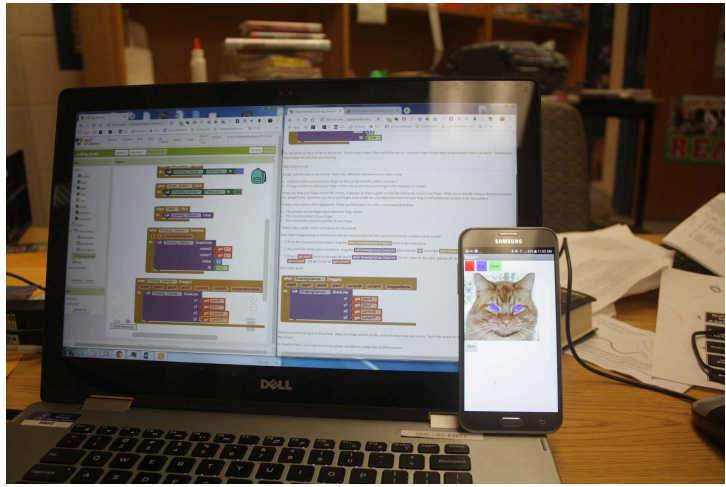
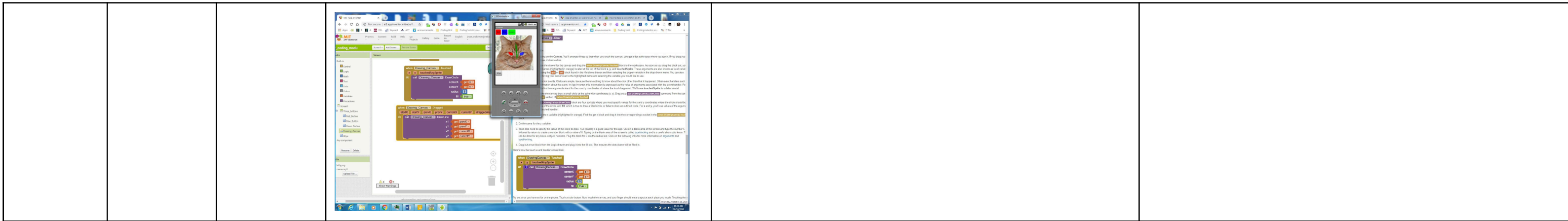


Exploring Coding Resources

Directions: Select one of the coding resources provided in D2L to explore. Everyone must explore a different resource (first come, first serve) and not all will be covered. Once you've spent some time exploring the resource (recommended at least 30 minute), complete your row in the table below.

Student Name	Coding Resource	Age Level	Description - Briefly describe the activity you completed on the resource (include a screenshot of final product)	Resource Format (self-led tutorials, teacher led activities, step-by-step instructions)	What level of coding knowledge might a teacher / librarian need to utilize this resource with students?
<p>EXAMPLE: Jesse Mckeever</p>	<p>MIT App Inventor</p>	<p>Students as young as 6th grade have used this though having some basic coding understand would be beneficial before using this product</p>	<p>I completed the basic PaintPot Tutorial which consisted of creating an app with a picture of a cat that you could draw on with three different colors and allowed the screen to be cleared. This resources breaks down creating an app into a design element and a coding element which was nice. I choose this app because it had an easy to follow tutorial, however looking at the directions if feels as though the tutorial expects users have a somewhat basic understanding of logic and coding. The interface had a lot of features which could be a positive or a negative. It also focuses on the Android platform. In order to test the final App you could either run an emulator (which needs a seperate program to be installed on a window, linux or mac), which is nice as a phone is not needed or you could download the MIT inventor app. In my case I followed the tutorial to connection the MI inventor app wirelessly with my computer / coding work space and able to run the app I create right on my android phone, which was pretty cool. I will say I attempted to connect my phone at UW whitewater to my computer and had some issues but everything worked fine once I was in my own district's network. Pictures below show the emulator (which does look dated but worked) and app running on my phone. My actual code workspace is on the left and the tutorial for the project is on the right.</p> 	<p>I didn't look through all the projects available on the website but I did see a good many projects that offered tutorials to walk through. If I was a teacher I would want to mess around with a few tutorials to get familiar with the emulators and the layout. If you had a group of middle schoolers or older that had a basic level of programing a teacher familiar with this platform could walk students through a few tutorials to get them familiar with the layout and operation of this resource and then either let them go on more difficult self paced lessons or possibly let them try and create a totally new app.</p> <p>On their site the highlighted a few students who have really leveraged this platform. One was a group of 6th graders in Texas who built an app to help their blind classmate navigate the halls in their school (this was back in 2014).</p>	<p>I would think a person wishing to utilize this platform with students of any age should have a basic understanding of coding. If you are a self learning I think one could spend a few hours doing some tutorials to gain an understanding of how they platform works. However it would probably be a good idea to do some research and or start on a bit more basic platform which can cover some of the essentials of coding. Additionally I could see how this resource could easily cause some not technical people headaches, mainly through the emulator (needing permissions to install and update) or students who don't have Android phones might feel left out that they can't run their app on their phone / show it off to people.</p> <p>It's a powerful resource with other students have used to create useful apps of their own, so this could be really hander for intermediate programs as well as a somewhat easy way give students a tangible way show off something they've coded (by it being on their phone), but I might shy away from this platform as an introduction to coding for some students.</p>



Kristin Henning

[Compute It](#)

This was designed to be suited for beginners who have never coded.

```

function fall () {
  while { } {
    down ()
  }
}
function eye () {
  repeat (2) [
    right ()
    if { } {
      fall ()
    }
  }
}
while { } {
  up ()
}
repeat (3) [
  left ()
  if { } {
    down ()
    right ()
    eye ()
  }
}
  
```

Compute It was designed to have the user think like the computer. A series of functions were given which the user had to interpret and follow. In all, there are 58 challenges a user must get through in order to achieve the end goal (which you see pictured here). The functions start out simple and increase in difficulty as the coder accomplishes the task at each level. It is challenging but fun at the same time. I wanted to make it through all of the levels just to say I could. I worked my way through basic commands, repeat functions, and multi-step functions with exceptions and additions. If I lost my place, I often made a mistake and had to start over. This required quite a bit of focus.

There are no tutorials available. The designers really just want kids to jump right in and engage in the programming. They are trying to build critical thinkers. This is a great way to introduce coding without fear of failure and it is a great way to build grit.

There is a disclaimer for educators which they provide. It states:

Warning For Educators !

We designed “Compute it” to be usable by educators with zero knowledge, or even without educator. If you plan on using “Compute it” with a classroom or friends of yours, and if you are already comfortable with the notions we use, try not to explain too much, and let the players figure it out by themselves.



It’s easier for the beginners to be introduced to abstract concepts by practicing than by using complex words. And figuring it out will reward them with a “eureka” moment, and have more impact.

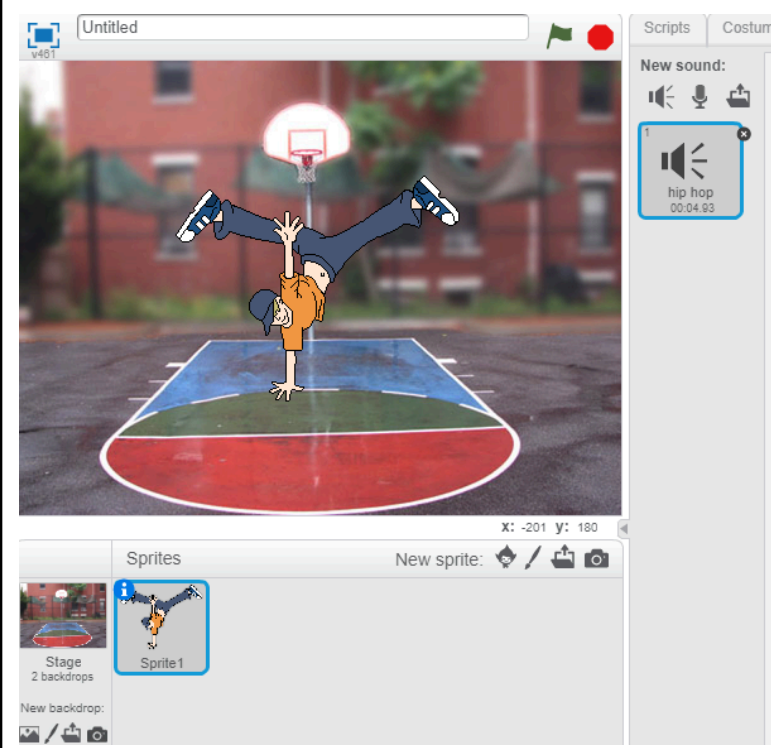
We witnessed it several times in our test sessions : beginners assisted by an expert friend typically have a slower progression.

To have a positive effect as a guide : only provide small hints, and only when the player is asking for it or seems stuck.

Here is an image of the first task:

The only prior knowledge a teacher or librarian would need to know is the difference between a main command and a function command. Once that is understood, the rest is self explanatory. If anything, I would suggest that the teacher or librarian play with this tool for a bit just to understand the task kids will be attempting and to gain a feel for their potential pitfalls and frustrations. I think the creators are correct that this is a module for beginners. This is a great tool for introducing coding.

<p>Leah Scheid</p>	<p>Made with Code</p>	<p>Tween/Teen Girls</p>	 <p>Since I know next to nothing about coding, I tried their “Rookie Collage” project to make a GIF using Blockly coding. I thought it was rookie as in newbie, but when I completed the project it offered to have participants send their GIF to be published on rookiemag.com, which is an online magazine for writing and art for teens. To create this project, they had you pick a canvas (background), add objects, add modifiers to the objects (position x or y, scale, animation, rotation). You could choose how you shared it. I downloaded the GIF to upload it here. It gave other social media options.</p>	<p>The project had self-led tutorials with step-by-step directions, which were very easy to follow. It gave a description of what you are going to create and the coding skills you’ll use to create it (this one was Encapsulation, Objects, and Properties). There were left-side tabs for adding code including a canvas (background), modifier blocks to control position, scale, rotation and opacity of objects. There were also animation blocks already added to the code.</p> <p>Each modifier had a question mark for help and there was a help button. There was also a hint button at the top.</p> <p>When I completed the coding, they reviewed the coding concepts in this project and gave several options for sharing the project.</p>	<p>This project was very easy, so I don’t think a teacher would need to know a lot about coding to have students try these projects. This would be a good tool to introduce coding.</p> <p>One issue that you may have with this resource is that boys and tomboy girls may not be interested in using it. My daughter is a sophomore and she was offended at the girly-ness of the website. My niece is a freshman, and she thought it was “cute” and it would be fun to try the projects.</p> <p>I would definitely have this site as an option for students discovering coding projects independently.</p>
<p>Angela Brown</p>	<p>Scratch</p>	<p>8-16</p>	<p>Scratch offers many activities to follow-I am familiar with Scratch Jr. on the iPads but not Scratch via web browser. I choose one that seemed fairly simple. It had directions to follow to create a breakdancer code. While the directions were step by step, this particular activity was a bit overwhelming. I would even suggest the age difference to start at age 10. I had to choose a background, dancer, music, and have my dancer switch poses every 5 seconds.</p> 	<p>This project had step by step directions-although at times the directions were not clear and the process was confusing.</p>	<p>A librarian or teacher using this would need to know the basics of coding. They would also need to familiarize themselves with the website specifics and activities. More time is spent figuring out where things are than how the program works.</p>

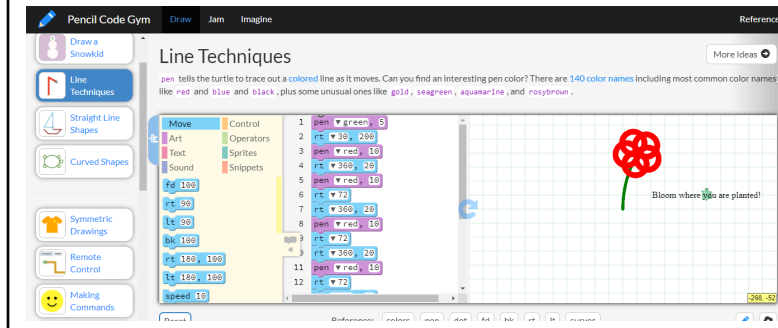


Tina Kiekhaefer

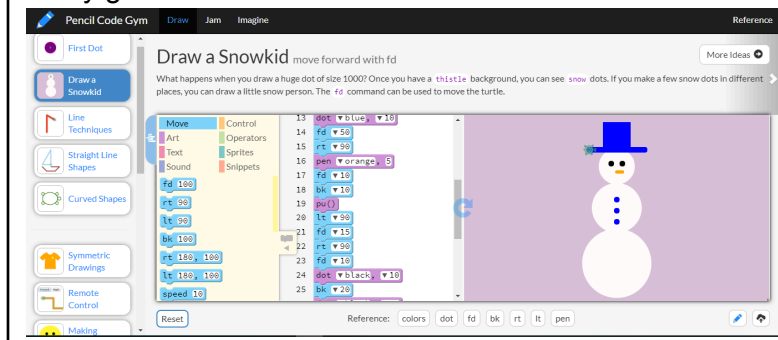
[PencilCode](#)

The Pencil Code Teacher's Manual is intended for high school students. However, the activities on the website vary in complexity. I think some may be appropriate for middle school students.

As a person with little coding experience, I started by exploring some of the beginner and intermediate printable activities that give you step by step coding directions to achieve a given result. This is an example of a beginner activity called "Drawing a Flower". I followed the basic directions, but experimented a bit with color and added some text.




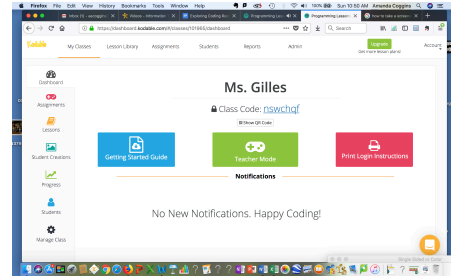
Using what I learned through that exploration, I completed the "Draw a Snowkid" activity within the draw part of the Pencil Code Gym. The first few commands were provided to build the base of the snowman. Then I added different move and art commands to build the middle and top parts of the snowman, along with buttons, a nose, eyes, and hat. This required quite a bit of trial and error for me, but I finally got it to work!

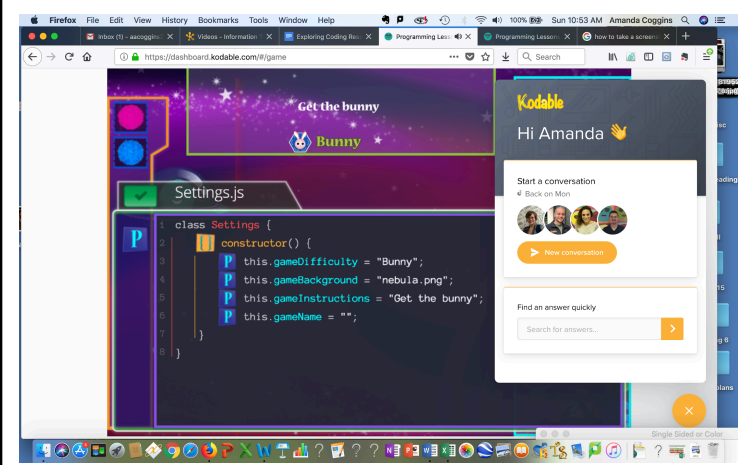


Pencil Code allows users to create art, music, games, and stories. It is also a place to experiment with mathematical functions, geometry, graphing, webpages, simulations, and algorithms. There is a short video overview and video tutorial within the site (which give general information about Pencil Code), as well as a lines video and angles video within the resource links. However, these videos are not easy to find.

Pencil Code provides a variety of online ideas, examples and activities for art, music, and games/stories within the Pencil Code Gym, as well as allowing students to start with a blank page. There is a lot to experiment with and explore! The website contains helpful resources including materials for teachers, a teaching manual, a Pencil Code book (programming primer with a choice of over 100 activities), and printable activities at the beginner, intermediate, and advanced levels designed for classroom or home use.

To use Pencil Code with students, I feel some coding knowledge is needed and beneficial. I have very little experience with coding and was able to learn through the printables and apply what I learned to try some of the activities within the Pencil Code Gym. However, some of the intermediate and advanced printables and activities in the Pencil Code Gym were well beyond me! I would definitely need to devote more time to self-teaching and exploration before I could feel comfortable using this resource with students. Luckily, there seem to be some good teaching materials and resources available to support teachers in this endeavor!

Mandy Meloy	Scratch Jr	Advertises for ages 5 to 7	<p>Can get it via app on Google product including chromestore, Apple, and Amazon.</p> <p>I chose a background (the sea) and chose characters (crab and whale). I played with the color of the crab and then ended up keeping it the same. I programmed the crab to move across the ocean floor and then up stopping to say "hi" and to ask "am I too fast for you. I then programmed the whale to continue across the screen. I played with the sound and recorded my version (voice) of a whale sound. I then tried to get the crab to hide behind the whale. I also titled my story the "Speedy crab"</p> 	<p>As I entered the chrome version on my chromebook, there was a house for home or a question mark for tutorial. I tried to click the tutorial but it did not work. I am wondering if the app would be better. I then went to youtube and watched an introductory of scratch junior. After learning the basic steps, I played with it and created my scene. It took me a few minutes and I used my creativity. I then realized if I hovered over the blocks that the description would come up as words. I think this would have to be teacher led at the very beginning and then the students could play with it. Many of my 5 and 6 year olds have difficult reading. The colors, visuals, and buttons to click make it much easier and appealing to pre-readers.</p> <p>There is also curriculum for teachers with assessments in https://www.scratchjr.org/teach/activities</p> <p>Along with a brief introduction: https://www.scratchjr.org/learn/interface</p>	A very basic understanding. This could be fun for K to 3 or so. Scratch junior invited those with a basic understanding of code to learn and gain confidence in instruction and following steps to complete a process.
Amanda Coggins	Kodable	Variety of activities, students as young as 2nd grade may use this.	<p>Kodable was designed and built from the ground up to meet the needs of teachers (according to the website.) The site has scripted lesson plans and assessment tools to use. The website has a ton of information to investigate, I played around with several different tools. After looking around the site for a while, I decided to try and create a game. I thought this would be an activity older students would enjoy at my district. As I have never done any coding before it was very impressive to see the different options and to explore. I plan on using Kodable with my 5th grade students and then hopefully for my enrichment groups with younger classes. I definitely need to explore the site more and figure out all of the different options available before teaching this to students. I've included in my screenshot that it also has a "help message" that you can use if you are having difficulty. I think that is a great added feature for students.</p>	<p>The site does have tutorials to help users become familiar with the different lessons, also includes testimonials from teachers that have used the product. If your school uses Google, you can login in with your account and will need to create classes for your students to be able to access the site. You are given a code students to use. After you have created your classes and imported students it takes you to "dashboard" page. This has different option on sidebar that allows you and your students to view the lesson/level of assignments, records progress, and also student creations. Another option that I really like is that it recommends what age group to use each activity for.</p> 	To use Kodable, I don't think teachers/librarians would need a lot of prior knowledge to use this. It would be very beneficial for teachers to look over the different options and activities included on this website. The website also has different levels of coding lesson plans/activities, of course to receive more curriculum/options you would need to purchase the advancements. I also think it is beneficial that Kodable shows you what standards each lesson meets.

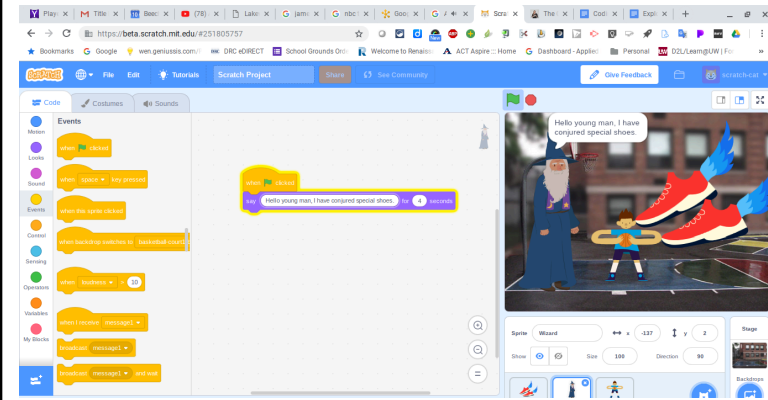


Vince Czahor

[Google CS First](#)

.9-14

I did the Google CS First “An Unusual Discovery” where I watched a video on coding and starting to tell an animated story by coding with Scratch. I have to choose one of four backgrounds with an object that appears and I had to add two characters and then program the first character to say some dialogue. I choose some special flying shoes on a basketball court, added a basketball player and wizard, and then some dialogue. I had the wizard say, “Hello young man, I have conjured special shoes.”



The resource was the CS First website. I signed in with my school Google account and as a teacher, I was able to see all of the curriculum. There was a great teacher help section here: <https://csfirst.withgoogle.com/en/teachers#training> where there were videos to get started. The activities all had difficulty ratings and easily could be added to Google Classrooms. Creating a club looks very easy. This resource is very dependent on Scratch.

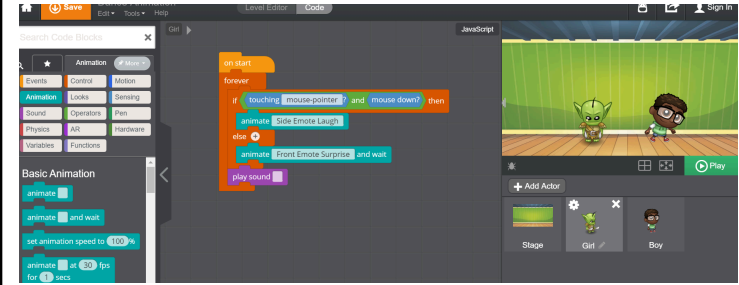
Teachers do not need a great deal of coding knowledge to use CS First. The teacher training will get teachers up to speed and also teachers can test out all activities they want to assign. It is very fun and the lessons in CS First are activities that utilize the power of Scratch with more focus on completing specific tasks. Upper elementary and middle school teachers and librarians could use CS First with ease.

Amanda LaLuzerne

[Tynker](#)

Ages 7 and up

I created a character and then set him up to dance when the music plays. I followed a easy to complete tutorial since I have no experience with coding. There are several different activities to choose from on the site. It really seems there is something for everyone.



There are self led tutorials, step to step directions that you can read, some are read aloud to you. There is definitely a variety of levels when it comes to tutorials. Teachers can create a classroom and give students a code. They can also link it Google classroom.

I think someone with any level of coding knowledge could use this resource. The tutorials help guide along but some also give the option to speed up the tutorial as well. The site also provides a curriculum and scope & sequence for coding and STEM.

Jamie Jestadt

[Code.org](#)

They have activities and modules for all ages!

I did the first activity listed for the youngest students in Code.org called, *Pre-reader Express*. This is an offline activity described below. I applied the concepts to a Marble Run activity with my daughters. Here is the [final result](#).

Code.org has a variety of instructional formats but the main premise of the site is to give students the opportunity to take self-paced lessons. Students can see which modules they’ve completed as they work their way through the course. Teachers can also assign specific modules within the course to students and see their progress. There are teacher led lessons included

I have very little knowledge other than what I’ve learned through this course and I was able to navigate it quite easily. Teachers can sign in with Google and then create a class to import their students. Once students are in the module, they can easily work step by step through the courses and track their progress. They even provide teacher courses!



This lesson will guide students through the steps of debugging. Students will learn the mantra: "What happened? What was supposed to happen? What does that tell you?"

- Express that they have noticed when something goes differently than what is expected.
- Identify what the expected result was before an error occurs.
- Determine and describe the difference between what was expected and what actually happened in the event of an error.

Research shows that some students have less trouble debugging a program than writing one when they first learn to code. In this lesson, we introduce the idea of debugging in a real world sense.

The goal in this lesson is to teach students steps to spot a bug and to increase persistence by showing them that it's normal to find mistakes. In later lessons, students will debug actual programs on Code.org.

with lesson plans that connect with some of the modules. They also provide step-by-step instructions for student activities.

Sarah Sheleski

[Microsoft MakeCode](#)

All ages - younger children will need adult support.

Areas I could not utilize without purchasing items included (but sure look fun and interesting):

(1) Micro:Bit - The micro:bit is a small programmable computer with an LED screen, two buttons, a light sensor, Bluetooth antenna, accelerometer, compass and more! From dancing robots to banana keyboards, your micro:bit has all the features you need to create amazing projects—the possibilities are endless!

(2) Circuit Playground Express: The Circuit Playground Express provides a great introduction to electronics and programming. It has 10 multi-color NeoPixels, a motion sensor, temperature sensor, light sensor, and a microphone and speaker, plus much more! Create a siren alarm, a watch timer, or a magic wand with just a few blocks of code.

(3) Minecraft: Minecraft is a game about placing and breaking blocks, where the only limit is the player's imagination. It's also a creative platform for teaching and learning computer science—build a program to summon a storm of chickens or create your own mini-game for friends to play!

(4) Lego Technology and computer science come to life

Once you do purchase a supporting item from Microsoft, there are tons of well laid out lesson plans with step by step directions and visuals as well as connecting to others creations.

These step by step directions would be easy enough for a teacher to learn and to support students in their investigation or even for older learners to read and trial these with independence.

Given the well laid out lesson plans with visuals and the supporting connection to others creations, I feel that novice could use these ideas, given that they had the funding to purchase the required coding supports.

They actually have a page dedicated to teachers to support their learning! <https://www.microsoft.com/en-us/makecode/resources>

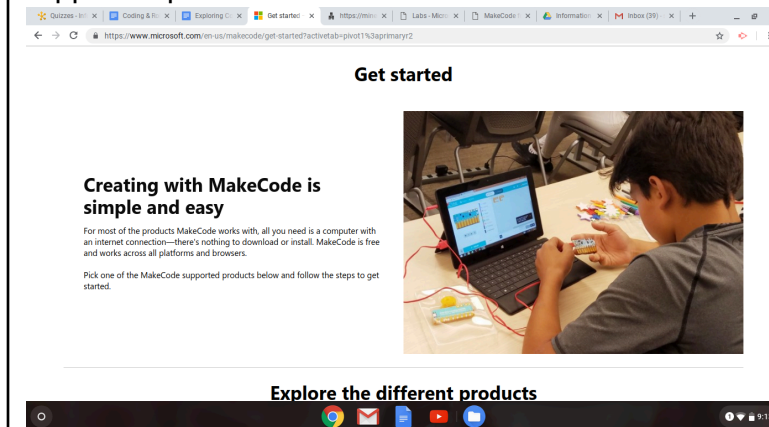
with LEGO MINDSTORMS Education EV3 which comes with different sensors and motors that can be programmed. Students can build robots that can navigate mazes, sort colors, or act as a musical instrument!

(5) CUE Robot: Cue is a clever robot with a witty attitude that's full of interactive surprises. It also has emotive AI systems, IR sensors, microphone and speaker, and programmable LED lights and buttons. Students can code Cue's creative interactions using MakeCode's Block or JavaScript programming editors.

(6) Chibi Chip: Learn technology through arts and crafts with Chibitronics paper circuits products—which include innovative LED light stickers and The Love to Code Creative Coding Kit. Students learn electronic circuitry and computer science, while creating interactive art pieces like storybooks and greeting cards.

(7) Grove Zero This set of plug-and-play electronic modules are programmable with Microsoft MakeCode. Students can snap modules together using magnet connectors in any configuration. The Grove Zero Starter Kit comes with a buzzer, sound sensor, light sensor, LED light matrix, buttons, temperature sensor, and more.

I was able to view others coding on GitHub as well as view many different well laid out lesson plans on the site, but all requires you to purchase one of the above mentioned items or have Windows 10 to purchase minecraft. Says right on the page “all you need is a comptuer and internet connection...and to purchase a supported product.



Rebecca Blom

[Thunkable](#)

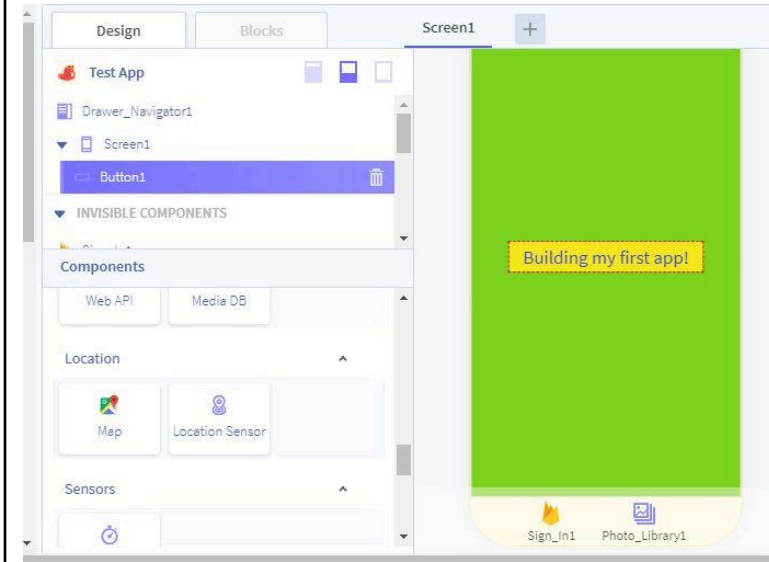
13 years or older

I made a functioning app! It even worked once I downloaded it onto my phone. I will need to dig a little more into the app, but I would love to use this or something similar to create an app for our Maker Faire! It will need a lot of time and design, but the fact that I got this to talk to me just by pressing a button on my phone

When you login to Thunkable, you will notice tutorials on the left hand side that you can watch, that will cover adding in certain functions to your app. Additionally, there is a community that you can partake in. In here you will find a lot of different ideas floating around, and all levels of expertise. It would be a great resource to someone who is just starting out (like myself).

With no knowledge of coding I found this pretty simple. The ages are 13 and up, and I feel that if you are staying within that parameter the student should be fine working with this program and the teacher should be fine teaching it as well. The website also provides video tutorials on how to add certain things to the app.

blew me away. I feel like one of our students!

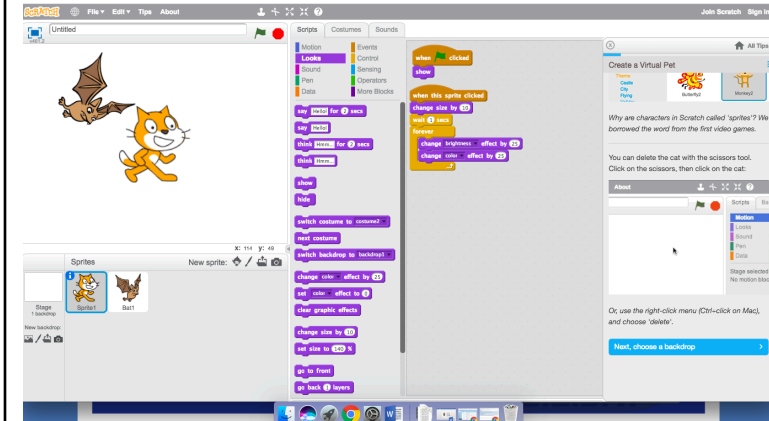


Margaret Ottenad

[Scratch](#)

Scratch is designed especially for ages 8 to 16, but is used by people of all ages. Millions of people are creating Scratch projects in a wide variety of settings, including homes, schools, museums, libraries, and community centers.

Scratch is so fun! My son is helping me learn how to navigate. It was way easier with his coaching. I read the tutorial and clicked around for quite a while, but he was helpful in making different animations.



The step-by-step direction was open next to the work space. I found this helpful. The steps were clear and easy to follow.

I like to dive right in. I don't think much prep is necessary. I would plan and demonstrate for my class, but I also wouldn't hesitate to turn kids loose with little guidance. It think this is best implemented in a group work setting. Letting the kids talk and share is great. They can coach each other to create great things.