

Block coding for beginners

Years 3-8



Overview

This activity assists teachers and students who are new to coding to get started with block coding or programming. Students can create simple code, choosing a digital sprite/character and using block instructions to make it move.

Curriculum links:

LEARNING AREAS	ACHIEVEMENT OBJECTIVES	LEVELS	YEARS
Technology: Designing and Developing Digital Outcomes (DDDO): Progress outcome 1	In authentic contexts and taking account of end-users, students participate in teacher-led activities to develop, manipulate, store, retrieve and share digital content in order to meet technological challenges. In doing so, they identify digital devices and their purposes and understand that humans make them. They know how to use some applications, they can identify the inputs and outputs of a system, and they understand that digital devices store content, which can be retrieved later.	2-4	3-8

Teacher information

Learning intentions

Students are learning to:

- become familiar with block coding using Scratch
- use block coding to program a character or sprite to move

Example success criteria

Students can:

- use Scratch to select a sprite and create a set of block coding instructions to move it

Resources needed

Internet access, digital device, Scratch website: <https://scratch.mit.edu>, Getting started with Scratch V2 Slideshow: https://docs.google.com/presentation/d/1-LU3rTvqwfgwa3eqrMrSzM_46FgOIHML0qYioxHLnrE/edit?usp=sharing

Vocabulary:

Coding, programming, Scratch, blocks, motion, event, sound, sprite, character, backdrop, stage, de-bug.

Background information and supporting resources

Block coding or programming

Block coding is a simple coding language suitable for introducing coding and programming to students at middle - upper primary school level. When we select blocks and drag and drop them into the coding space, they can be joined together to make sets of instructions. Block coding can form complex sets of instructions which can create movements, sounds, looks, and events for characters, images or hardware.

More complex coding languages can be more easily understood when a student is familiar with using blocks to code rather than text. This activity will focus on Scratch 3.0 and Tynker as these are free online websites. If your school has a subscription to an alternative coding website, or you use another free application, you can also use this activity and adapt the instructions with the terms and structure which is most appropriate.

But coding seems so complicated...

Coding can seem intimidating to people who haven't grown up with digital technology but block coding is easy once you have tried it a few times. Block coding provides creative opportunities for students to familiarise themselves with a simple programming language. Like any language, it takes time to learn, but with practise you will be able to code more complex instructions. Don't be afraid to try and see what happens when you put some coding blocks together. You can't go wrong, just drag the blocks back and try again.

Learning experience suggestions:

Block coding for beginners

Note: These are suggestions only and teachers are encouraged to adjust the activity to suit the needs and interests of their students.

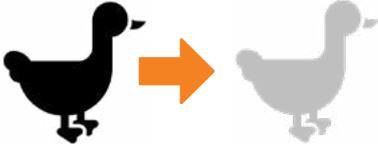
Introducing block coding

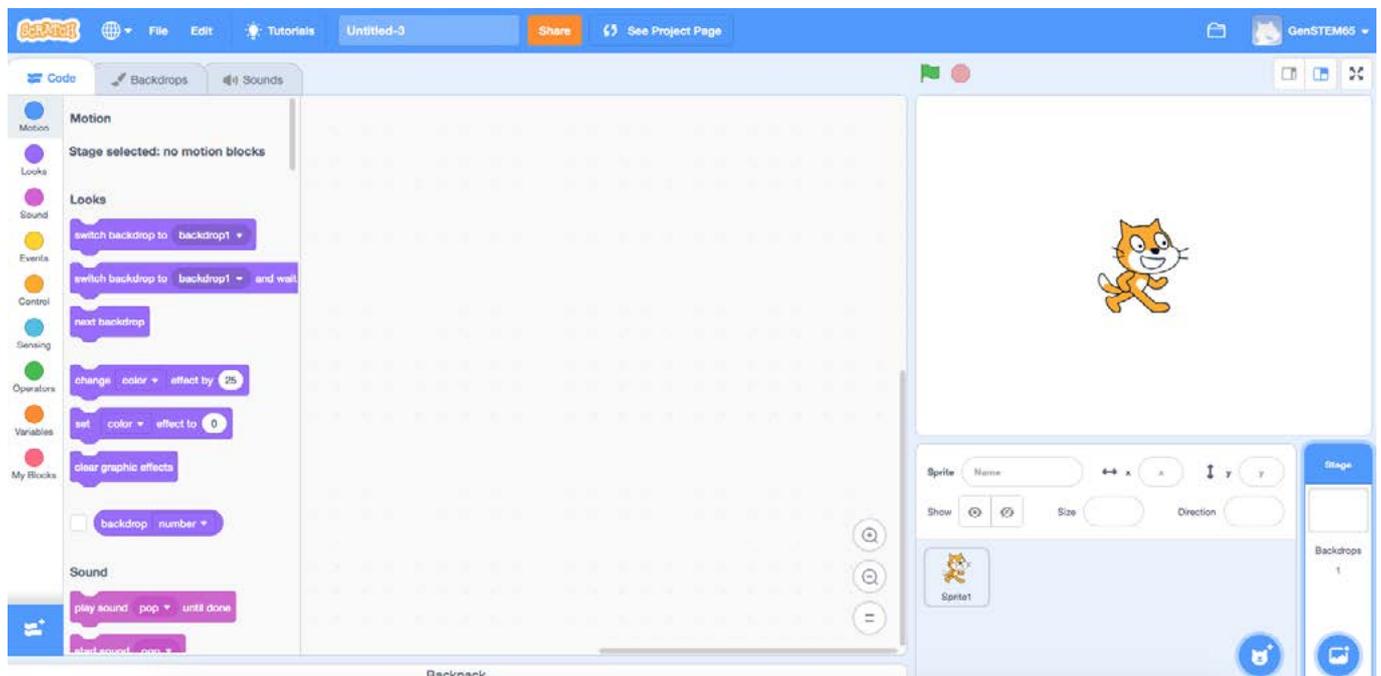
Allow approximately 15 minutes

- Identify students' prior knowledge and experience with coding. This activity is for those new to block coding. If your students are experienced block coders, consider an alternative activity.
- Block coding is like making a set of instructions for a character, sprite or image. The instructions must be in the correct order and can be movements, sounds, looks, operations or other factors. What would students like to learn about block coding? Set a goal for your session.
- View the *Getting started with Scratch V2* tutorial slideshow to learn how to use the Scratch Editor: https://docs.google.com/presentation/d/1-LU3rTvqwfqwa3eqrMrSzM_46FqOIHML0qYioxHLnrE/edit?usp=sharing
- While viewing the slideshow, point out to students the features of Scratch Editor, for example those shown on page 4.
- Hand out devices and connect to Wi-Fi. Students can open a web browser and go to the Scratch website: <https://scratch.mit.edu>. If you have internet connection issues, consider downloading Scratch onto your devices.
- Explore the Scratch 3.0 website <https://scratch.mit.edu> and practise selecting blocks and dragging and dropping them into the coding space. Make sure your blocks connect up and start with an 'event' block to begin the sequence of instructions.
- For some examples of America's Cup related race games made using Scratch, see: <https://scratch.mit.edu/projects/494892840>.

Getting to know the block coding workspace

Pointing out the following areas of the coding space on the Scratch website may help students new to coding. Use the *Getting started with Scratch V2* tutorial slideshow to guide discussion: https://docs.google.com/presentation/d/1-LU3rTvqwfqwa3eqrMrSzM_46FqOIHML0qYioxHLnrE/edit?usp=sharing

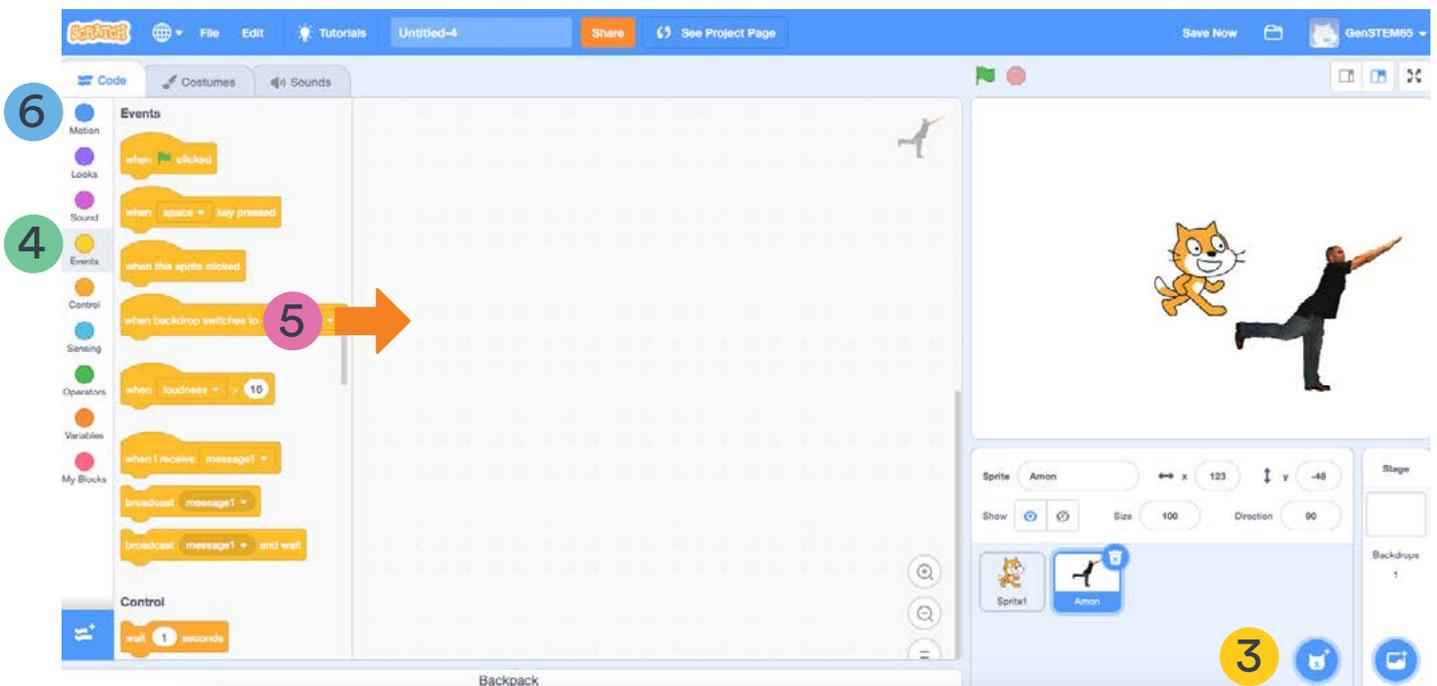
Block categories	Blocks Palette <i>Coding blocks (motion, events, etc...)</i>	Coding Area Workspace <i>(drag coding blocks here to make code)</i>	The Stage <i>Preview of what your code will look like</i>	
<ul style="list-style-type: none">  Events  Looks  Motion  Sound  Control 	    	    		
			Sprite details Position, hide/show	Stage
			Selected sprite	Backdrops 



Coding to make a sprite move

Allow approximately 15 minutes

Open your coding programme and create a new project	Select your sprite/actor or character	Make your sprite move
<p>Go to https://scratch.mit.edu/ (there are also great video tutorials here to help you in the <i>Tutorials</i> tab)</p> <ol style="list-style-type: none">1. Select 'create' from the menu on the top left of the screen.2. Name your project at the top of the screen and log in to save it. <p>Helpful hint: Log in to save your projects.</p> <p>You can also use Tynker: https://tynker.com or another program of choice. Instructions may vary.</p>	<ol style="list-style-type: none">3. Click on 'add sprite/actor' and select a character to use. See 3 below. Select the magnifying glass to view available sprites and click on the one you want to use. Scratch will put this sprite in your preview and select it to code.4. If you can't find a sprite that works for your animation/game you might want to make your own sprite. To make a new sprite, hover over the cat icon (see 3 below) and select the paint option. A paint canvas will come up to create your new sprite with.5. Delete unwanted sprites and select the one you want to write instructions for. <p>Helpful hint: <i>In Tynker sprites/characters are called actors.</i></p>	<ol style="list-style-type: none">6. Go to the events tab on the left of your screen. See 4 below. An event is when your code will start. Events are usually yellow or orange blocks with a 'hat' on top.7. Choose an event block such as 'when this sprite is clicked'. See 5 below. Drag this into your middle coding window (see orange arrow).8. Next, click on the motion circle on the left of your screen to bring up the blue motion blocks. See 6 below. Click on blocks to try them out.9. Drag a motion block under the 'when this sprite is clicked' block and join them together.10. Now your selected sprite will move when the event you have selected happens. Ka rawe, you're coding!



Next steps with coding

Try a Scratch tutorial or look on the ideas page: <https://scratch.mit.edu/ideas>

- Create more sets of instructions for your sprite. Choose a motion or sound block to add to an event block. Drag these blocks into the coding space and put the motion or sound block under the event block so that they fit together. For example, try a new set of instructions for when a user clicks the green flag by choosing the green flag event block and putting other instructions under that one. Click the green flag to start your code. The result of your coding will show on the right-hand side of your screen.
- Add another motion or movement tab to the coding area. Try a sound or look block to make a sound or change the look of your sprite.
- Click your sprite or event trigger to try out your code.
- Where did your sprite move to? What did it do? Is this what you intended? If the coding doesn't do what you wanted it to then there is a 'bug' in your code. Edit your code (drag blocks in and out of the coding window) until your sprite does what you want it to do. This is called de-bugging.

Review and reflect

- How did you go with coding your first program? Did you reach the goal you set at the beginning of the lesson?
- What went well? What was challenging? What are your next steps for learning? Record ideas for your next coding session.