RISING STARS

Learn to Code

Practice Book 1

TEACHER GUIDANCE

Help your pupils to become super-coders!

Introduction

This booklet supports Learn to Code Practice Book 1.

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

PowerPoint

Excel

Published in association with:

HES
Dedicated to education

Tool: Scratch

scratch.mit.edu

Before you start: Ensure that Scratch 3.0 is installed on all computers or that children have access to Scratch 3.0 online. Try out the activities to get a feel for how they progress and where to locate the different tools used.

KS2 Computing objectives covered:

- Design, write and debug programs that accomplish specific goals.
- Use sequence and repetition in programs.
- Work with different forms of input and output.
- Use software to create a range of programs that accomplish given goals.

Activity 1: Programming a sprite (page 8)

How to use this activity

Use this activity to get children familiar with the Scratch interface and the basic method of programming (dragging and dropping). You can also introduce the concept of sequence in programs.

Tips

- The cat sprite is the default sprite that is in place whenever you start a new Scratch project. Allow children plenty of time to explore the large library of possible sprites. They can also create their own sprite using the *Paint new sprite* option or turn themselves into sprites using the *New sprite from camera* feature if what they want doesn't already exist.
- Like the sprites, there is a whole host of backdrops in the library and you can also paint your own, upload an image you have saved on your computer or take a photo to use as a backdrop to suit your own needs.
- To remove any block from the scripts area at any time, all you need to do is right-click on it and select *delete*, or simply drag and drop it back into any of the block palettes.
- A genuine understanding of how the *glide* command works requires knowledge of *x* and *y* coordinates, so it might be useful to tackle this with children before running the activity.
- The *say* command block doesn't play a sound, but instead instructs the sprite to display a speech bubble; this is why it is found in the *Looks* category and not the *Sound* tab.

Assessment

All children should be able to:

- sequence scripting blocks to make a simple animation.

Most children should be able to:

- change the speed at which the sprite moves on screen.

Some children should be able to:

- adjust the script so the sprite moves between four different points.

Activity 2: Programming a sprite using repetition (page 12)

How to use this activity

Use this activity to introduce the idea of repetition in programming through the use of costumes and a *forever* loop.

Tips

- Some sprites have more than one costume. Costumes are different versions of the same sprite but in different positions. When a sprite is programmed to switch between two costumes continually, it can create a more realistic animated effect (for example, walking). By switching between the two octopus costumes it will look like the octopus is swimming on the screen.
- The *forever* block is a *Control* block. Any blocks that are placed within a *forever* block will be in an infinite loop, which means the commands will be followed repeatedly until the script is stopped.
- You can create multiple scripts for each sprite, depending on what you want the sprite to do.
- Infinite loops are really useful commands for programming background sprites in an animation such as birds flying about in the sky.

Assessment

All children should be able to:

- use the *forever* command block to program a sprite to move repeatedly.

Most children should be able to:

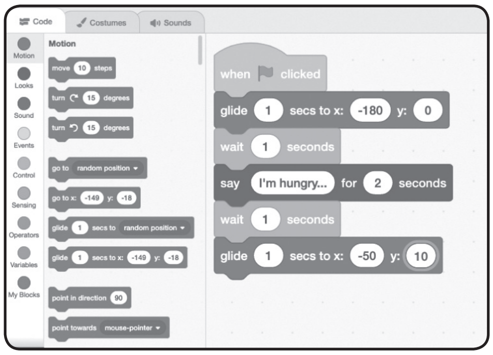
- adjust the *wait… secs* or *glide…* blocks to program the sprite to move faster or slower.

Some children should be able to:

- adjust the coordinates to change the direction in which the sprite moves.

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 are arranged.



```
when ⚑ clicked
glide 1 secs to x: -180 y: 0
wait 1 seconds
say I'm hungry... for 2 seconds
wait 1 seconds
glide 1 secs to x: -50 y: 10
```

Activity 3: Executing actions in different ways (page 16)

How to use this activity

Use this activity to introduce the idea of the programming concept of selection: **if** this happens **then** do this. For example, **if** the green flag is clicked, **then** do this…, or **if** the space bar is pressed, **then** do this…

Tips

- Children could substitute the penguin sprite for another sprite, with two costumes, and pick a background that matches its likely habitat.
- Children can use other blocks in the *Events* palette to start a script. For example, instead of using the *when* ⚑ *clicked*, certain scripts can be programmed to run when different keys are pressed. At this point, the animations the children are programming start to take on an interactive element: the viewer can make things happen.

Assessment

All children should be able to:

- program a sprite to change colour when the space bar is pressed.

Most children should be able to:

- adjust the script so the sprite changes colour when a different key is pressed.

Some children should be able to:

- add a fourth script so the sprite plays a sound when another key is pressed.

Activity 4: Using time to sequence events (page 20)

How to use this activity

Use this activity to consolidate learning from the previous three activities.

Tips

- You could ask children to prepare a simple script of a conversation between two sprites before they start programming their animations. Encourage children to read the script (each taking on the role of one of the sprites) and then consider how they could make this happen in Scratch.
- It’s important that children remember to click on the sprite they want to program so they can access that sprite’s script area.
- The *wait* blocks allow the conversation between the two sprites to work in this activity. The right amount of time needs to be programmed between each block so that they appear in the right order, the conversation makes sense and the actions happen at the right moment.

Assessment

All children should be able to:

- use *wait* blocks to program a conversation between two sprites.

Most children should be able to:

- add some more *say…* and *wait…* blocks to extend the conversation.

Some children should be able to:

- add a third sprite to the conversation and adjust the script accordingly.

Tool: PowerPoint

Before you start: Ensure PowerPoint is installed on all computers. Try the activities to get a feel for how they progress and where to locate the different tools used.

KS2 Computing objectives covered:

- Use sequence and selection in programs.
- Use logical reasoning to explain how some simple algorithms work, and to detect and correct errors in algorithms.
- Use a variety of software to design and create content that accomplish given goals, including presenting data and information.

Activity 5: Sequencing an animation (page 24)

How to use this activity

Use this activity to introduce or develop the concept of sequence. Sequence is all about carrying out actions or instructions in a particular order to ensure a desired outcome. In this activity, children will sequence the process of plant growth to create an animation. Each part of the animation will be on a different slide. When they are played in sequence the process will be animated.

Tips

- Ideally, children will have some prior knowledge of PowerPoint before getting started on this activity, particularly how to use the drawing tools and how to insert simple shapes.
- The *Unplugged* activity on page 27 might be a useful way to start the session and establish the idea of sequencing: *Draw a sequence of pictures that show the full life cycle of a plant. Mix them up and challenge a partner to put them in order. How did they know what order to put them in?* This will also ensure they are familiar with the science behind the animation.
- This animation works because each new slide is based on the previous slide, with a slight change made. This makes the duplicate tool really useful; children don't need to start each slide from scratch.
- Ensure children know how and where to save their PowerPoint animation as they will need it for the following activities.

Assessment

All children should be able to:

- create a sequence of slides that shows the process of a plant growing.

Most children should be able to:

- add some more detail in the background, such as birds in the sky, that change slightly throughout the animation.

Some children should be able to:

- create a similar animation for a different plant or flower.

Activity 6: Creating an animation on one slide (page 28)

How to use this activity

Use this activity to introduce children to a different way of animating using PowerPoint. This time, their animation will take place on a single slide and will be powered by animation effects and motion paths. They will still consider the sequence of the animation effects they use to make sure different events take place in the right order. This animation will show how bees pollinate flowers. Each bee will move by following a different motion path (like an invisible path).

Tips

- Children will need to complete activity 5 before embarking on this one.
- Children will base the one-slide animation in this activity on the last slide of the animation they made in activity 5. They will need to know how to use *Save as* to avoid saving over the top of their original animation.
- It would also be useful if children had some knowledge of selecting and grouping together several objects at once as this will make the drawing process much easier; children will be able to copy and paste elements of the animation, such as the flower and the bees, allowing them to focus on the sequencing rather than the drawing.
- Allow children plenty of time to explore the different animation effect options.
- Make sure children know where to save their work.

Assessment

All children should be able to:

- create an animation on one slide using motion paths and animation effects.

Most children should be able to:

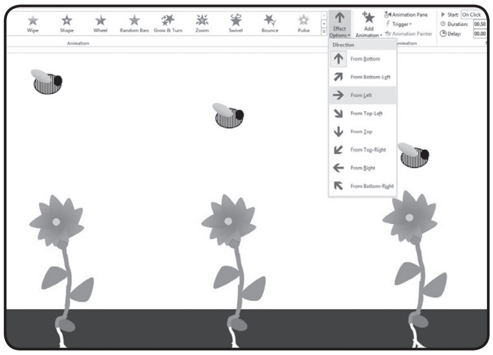
- adjust the order of the different animations.

Some children should be able to:

- add other objects, such as more bees moving between the flowers.

About PowerPoint

Microsoft PowerPoint 2013 is used here; options may differ in other versions. PowerPoint is a presentation program that is traditionally used to create slideshows for presentations. It also allows the user to create hyperlinks to different slides within a presentation and contains a host of animation tools, making it ideal for creating animations and teaching children about the basics of sequence: a key programming concept.



Start: On Click
Duration: 00.50
Delay: 00.00

Activity 7: Creating an interactive animation (page 32)

How to use this activity

This animation introduces the idea of interactivity: parts of the slide will be animated when clicked. This animation will show the different things a plant needs to grow: sunlight, nutrients in the soil and water. Each will become animated when it is clicked. Children will use triggers to do this. Using triggers is an ideal context in which to talk about input and output. Input is data going into a computer or program – in this case the user clicking on an image – and output is data or information produced by a program – in this case an animated image.

Tips

- Children will need to have completed activity 6 beforehand as this activity will be based on that work.
- As before, children will need to know how to use *Save as* to avoid saving over the top of their work from activity 6.
- As before, it will be useful if children already have a secure knowledge of how to create different shapes in PowerPoint so the focus of the activity is on setting triggers rather than the drawing.

Assessment

All children should be able to:

- create an interactive animation on one slide using triggers.

Most children should be able to:

- adjust the custom animation repeat settings so some animations continue and don’t stop after one click.

Some children should be able to:

- add in another element, such as a rainbow, that appears after one of the other animations has finished.

Activity 8: Using hyperlinks in slides (page 36)

How to use this activity

Use this activity to introduce children to a different take on sequencing: non-linear structures. PowerPoint allows you to hyperlink different slides to one another so you don’t automatically have to follow the one before. This hyperlinking means that the user can choose which slide to go to next. This idea will be used to create a flower designer. Users will choose different elements of a flower, such as petal shape and petal colour, at each stage of the animation to design their own flower. This reinforces the idea of selection: the ability to make choices, which result in different outcomes.

Tips

- Creating a tree diagram (similar to the one suggested in the *Unplugged* activity on page 39) would be a useful starting point so children can see what text should link to what slide.
- You could relate the hyperlinks to hyperlinks on web pages and begin to give children an understanding of how the internet and other networks work.
- It would be useful if children had some knowledge of how to insert a simple table before they begin this activity.

Assessment

All children should be able to:

- create a non-linear slideshow that allows users to design their own flower.

Most children should be able to:

- add a button on each end slide that takes the user back to the starting slide.

Some children should be able to:

- add a fourth petal colour option and update their slideshow and links accordingly.

Tool: Excel

Before you start: Ensure that Excel is installed on all computers. Give yourself plenty of time to work your way through the activities and familiarise yourself with Microsoft Excel and all the functions needed for these four activities.

KS2 Computing objectives covered:

- Design, write and debug programs that accomplish specific goals, including simulating physical systems.
- Use selection in programs; work with variables and various forms of input and output.
- Use a variety of software on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including analysing data.

Activity 9: Using the random function (page 40)

How to use this activity

Use this activity to show children how using computers to simulate events can be very useful.

Tips

- To introduce the activity, you may want to ask children to start rolling a die several times over, and to record the results, so they get a feel for why using computers to simulate such events can be very useful.
- The F9 key is used throughout these activities to repeat functions, e.g. step 3 in activity 9. Be aware that this feature is only available in the latest versions of Microsoft Excel. To achieve the same effect, click in the formula bar and press Enter again (as explained in the activity).
- Functions and formulae must be typed accurately for the code to work so encourage children to check this carefully or to work in pairs to check each others' typing.
- The actions in step 6 allow children to quickly copy and paste the function into 100 cells in one go.
- The formula for simulating a die throw can be repeated to simulate more dice: each RANDBETWEEN(1,6) function represents one die, so two dice would be RANDBETWEEN (1,6)+RANDBETWEEN(1,6) and so on.
- The subsequent activities in this section build on this one so ensure children know how to save their Excel file in an appropriate place so they can access it for activities 10–12.

Assessment

All children should be able to:

- use the random function to simulate 100 die throws.

Most children should be able to:

- adjust their spreadsheet slightly to simulate 200 throws.

Some children should be able to:

- change the formula to simulate two dice thrown at once.

Activity 10: Using conditional formatting (page 44)

How to use this activity

This activity is an ideal opportunity to introduce or reinforce the idea of selection: a key programming concept. Selection is really based on choice: **if** one option is chosen, **then** a certain outcome will follow. This **if** and **then** structure is the basis for selection. This activity uses this idea to colour-code the different throws of the die: **if** a 1 is 'thrown', **then** it will be coloured purple; **if** a 2 is 'thrown', **then** it will be coloured dark blue, and so on.

Tips

- Ensure all children know where to find the 100 die throws spreadsheet from activity 9 as they will need this for this activity.
- Make sure they know how and where to save their work as they will need their completed speadsheet in the final two activities.
- Children can use different colours to those suggested in the activity; the important thing is to ensure that each cell is formatted to display a different colour.

Assessment

All children should be able to:

- use conditional formatting to display different scores in different colours.

Most children should be able to:

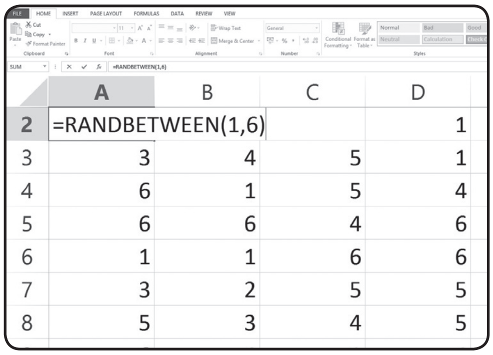
- change the conditional formatting rules so the colour of the text changes instead of the fill colour.

Some children should be able to:

- add additional formatting rules so that scores of 6 only have white text.

About Excel

Microsoft Excel 2013 is used here; options may differ in other versions. Excel is spreadsheet software that allows a user to create tables and to calculate and analyse data using formulae and functions. In Excel, rather than creating a sequence of instructions, you write code (Excel formulae) to create a system of interlinked functions, which take values in the spreadsheet cells and return the results of those functions. This can be useful when a large amount of numerical data needs to be processed or where mathematical functions provide a good way to model a complex real-world problem.



=RANDBETWEEN(1,6)

	A	B	C	D
2	=RANDBETWEEN(1,6)			1
3	3	4	5	1
4	6	1	5	4
5	6	6	4	6
6	1	1	6	6
7	3	2	5	5
8	5	3	4	5

Activity 11: Using functions to analyse data (page 48)

How to use this activity

The aim of this activity is to begin to analyse the data produced from the 100 die 'throws'. Use it to support and extend maths work on data handling.

Tips

- In this activity, children will use formulae to calculate the average (mean), median and mode of the 100 simulated die 'throws' so it is worth explaining these terms beforehand. Children can complete this activity without these explanations but may need additional support throughout.
- Children will be calculating the sum of all the scores so check they understand the term 'sum' beforehand.
- To help children understand why it is useful to program computers to analyse data rather than doing this by hand, set some similar calculations first.
- In step 5, children create a formula where they divide the sum total by 100 to calculate the mean average. If some children went on to simulate 200 die throws in activity 10, they will need to change this to 200.
- To add the total of the first 50 throws, change the SUM function to =SUM(A2:E11) or =SUM(F2:J11) for the final 50.
- To calculate the average of the first 10 throws, you could use the SUM function to add the first 10 scores and then divide this by 10. The same could be done for the last 10.

Assessment

All children should be able to:

- create functions to calculate the total, mean, median and mode of the 100 die throws.

Most children should be able to:

- adjust the SUM function so it only adds up the first 50 throws.

Some children should be able to:

- add two more AVERAGE functions to calculate the average of the first 10 and final 10 throws so they can compare them.

Activity 12: Using the COUNTIF function and making a bar chart (page 52)

How to use this activity

Use this activity to support work on graphs and frequency tables, and to introduce or build on children's knowledge of variables: information that can change.

Tips

- The first thing children will do in this activity is create a frequency table so it would be useful if they had some knowledge of this and had created one themselves by hand.
- In step 3, children are introduced to the COUNTIF function, which will be used to build their frequency table. In this instance, it enables us to instruct Excel to count all the die 'throws' with a score of 1 and put the total in a cell. A different COUNTIF function will be used for each throw up to 6. These six numbers generated by the COUNTIF function are variables because the number of each score thrown will change each time a new 100 throws are generated.
- To change the colours of the bars in the bar chart, click on a bar once to select all the bars, then click on the bar you want to change again. Right-click on it and use the *Fill colour* tool to change the colour.
- To apply conditional formatting to the frequency table, highlight the six cells containing the frequencies and then click on the *Conditional Formatting* button. Select *Highlight Cells Rules* in the drop-down menu and click on *Greater Than….* Type '20' into the first box and select *Red Text* in the second box.

Assessment

All children should be able to:

- use the COUNTIF function to create a frequency table.

Most children should be able to:

- create a bar graph based on the frequency table.

Some children should be able to:

- apply conditional formatting to the frequency table.

Glossary

- **Animation:** a moving picture.
- **Cell:** a 'box' or container in which you can put a number.
- **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.
- **Formatting:** to change the style of text, e.g. alignment, spacing and bold.
- **Formula:** a maths calculation. In Excel, a formula always begins with an equals (=) sign.
- **Function:** a section of code that makes something happen.
- **Hyperlink:** a link that takes a user to another document or web page, or to another part of a document.
- **Input:** information given to a computer to make something happen, e.g. a mouse click or button press.
- **Motion path:** this allows you to create a 'journey' for an object so it looks like it is moving.
- **Output:** something that a computer produces when given an instruction, e.g. an on-screen image, a sound or vibration.
- **Program:**
  1. a sequence of instructions to perform a task or solve a problem, using a programming language (noun)
  2. to create or change a program (verb).
- **Repetition:** the act of repeating something: following an instruction again (and again).
- **Rule:** an instruction that has to be followed.
- **Script:** a set of commands that are followed by a program.
- **Sequence:** a set of commands that are performed one after another.
- **Simulate:** use a computer to produce a 'model' of something happening, without actually having to do it.
- **Sprite:** an object or character that can be programmed.
- **Trigger:** how something is started.

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

www.risingstars-uk.com

Author: Claire Lotriet

Published 2015, reprinted 2019

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 *Microsoft* PowerPoint® used with permission from *Microsoft*®; page 7: screenshot from *Microsoft* Excel® used with permission from *Microsoft*®.