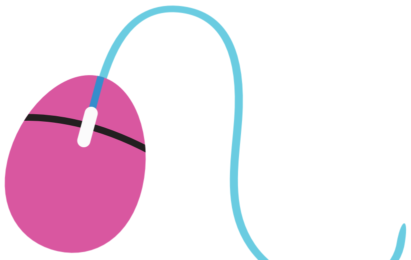


# ACTIVITY GUIDE FOR FAMILIES

verizon innovative learning

# *HOUR OF CODE

# Toolkit: Coding for families 101

## Have an hour to spare?
## Let's get your family coding!

This family guide will help you enjoy learning how to code with three simple exercises — no previous experience required. These tools are free, the activities are lots of fun, and in an hour you'll be able to call yourselves coders!

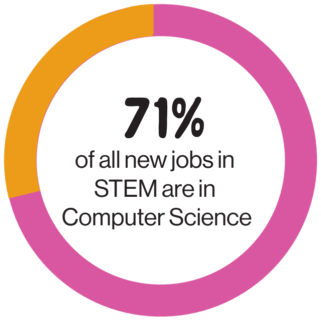
## What is coding?

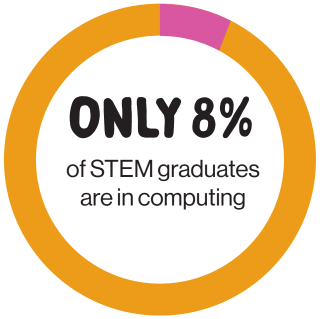
When you code or program, you're writing instructions to a computer to get it to do something — these instructions are called programs. They are written in special languages and rules designed for a computer to understand.

There are tools and apps that help kids code without them having to know advanced computing, so that they can more easily learn programming and thinking skills. This helps them grasp concepts like algorithms, recursion, and heuristics while keeping the learning process light and easy. That's what we'll do in this hour.

## Why should kids learn to code?

1. Learning to code helps kids express themselves, grow confidence, build things, solve problems, and understand computational thinking and logic.
2. The skills that kids learn from coding can apply to other subjects like math, science and art.
3. Kids can better understand technology by interacting AND creating with it.
4. We need more coders!





(resource: code.org)

# Getting started

## What will we use to learn coding?

There are many programs that you can use with your family to learn coding. For this guide, we will be using Scratch. Scratch is a free programming language and online community for novice coders and kids. It's easy to use and includes support tools, libraries of fun graphics, sounds and activities that kids love. Scratch's community of makers, artists, kids and teachers share projects and encourage each other to code.

Scratch lets you design programs by connecting colorful blocks. The groups of blocks are called scripts, and they tell the sprites (or "characters") to do something. Through the online application, you can make interactive stories, animations, art, and music, and share your creations with a community of new coders just like you.

This guide has three Scratch exercises to choose from. All three will take an hour. Scratch was developed by the Lifelong Kindergarten Group at the MIT Media Lab.

## Where can I get Scratch?

Scratch is an online programming language so you can start using it here: https://scratch.mit.edu/.

## Key terms

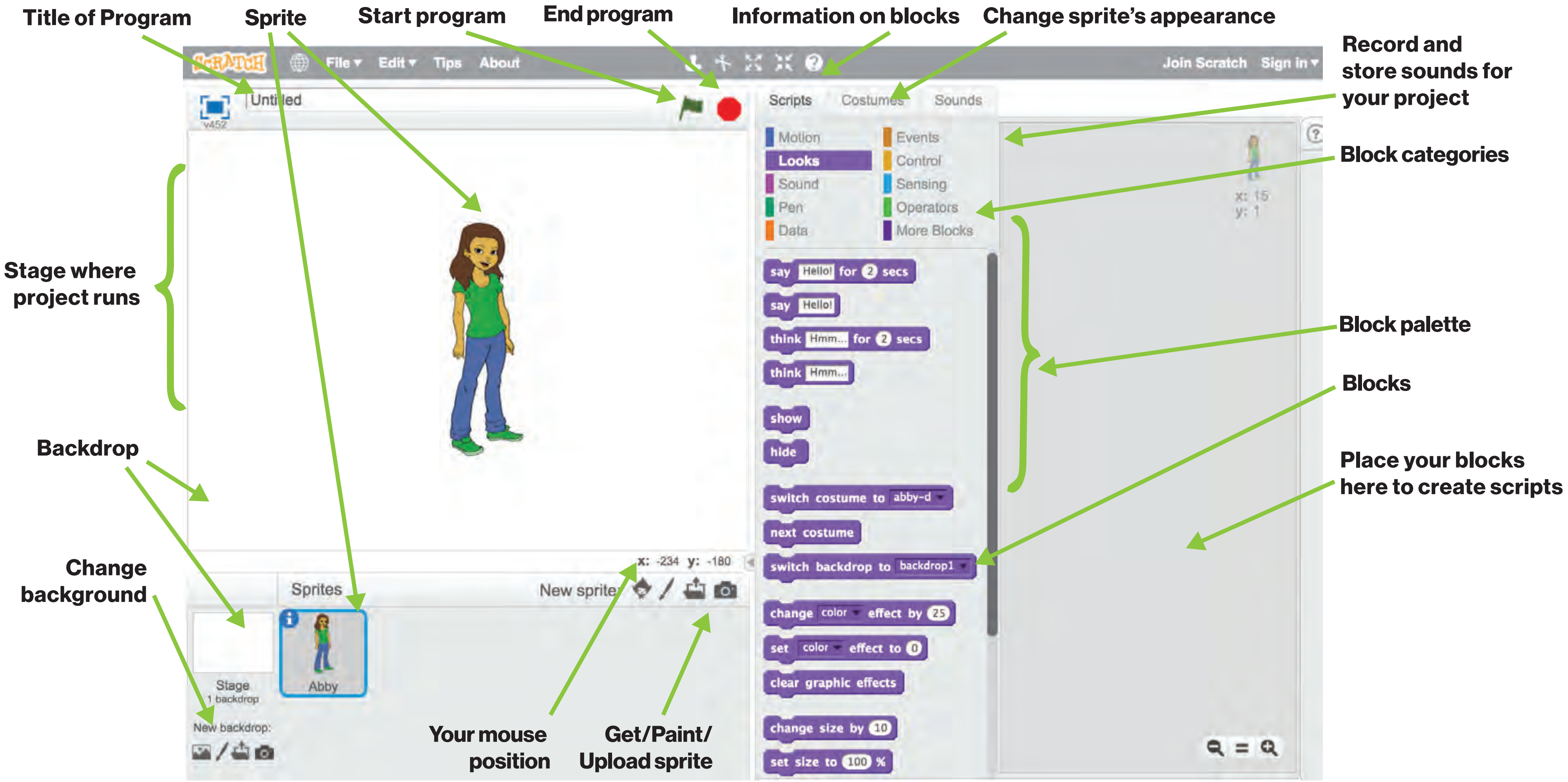
Before diving in, let's get some definitions of things we'll see in Scratch!

- **Animation** – A project that makes pictures look like they are moving.
- **Backdrop** - The picture behind the sprite/character on the stage.
- **Block** - An instruction. A series of blocks make up scripts, which make the project work.
- **Condition** - An if-then statement that is used to make a decision in the program.
- **Costume** - Your sprite's appearance.
- **Event** - Something that happens on the computer (e.g. a mouse click).
- **Library** - A collection of sprites, sounds, or costumes.
- **Loop** - An instruction that makes other instructions repeat themselves.
- **Program** - A list of instructions that tells a computer what to do.
- **Run** - Start a program.
- **Script** - A script is a collection of blocks that are placed together. The blocks and their order determine how sprites interact with each other and the backdrop.
- **Sprite** - A picture on the stage that a script can do things to.
- **Stage** - The main area in Scratch that holds the sprites and where the project runs.
- You can review all Scratch terms here: https://wiki.scratch.mit.edu/wiki/Scratch_Terms_Glossary

# Getting started

## The Scratch screen

When you log into the Scratch platform and create a new project, you will see this screen. We've pointed out some of the key things you'll be using on the screen. We encourage you to click around the screen — the best way to learn is to explore.



Title of Program

Sprite

Start program

End program

Information on blocks

Change sprite's appearance

Record and store sounds for your project

Block categories

Stage where project runs

Block palette

Blocks

Backdrop

Place your blocks here to create scripts

Change background

Your mouse position

Get/Paint/ Upload sprite

# The exercises

## What will you learn?

With each of these three exercises, you will learn how to:

- Make a character do something through basic programming commands.
- Create an original project of your own.
- Animate it!

## Exercise 1 - The basics

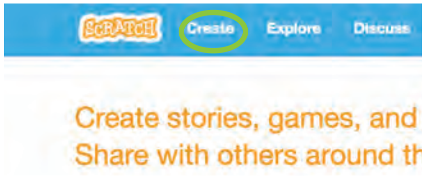
### Decorate the sprite, make it move and have it say something!

In this exercise, you will learn how to create a character, called a Sprite. We will use the sprite and learn how to write a simple program as well as basic "conditions." Conditions are if/then statements in which you have the program execute instructions depending on whether a statement is true or false.

1. Open up the program.

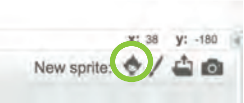
Step 1: Go to https://scratch.mit.edu/

Step 2: Click on "Create" in the top left menu to create a new project. You will see a sprite on the stage. Delete the sprite.



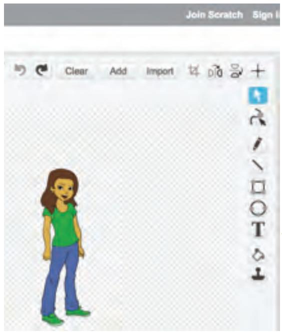
Create stories, games, and
Share with others around th

2. Choose a sprite from the sprite library.



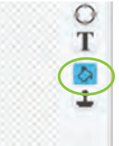
3. Change the sprite's costume.

Step 1: When you're on the main project screen, click on the "Costumes" panel. Sometimes there are already multiple costumes for the sprite loaded into the program. We're going to change the look of these costumes using the drawing palette.



Drawing palette

Step 2: While in the Costumes panel, select a favorite color and the "Color a Shape" bucket in the top right of the top right drawing menu.



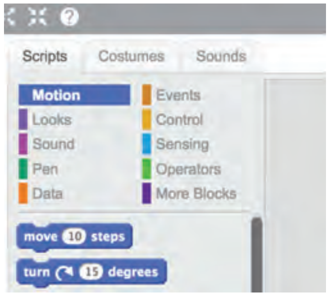
Step 3: Once you've selected the color and fill bucket option, select parts of the sprite to fill it in with the paint.

Great! You've changed your sprite's costume! Now it's time to make it move.

# The exercises

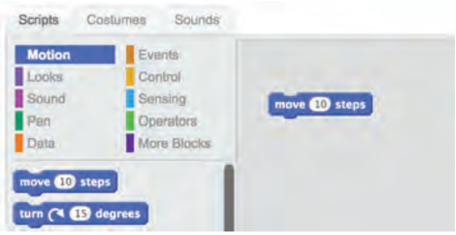
4 Make the sprite move.

Step 1: Go back to the main scripts panel and select the “motion” category. All of the blocks in “motion” are blocks that can make your sprite move!



```
move 10 steps
turn ↻ 15 degrees
```

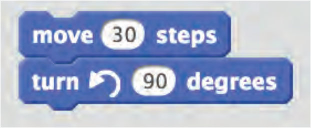
Step 2: Let’s start with a basic movement. Select the “move 10 steps” block and drag it to the scripts editor screen.



```
move 10 steps
turn ↻ 15 degrees
```

```
move 10 steps
```

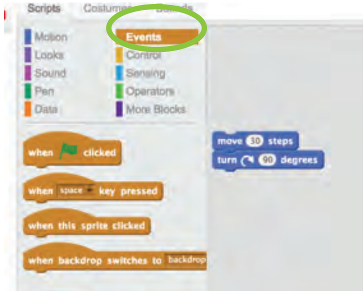
Step 3: Change 10 steps to 30 steps inside the block.



```
move 30 steps
turn ↶ 90 degrees
```

Step 4: Let’s have some fun and add additional blocks to get the sprite to turn left at 90 degrees. Drag the “turn right 15 degrees” block and change it to 90 degrees by clicking your cursor inside the block.

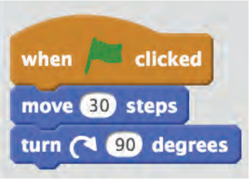
Step 4: In order to make the sprite move when you start the program, you’ll need to use an “event” block. Click on the “event” set of blocks.



```
when 🏁 clicked
when space key pressed
when this sprite clicked
when backdrop switches to backdrop
```

```
move 30 steps
turn ↻ 90 degrees
```

Step 5: The first block you’ll see is a green flag. Using this block sets the condition that when the green flag is clicked, something will happen. Let’s attach it to the script, or series of blocks, that we’ve created so far.



```
when 🏁 clicked
move 30 steps
turn ↻ 90 degrees
```

Step 6: Click on the green flag to move the sprite!

**TIP 1:** This is a great time to ask your child to explain what happened to see if they understand the different parts of the code.

**TIP 2:** Encourage your child to try out different motion blocks and see what happens.

**TIP 3:** Use other types of blocks to do something besides moving. You can attach a sound block to to make the sprite make a sound. Encourage your child to experiment!

# The exercises

## Exercise 2 - Create a card with your name

In this exercise, you and your child will create a greeting card on Scratch and personalize it.

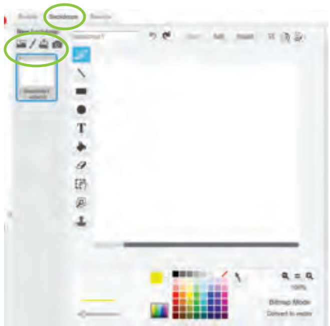
Now that you have moved a sprite, let's make something new!

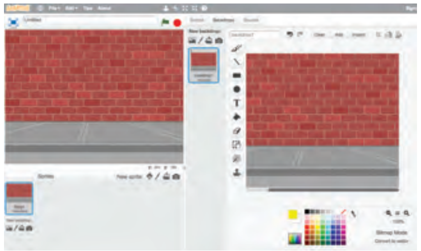
1. Plan out the card with your family.
   Step 1: Using a piece of paper and pen, have your child draw or write out what they would like to make for their card. You can change lots of things on a card, including adding a name, a background (like photos on your desktop, existing backgrounds in Scratch, or your own drawing in Scratch) and decorations (e.g. sprites, drawings, other graphics). Planning what you and your kids will make will allow them to better visualize the project.

2. Once you plan the card, open up Scratch and create a new project.

3. Select a backdrop.

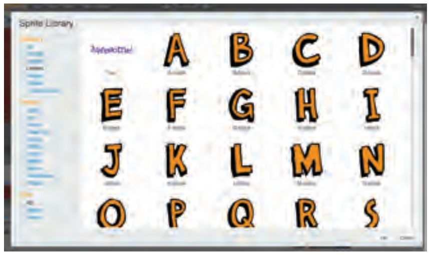
   Step 1: Have your child click on the backdrop panel. In the backdrop window, encourage them to draw, "add" an image from the Scratch library, or import an image from your computer. In this example, we chose a lovely brick wall.



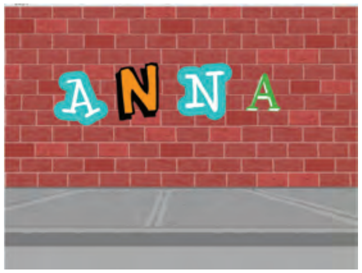


4. Add text sprites to represent your child's name.

   Step 1: In the "Sprites" window, you can click on the little elf character to add a new sprite. Click on it to open up the library of sprites which will have letter sprites.



   Step 2: Have your child add letter sprites that spell out his or her name.



   Step 3: Encourage them to add more sprites and their own decoration to the card.

Now that your child has created an awesome card, it's time to animate it!

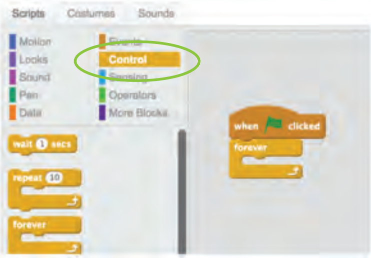
# The exercises

## Exercise 3 - Animate the card

In this exercise, you'll animate the text in the card. Through this animation exercise, you'll learn "loops." A loop is when you repeat a set of commands a specified number of times or forever (that's what animation is made of!).

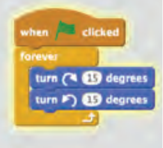
1. Double-click on one of the letters you would like to animate. We will choose the A sprite.
2. Place the event block you would like to use to start the script into the script editor. We will use the flag block to start the script.
3. We want to animate the A so it moves back and forth until we press the stop button. To do this, we will create a script that loops, which means that the set of commands in the script will repeat over and over again. Since we want it to run till we press the stop button, we need the loop to go on forever.

Step 1: Go to the "Control" set of blocks and drag the "forever" block to the "flag" block and snap it into place.



```
when clicked
forever
```

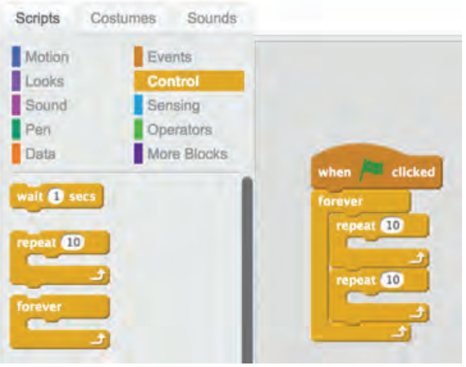
Step 2: To have it to move back and forth, we need to move the sprite to the left and to the right a few degrees. We will use the "turn left" and "turn right" motion blocks.



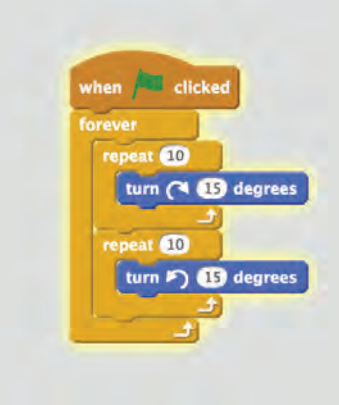
```
when clicked
forever
    turn ↻ 15 degrees
    turn ↺ 15 degrees
```

However, when you click the green flag, what happens? Nothing! Why? Your sprite is moving 15 degrees and simultaneously moving back 15 degrees at such a fast rate that it really doesn't make the letter look like it's moving.

Step 3: To keep the letter moving to the right before it moves to the left, you have to get it to repeat the right motion a few times for you to be able to see it, then start heading to the left a few times. To do that, insert "repeat" blocks around the blue motion blocks. Go to the "Control" set of blocks to get the "repeat" block and place the "turn right" and "turn left" blocks into the "repeat" blocks.



```
when clicked
forever
    repeat 10
    repeat 10
```



```
when clicked
forever
    repeat 10
        turn ↻ 15 degrees
    repeat 10
        turn ↺ 15 degrees
```

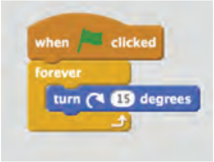
Now what happens when you click the green flag to start the program? It starts moving back and forth. Success! You've animated one of the letters.

**TIP 1:** Have your kid change the numbers in the "repeat" block and the "turn right" or "turn left" block. Have them observe what's happening to the letter. How far does it turn and how slow or fast is the animation?

# The exercises

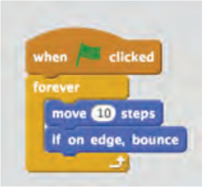
4 Animate the rest of the letters!

Double click on each of the sprites and create another script for them. To do this, you'll need to set up another "event" block with other "motion" blocks. Here are some examples:



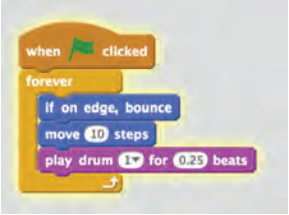
```
when ⚑ clicked
forever
    turn ↻ 15 degrees
```

a. Spin forever!



```
when ⚑ clicked
forever
    move 10 steps
    if on edge, bounce
```

b. Bounce everywhere!



```
when ⚑ clicked
forever
    if on edge, bounce
    move 10 steps
    play drum 1▾ for 0.25 beats
```

c. Make a really annoying drumming sound each time it moves and then bounce in another direction when it hits the wall!

**TIP 1:** Encourage your child to experiment. They can have the animations make noise as they move, change direction if it hits an edge, and more. Check out the "Looks" blocks, which can make the letters display text as it moves or the "Sensing" blocks, which could change the direction of the letters as it moves towards something on the screen.

**TIP 2:** Jump into the help area if your child is curious about each type of block and read through the descriptions to get a better understanding of what they do.

GREAT JOB!
YOU'RE DONE!

# Tips and tricks

## Think and plan

If your child wants to code a project, have them storyboard or write what they would like the program to do. Then have them explain it to you. For example, they can write out the text, choose what the background could look like and how they want each letter to move. Thinking and planning are key to helping plot out your project.

## Share and get feedback

One of the greatest resources when coding is each other. Have your child share his or her project with other kids through the Scratch online community or with family and friends.

## Show vs. tell

The best way to understand programming is to play with existing code, edit it, and see what happens. Find an existing program on Scratch, and encourage your child to change the code to see what happens. Let them build out new versions of what they see.

## Encourage interests

Programming allows kids to express their interests. They can bring in their own photos, images, and drawings. Common coding projects for kids include making simple games, art, stories, and animations.

## Hands off for you, hands on for them

Coding is best learned by doing it, so try to steer clear of writing the code, handling the computer, or building the project for your kids. Guide them as appropriate and allow them to take full reign if they are comfortable.

# Resources

## Schools and organizations

Here are a few schools and organizations specifically for children to learn coding:

**Code.org**: https://code.org/learn

**Girls Who Code**: https://girlswhocode.com/

**Black Girls Code**: http://www.blackgirlscode.com/

**Made with Code**: https://www.madewithcode.com/

**Google CS First**: https://www.cs-first.com/en/home

**Teaching Kids Programming**: http://teachingkidsprogramming.org/

**Khan Academy**: https://www.khanacademy.org/computing/computer-programming

**Hour of Code**: https://hourofcode.com/ie/

## Websites

Below are a few great sites that provide useful information and resources on how to teach coding to your kids.:

**7 Apps For Teaching Children Coding Skills**: https://www.edutopia.org/blog/7-apps-teaching-children-coding-anna-adam

**Best Apps and Websites for Learning Programming and Coding**: https://www.commonsense.org/education/top-picks/best-apps-and-websites-for-learning-programming-and-coding

**Edsurge's Guide to Teaching Kids to Code**: https://www.edsurge.com/research/guides/teaching-kids-to-code

## Coding platforms

There are many other coding platforms you could use other than Scratch. Here are a couple of them:

**Tynker**: https://www.tynker.com
Kids learn to code with visual blocks and text-based coding while designing games and apps.

**Hopscotch**: An app designed to allow beginner programmers to develop simple projects using a visual programming language.

**Kodu Game Lab**: https://www.kodugamelab.com/
Kids create games on the PC and XBox via a visual language.

**MIT App inventor**: http://appinventor.mit.edu/explore/
Program apps using  visual, drag-and-drop building blocks.

**Alice**: http://www.alice.org/
A programming environment that teaches coding concepts through creating animated movies and simple video.

**Wonder for Dash & Dot**: This toy set teaches kids how to program with toy robots that make coding fun using apps.

**Sphero**: A spherical robot that kids can code to do things.

**JewelBots**: Friendship bracelets that kids can program.

**Kano**: Kano is the all-in-one computer kit. Kids can build a computer and then learn to code with games and challenges.

**Raspberry Pi**: A tiny and affordable computer that you can use to learn programming through fun, practical projects.

**LEGO Mindstorms**: A series of kits contain software and hardware to create customizable, programmable robots.