puzzle Tune](#)

Flappy Bird Game



[MSTE 6 - Hour of Code.docx](#)

Minecraft



Frozen



Class Activity...Ping Pong Rescue

- **Objective** - write a program that instructs a student, who is blind folded, to pick up a ping pong ball
- **Need** - A blind folded Robot and a person who calls out the commands
- **To Do...**
 - > With the person blind folded...decide on a START/FINISH
 - > Insert a few obstacles
 - > Write the code
 - > TRY IT!

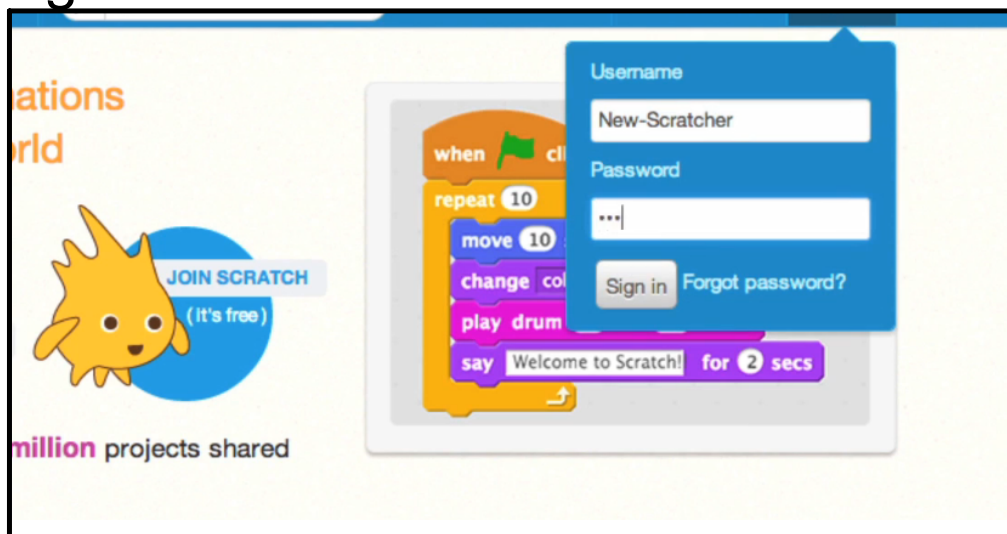
Introduction to SCRATCH...



<https://vimeo.com/65583694>



Getting Started with SCRATCH...



<https://vimeo.com/80961102>



CREATING WITH SCRATCH

People have access to an incredible variety of interactive games, stories, animations, simulations, and other types of dynamic, interactive media on their computers today. But for the most part, these programs are a one-way street: you can only browse and click what others have created; you can't design and create your own.

Scratch changes that, broadening the range of what you can design and create on the computer, making it easier to combine graphics, photos, music, and sound into interactive creations. With Scratch, you can create characters that dance, sing, and interact with one another. Or create images that whirl, spin, and animate in response to movements of the mouse. Or integrate images with sound effects and music clips to create an interactive birthday card for a friend, or an interactive report for school.

The name *Scratch* comes from the *scratching* technique used by hip-hop disc jockeys, who spin vinyl records back and forth with their hands to mix music clips together in creative ways. You can do something similar with Scratch, mixing different types of media clips (graphics, photos, music, sounds) in creative ways.

At the core of Scratch is a graphical programming language that lets you control the actions and interactions among different

media. Coding in Scratch is much easier than in traditional programming languages: to create a script, you simply snap together graphical blocks, much like LEGO bricks or puzzle pieces.

Once you've created a Scratch project, you can *share* it on the Scratch website, the same way you might share videos on YouTube or photos on Flickr. Or you can embed your Scratch project in any other webpage—for example, embedding an interactive Scratch animation on your MySpace or Facebook homepage.

You can get new ideas for Scratch projects by browsing through projects on the Scratch website. If you like one of the characters or images or scripts in another project, simply *download* the project and use parts of it in your own Scratch project.

Below are snapshots from projects that other people created with Scratch. What do you want to create with Scratch?



Library: Kindergarten Group, MIT Media Lab

<http://scratch.mit.edu> SCRATCH

creating-with-scratch.pdf

LEARNING WITH SCRATCH

What do students *learn* as they create interactive stories, animations, games, music, and art with Scratch?

For one thing, they learn **mathematical and computational ideas** that are built into the Scratch experience. As students create programs in Scratch, they learn core computational concepts such as iteration and conditionals. They also gain an understanding of important mathematical concepts such as coordinates, variables, and random numbers.

Significantly, students learn these concepts in a **meaningful and motivating context**. When students learn about variables in traditional algebra classes, they usually feel little personal connection to the concept. But when they learn about variables in the context of Scratch, they can use variables immediately in very meaningful ways: to control the speed of an animation, or to keep track of the score in a game they are creating.


As students work on Scratch projects, they also learn about the **process of design**. Typically, a student will start with an idea, create a working prototype, experiment with it, debug it when things go wrong, get feedback from others, then revise and redesign it. It's a continuous spiral: get an idea, create a project, which leads to new ideas, which lead to new projects, and on and on.

This project-design process combines many of the **21st century learning skills** that will be critical to success in the future: thinking creatively, communicating clearly, analyzing systematically, collaborating effectively, designing iteratively, learning continuously.

Creating projects in Scratch also helps students develop a deeper level of **fluency** with digital technology. What do we mean by fluency? To be considered fluent in English, Spanish, or other language, you must learn not only how to read but also to write—that is, how to express yourself with the language. Similarly, to be fluent with digital technology, you must learn not only how to interact with the computer but also to create with it.

Of course, most students will not grow up to become professional programmers, just as most will not become professional writers. But **learning to program** offers benefits for everyone: it enables students to express themselves more fully and creatively, helps them develop as logical thinkers, and helps them understand the workings of the new technologies that they encounter everywhere in their everyday lives.

References
 Reinventing Learning in the Digital Age
<http://www.media.mit.edu/~mhs/scratch2012.pdf>
 Learning for the 21st Century (<http://www.21stcenturyskills.org/>)



Library: Kindergarten Group, MIT Media Lab

<http://scratch.mit.edu> SCRATCH

learning_with_scratch.pdf

PROGRAMMING WITH SCRATCH

Most people view computer programming as a tedious, specialized activity, accessible only to those with advanced technical training. And, indeed, traditional programming languages like Java and C++ are very difficult for most people to learn.

Scratch, a new graphical programming language, aims to change that. Scratch takes advantage of advances in computing power and interface design to make programming more engaging and accessible for children, teens, and others who are learning to program. Key features of Scratch include:

- Building-block programming.** To create programs in Scratch, you simply snap graphical blocks together into stacks. The blocks are designed to fit together only in ways that make syntactic sense, so there are no syntax errors. Different data types have different shapes, eliminating type mismatches. You can make changes to stacks even as programs are running, so it is easy to experiment with new ideas incrementally and iteratively.
- Media manipulation.** With Scratch, you can create programs that control and mix graphics, animations, music, and sound. Scratch extends the media-manipulation activities that are popular in today's culture—for example, adding programmability to Photoshop-style image filtering.

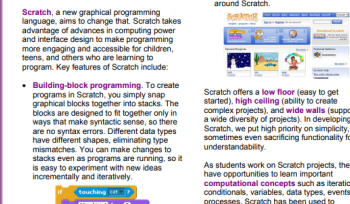
Sharing and collaboration. The Scratch website provides inspiration and audience: you can try out other people's projects, reuse and adapt their images and scripts, and post your own projects. The ultimate goal is to develop a shared community and culture around Scratch.

Scratch offers a low floor (easy to get started), high ceiling (ability to create complex projects), and wide walls (support for a wide diversity of projects). In developing Scratch, we put high priority on simplicity, sometimes even sacrificing functionality for understandability.

As students work on Scratch projects, they have opportunities to learn important computational concepts such as iteration, conditionals, variables, data types, events, and processes. Scratch has been used to introduce these concepts to students of many different ages, from elementary school through college. Some students transition to traditional text-based languages after getting introduced to programming with Scratch.

Scratch is built on top of the *Squeak* programming language. It was inspired by previous work on Logo and Squeak Etoys, but it aims to be simpler and more intuitive.

Scratch is an open-source but closed-development project. The source code is freely available, but the application is developed by a small team of researchers at the MIT Media Lab.

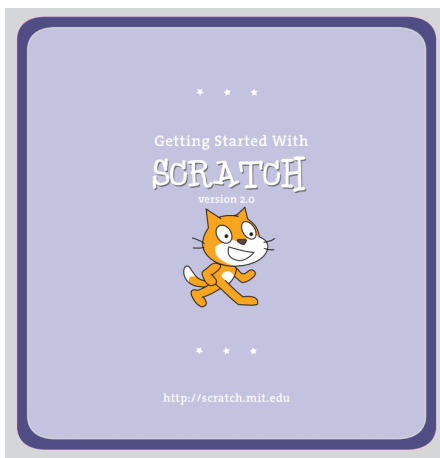


Library: Kindergarten Group, MIT Media Lab

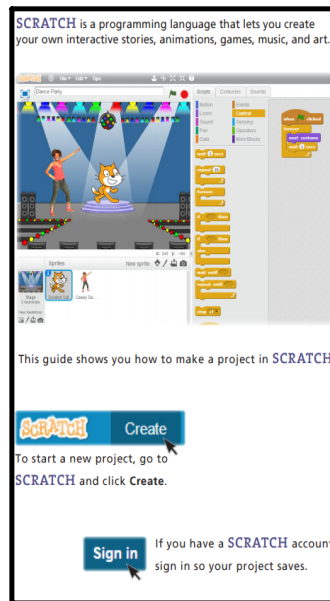
<http://scratch.mit.edu> SCRATCH

programming-with-scratch.pdf

First Program...



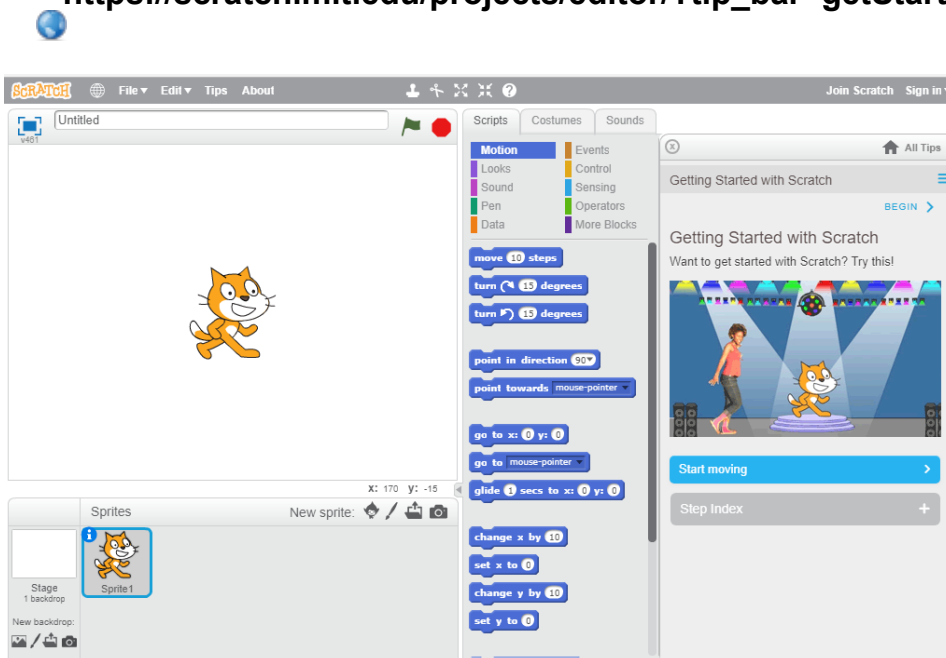
Scratch Intro Lesson.pdf



SCRATCH is a programming language that makes it easy to create your own interactive stories, games, and animations...and share your creations with others on the web.

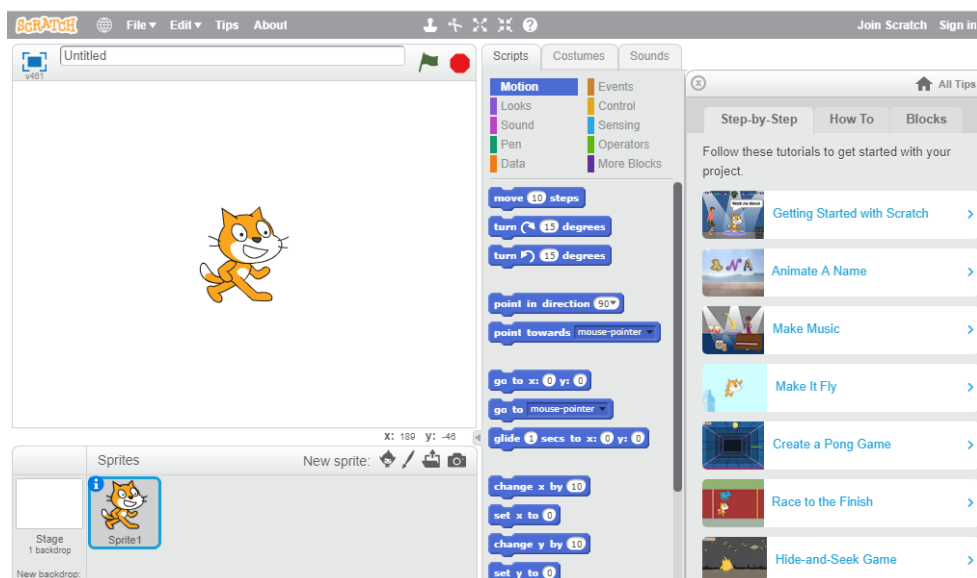
ONLINE TUTORIAL...

https://scratch.mit.edu/projects/editor/?tip_bar=getStarted



TIPS FEATURE...

https://scratch.mit.edu/projects/editor/?tip_bar=home



ACTIVITIES TO TRY...

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

Scratch Create Explore Tips About Search Join Scratch Sign in

Things to Try

What do you want to make with Scratch? For each activity, you can try the **Tutorial**, download a set of **Activity Cards**, or view the **Educator Guide**.

Tutorial

Animate a Name

Animate the letters of your username, initials, or favorite word.

[See Cards and Guides](#)

Tutorial

Make It Fly

Animate the Scratch Cat, The Powerpuff Girls, or even a taco!

[See Cards and Guides](#)

Tutorial

Make Music

Choose instruments, add sounds, and press keys to play music.

[See Cards and Guides](#)

STARTER PROJECTS...

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



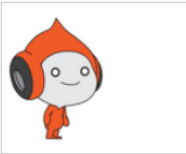
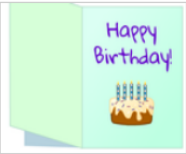
Scratch Create Explore Tips About Search Join Scratch Sign in

Starter Projects

Try out these starter projects from the Scratch Team. Look inside to make changes and add your ideas.

Animation Games Interactive Art Music and Dance Stories Video Sensing

Animation

-  Dance Party
-  Animate the Crab
-  The Pico Show: Intro
-  Greeting Card

Attachments

MSTE 6 - Hour of Code.docx

creating-with-scratch.pdf

learning_with_scratch.pdf

programming-with-scratch.pdf

Scratch Intro Lesson.pdf