RISING STARS

Learn to Code

Practice Book 2

Second Edition

TEACHER GUIDANCE

Help your pupils to become super-coders!

Introduction

This booklet supports Learn to Code Practice Book 2 Second Edition.

The practice book is broken into three sections, with one coding tool per section. In most cases, the activities within a section are sequential and should therefore be followed in order. Ensure children know how and where to save their work, where relevant, so they can quickly pick up from where they left off.

Try out the activities in the practice book before you give them to children. This is a great way to develop your own confidence in coding.

Contents

Scratch

Lightbot

w3schools

Published in association with:
HES
Dedicated to education

Tool: Scratch

https://scratch.mit.edu/download

Before you start: If you want to work offline, install the offline Scratch 3.0 editor on all the machines you will use in the lesson (see link above). Familiarise yourself with each activity and the progression from one to the next.

KS2 Computing objectives covered:

- Design, write and debug programs that accomplish specific goals.
- Use sequence, selection and repetition in programs.
- Work with variables and various forms of input and output.
- Use logical reasoning to explain how some simple algorithms work.
- Use software to design and create programs.

Activity 1: Programming using broadcast blocks (page 8)

How to use this activity

Activity 1 introduces the idea of using broadcast blocks to control when different events happen. Children will use them to build buttons that draw different shapes when the user clicks on them.

Tips

- Allow children plenty of time to familiarise themselves with the drawing tools that they will need to draw their own button sprites in step 1. The tools are very similar to drawing tools in other programs such as Word and PowerPoint.
- Using the *Set costume centre* tool to set the centre of the sprite as the tip of the pencil in step 4 is vital, so when the pencil sprite draws, it looks like the line is being drawn by the correct part of the pencil.
- It's useful to think of the broadcast blocks as blocks that allow 'invisible' messages to be sent between sprites. This whole activity is based on this idea. When one of the sprite buttons is clicked, it broadcasts a message to the pencil sprite to appear and then draw the corresponding shape.
- The repeat block, as used in steps 9 and 10, is really useful when creating scripts to draw regular shapes, as the same sides and angles are repeated several times over.

Assessment

All children should be able to:

- create a script that creates buttons that draw different shapes when clicked.

Most children should be able to:

- add in *set pen color to…* blocks to program the colour of the shapes drawn.

Some children should be able to:

- create additional scripts to program buttons that draw other shapes such as equilateral triangles and regular hexagons.

Activity 2: Programming a maths quiz (page 12)

How to use this activity

This activity provides a useful way of securing children's knowledge of the concept of selection, and programming using conditional statements. Children will be programming a maths quiz that tells the user when their answer was right or wrong.

Tips

- Using *if … then* statements is useful for explaining the idea of selection and conditions, e.g. **if** the user gives the correct answer **then** they will get *this* response; **if** the user gives an incorrect answer **then** *this* will happen.
- *Ask … and wait* blocks are introduced in step 3. Using these blocks prompts a sprite to ask the user the question that is typed into the block and then waits for the user to type in their answer.
- The *if … then* blocks that are introduced in step 4 instruct the program what the correct answer is and what events should follow if this is given, as well as what should happen if any other answer is given. Remind children that their spelling must be correct, as whatever they type into the *if* block is what Scratch will accept as a correct answer. It is also case sensitive.

Assessment

All children should be able to:

- create a script that builds a maths quiz that tells the user if their answer was correct.

Most children should be able to:

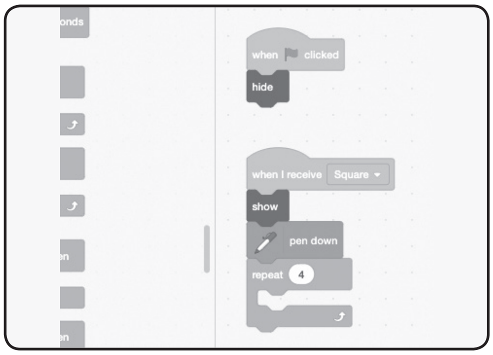
- adjust the script to change the questions and answers.

Some children should be able to:

- program the sprite to turn green if the correct answer is given and red if any other answer is given.

About Scratch

Scratch is a visual programming language that allows users to create their own animations, games and interactive stories. Its ‘drag and drop’ interface makes it a useful tool for programming in the primary phase. Users are able to program objects, called sprites, by building scripts that comprise blocks of commands. The command blocks are grouped into 10 categories: *Motion*, *Looks*, *Sound*, *Pen*, *Data*, *Events, Control, Sensing, Operators* and *More Blocks*. When a script is run, the commands are followed in the order in which the blocks have been arranged.



```
when ⚑ clicked
hide

when I receive Square ▾
show
pen down
repeat 4
```

Activity 3: Using variables (page 16)

How to use this activity

Use this activity to further reinforce children’s understanding that variables are data that can change or be changed. The variables used in this activity are common ones: a time limit and score in a game. Both of these elements will change depending on how much time has passed and how well the user is doing. Children will create a game where the user has to recognise as many multiples of 5 as they can within a set time.

Tips

- Two backdrops will be used in this game. The script will switch between them to show when the time is up and when the game is over.
- The balloons that have multiples of 5 on them – the ones the user should click – will be programmed to make a popping sound and disappear when they are clicked. The balloons with other numbers on will make a rattling sound signalling this was an incorrect selection, and will turn red.
- To save time, children can program one of each type of balloon sprite and then duplicate it and just change the number on it. When a sprite is duplicated, so are all of its associated scripts.

Assessment

All children should be able to:

- create a script for a multiples maths game that has a time limit and keeps track of the score.

Most children should be able to:

- adjust the script to increase the score for each correct selection.

Some children should be able to:

- add more balloons to the game to make it more complex.

Activity 4: Using random variables (page 20)

How to use this activity

Use this activity to build on the idea of variables, which were used in activity 3. This activity will use random variables to create a quiz that generates questions at random. The game will have two sprites, each of which will display a random number for the user to add together.

Tips

- The *pick random* operator, as seen in step 4, allows you to define the minimum and maximum numbers of the range you want a sprite to select from.
- New numbers will be randomly generated every time the space bar is pressed.

Assessment

All children should be able to:

- create a maths game that generates two random numbers for the user to add together.

Most children should be able to:

- adjust the code so three-digit numbers are randomly selected.

Some children should be able to:

- improve the interface so the purpose of the game is clearer.

Tool: Lightbot

bit.ly/1iVemu6

Before you start: Check that you have internet access on all computers to be used for this activity. The link above provides an advert-free version of Lightbot. Ensure that this web link is not blocked.

KS2 Computing objectives covered:

- Design, write and debug programs that accomplish specific goals.
- Solve problems by decomposing them into smaller parts.
- Use sequence, selection and repetition in programs.
- Use logical reasoning to explain how some simple algorithms work, and to detect and correct errors in algorithms and programs.

Activity 5: Programming a robot to move (page 24)

How to use this activity

Use this activity to familiarise children with the programming interface of Lightbot: how to build different grids as well as the different basic commands and how to run programs. A command is an instruction given to an object or character to make something happen. In this activity, children will learn how to use commands to program Lightbot to move forward, turn and jump.

Tips

- The *Level editor* allows children to create their own grid for Lightbot to solve rather than using a ready-made one. This will reinforce their understanding when it comes to using different commands to program Lightbot successfully.
- A grid is only completed when Lightbot has lit all the blue tiles.
- The grid built in this activity will require children to use basic movement commands to complete it: *walk forward*, *jump*, *turn clockwise*, *turn anticlockwise* and *light up tile*.
- Once a grid is built, children must click on *Test* to enter play mode, where they can program Lightbot to complete the grid they have built.
- Commands must be dragged and dropped into the *Main method* area and then run to test the program.

Assessment

All children should be able to:

- create a grid for Lightbot to move around on and then sequence a set of commands to program him to light up the blue tile.

Most children should be able to:

- change the order of the commands so Lightbot walks along the wall before jumping off it.

Some children should be able to:

- find several different solutions to complete the level and program Lightbot to reach the blue tile.

Activity 6: Using functions (page 28)

How to use this activity

Use this activity to further children's understanding of using repetition in programming. The *Main method* box in Lightbot has spaces for only 16 commands. As the grids get more complex, more spaces than this are needed to complete them. The way to do this is to create functions, separate sub-programs, which can then be repeated in the *Main method* program.

Tips

- It's worth reminding children that repetition is an important aspect of programming and it means to repeat an instruction a number of times.
- A function is a small section of code, which in Lightbot is separate from the *Main method* code, and which returns a result or answer.
- Children create a grid with three identical sides because their final program will repeat the same set of instructions three times.
- All the commands in the *Function 1* box can be repeated in the main program by dropping in an *F1* tile.

Assessment

All children should be able to:

- create a grid for Lightbot and a program that includes repeating a function.

Most children should be able to:

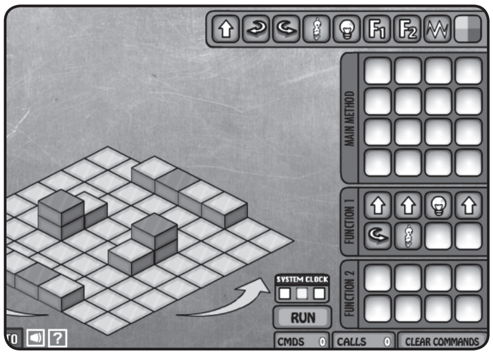
- adjust the grid to remove the raised tiles and then change the sequence in *Function 1* so it still works.

Some children should be able to:

- create a grid with four blue tiles to light up and four *F1* commands in the program.

About Lightbot

Lightbot is a free online tool that requires users to solve puzzles using programming logic. Lightbot is a small robot who moves around worlds made of grids. Users must program Lightbot to light up all the blue tiles in a grid. He can be programmed by dragging and dropping command blocks. Using Lightbot 2.0 enables users to build their own Lightbot grids as well as solve puzzles on ready-made ones. The activities all involve building grids as well as programming Lightbot to light up the blue tiles.



Activity 7: Using more than one function (page 32)

How to use this activity

Use this activity to build on children's understanding of repeating functions within a program.

Tips

- In the picture in step 3, you can see the grid has two sets of opposite but identical sides. This means children can create a function to guide Lightbot along each side and then repeat it.
- Wherever Lightbot is placed on the grid, prior to clicking *Run*, this is where he will start the game in play mode.
- If children are finding the activity difficult, ask them to imagine they are Lightbot and view the grid through his eyes, to help them work out what their next command should be.

Assessment

All children should be able to:

- program Lightbot to light up the tiles by repeating two different functions.

Most children should be able to:

- change part of the grid and alter their code so it can still be completed.

Some children should be able to:

- create and build their own grid and program Lightbot to complete it using two functions.

Activity 8: Coding with conditions (page 36)

How to use this activity

Use this final activity to reinforce children's understanding of the idea of selection and using conditions in their code.

Tips

- Selection is when different outcomes or events occur depending on what option was chosen. Using **if … then** statements helps to explain this idea, e.g. **if** Lightbot lands on *this* tile, **then** *this* will happen.
- In step 3, children are encouraged to incorporate their learning about repetition and functions into their code for this activity.
- Adding in coloured tiles allows you to change Lightbot's colour. This allows you to set up conditions, e.g. the *light up tile* command will only be followed if Lightbot is coloured purple. He will turn purple by lighting up a purple tile (this works in the same way with the orange tiles).
- In step 6, children set some commands before running the program. This means these commands will be locked and set in place, ready for the user in play mode.

Assessment

All children should be able to:

- program Lightbot to follow commands depending on what colour he is.

Most children should be able to:

- adjust the procedure so Lightbot only lights up the blue tiles if he is orange.

Some children should be able to:

- create their own grid that uses more than one condition in the program.

Tool: w3schools

w3schools.com

KS2 Computing objectives covered:

- Work with various forms of input and output.
- Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication.
- Use a variety of software to design and create a range of content that accomplish given goals, including presenting data and information.

Before you start: Familiarise yourself with w3schools and the different activities. Each one builds on the previous one so children will need to complete them in the given order.

Activity 9: Structuring a web page using HTML5 tags (page 40)

How to use this activity

This activity will teach the basic rules of how HTML5 is structured. Children will begin to build their own web page informing parents about their school sports day.

Tips

- HTML stands for Hypertext Markup Language and is the code used to build web pages. Web browsers interpret this code and display it as web pages.
- You can code how you want different elements on a web page to look by using different HTML tags. Each tag is made up of an opening and closing tag, e.g. <strong>computers </strong>. A web browser will interpret this code and display the word 'computers' in bold.
- Children will need to copy and paste their code and save it in a Word document or similar so they can paste it back into w3schools for the following activities.

Assessment

All children should be able to:

- create a structured web page with different levels of text in it.

Most children should be able to:

- display certain text in italics.

Some children should be able to:

- adjust the code to change the colour of different elements.

Activity 10: Creating hyperlinks using HTML5 (page 44)

How to use this activity

Use this activity to reinforce children's knowledge of networks through the creation of hyperlinks. Using hyperlinks allows different web pages to be linked to one another. This forms the basis of how the internet works.

Tips

- Children will need to copy and paste the code they saved in a Word document in the last activity into the w3schools editor so they can build links into it.
- In step 3, children are introduced to the unordered list tag, which creates lists using bullet points.
- Make sure children copy, paste and save their updated code as before.

Assessment

All children should be able to:

- add hyperlinks to other websites into their web page.

Most children should be able to:

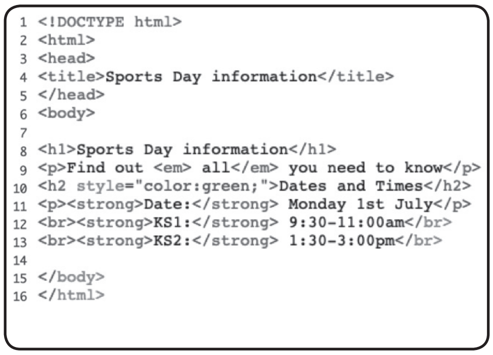
- adjust the code to change one of the links.

Some children should be able to:

- add in more links of their own.

About w3schools

w3schools is an online tool that makes it very easy to create your own web pages. It has two main windows: a visual editor, where you can edit HTML5 code, and a preview window where you can see the resulting web page based on your code.



```
<!DOCTYPE html>
<html>
<head>
<title>Sports Day information</title>
</head>
<body>

<h1>Sports Day information</h1>
<p>Find out <em> all</em> you need to know</p>
<h2 style="color:green;">Dates and Times</h2>
<p><strong>Date:</strong> Monday 1st July</p>
<br><strong>KS1:</strong> 9:30-11:00am</br>
<br><strong>KS2:</strong> 1:30-3:00pm</br>

</body>
</html>
```

Activity 11: Embedding content using HTML5 (page 48)

How to use this activity

This activity shows children how to add images and a Google map to their web page. It builds on the previous two activities.

Tips

- The image used in step 2 is copyright free and suitable for use. This is an ideal point to teach children about copyright and what images are OK to use as part of using technology responsibly and respectfully.
- For step 5, children need to be clear how to open a new tab so they can switch between Google Maps and their w3schools page easily.
- In step 5, children are instructed to put in their school's postcode to create the Google map for their website, but this should be done in line with your school's safety policy. Children must always be careful when sharing their location, even if it's their school's location, online.
- Ensure children remember to credit the source of their maps accordingly. See credits at the bottom of visuals on pages 49–50 of the Practice Book.
- As before, the code from this activity will need saving in a Word document or similar for use in the final activity.

Assessment

All children should be able to:

- add an image and a Google map to their web page.

Most children should be able to:

- adjust the size of the image and the Google map.

Some children should be able to:

- add other appropriate images.

Activity 12: Creating tables using HTML5 (page 52)

How to use this activity

This activity finishes the sports day web page by showing children how to create a table for the order of events. It is based on the work done in the previous three activities.

Tips

- Building a table requires several different tags so it is worth getting children to swap with a partner and check each other's code for possible errors.
- In step 10, children are shown how to format the type of grid lines they want in their table. The unit of measure px stands for pixels and indicates how thick the grid lines should be.

Assessment

All children should be able to:

- add a table to their web page.

Most children should be able to:

- adjust the code to change the colour and thickness of the grid lines.

Some children should be able to:

- add extra columns or rows into their table.

Glossary

- **Broadcast block:** allows you to send a message in Scratch to activate other commands or scripts.
- **Closing tag:** this tells the web browser where a tag ends. The closing tag must include a forward slash.
- **Code:** the commands or program you write using a programming language.
- **Command:** an instruction given to an object or character to make something happen.
- **Condition:** something that is true or false, e.g. *Number 6 is in the box. True or false?*
- **Data:** information that can be stored, collected or changed.
- **Debug:** to find and get rid of errors within code.
- **Formatting:** to change the style of text, e.g. alignment, spacing and bold.
- **Function:** a section of code that makes a specific process happen, and importantly returns a result (or answer).
- **HTML:** Hypertext Markup Language – the language used to create web pages.
- **Hyperlink:** a link that takes a user to another document or web page, or to another part of a document.
- **<iframe>:** the HTML tag for a frame in a web page that can contain an image or another web page.
- **<img>:** the HTML tag for an image within a web page.
- **Input:** information given to a computer to make something happen (e.g. a mouse click or button press).
- **Opening tag:** this tells the web browser where a tag starts.
- **Output:** something that a computer produces when given an instruction, e.g. an on-screen image, a sound or vibration.
- **Procedure:** a small section of code that makes a specific process happen.
- **Program:**
  1. a sequence of instructions to perform a task or solve a problem, using programming language (noun)
  2. to create or change a program (verb).
- **Repeat:** take a certain command (or set of commands) and follow the instruction again (and again).
- **Script:** a set of commands that are followed by a program.
- **Selection:**
  1. the act of selecting something
  2. one thing or a group of things that have been selected.
- **Sequence:** a set of commands that run one after another.
- **Sprite:** an object or character that can be programmed.
- **Tag:** code used to label the content in an HTML document.
- **Unordered list:** this <ul> tag allows you to create bulleted lists in HTML.
- **Variable:** a piece of data that can be changed to produce a different result.

Rising Stars, part of Hodder & Stoughton Ltd, Carmelite House,
50 Victoria Embankment, London, EC4Y 0DZ

www.risingstars-uk.com

Author: Claire Lotriet

Second edition updated by Adam Chase

First published 2015, updated 2020

page 3: screenshot from Scratch http://scratch.mit.edu licensed under Creative Commons licence. Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab; page 5: screenshot from Light Bot 2.0 www.light-bot.com; page 7: screenshot from w3schools