



Raspberry Pi

Year 2 – Programming quizzes

Unit introduction

This unit initially recaps on learning from the Year 1 ScratchJr unit 'Programming B – Programming animations'. Learners begin to understand that sequences of commands have an outcome, and make predictions based on their learning. They use and modify designs to create their own quiz questions in ScratchJr, and realise these designs in ScratchJr using blocks of code. Finally, learners evaluate their work and make improvements to their programming projects.

Software and Hardware requirements

All the lessons in this unit require access to ScratchJr. ScratchJr is available as an App for tablets and Chromebooks. An unofficial version of ScratchJr is available for computers ([Scratch Jr for Desktop - Open Source Community Port](#)).

If you've adapted this unit to better suit your school, please [share your adapted resources](#) with fellow teachers in the STEM community. Alternatively, if this unit isn't quite right for your school, why not see if an adapted version which better suits has already been shared?

Overview of lessons

Lesson	Brief overview	Learning objectives
ScratchJr recap	During this lesson, learners will recap what they know already about the ScratchJr app. They will begin to identify the start of sequences in real-world scenarios, and learn that sequences need to be started in ScratchJr. Learners will create programs and run them in full-screen mode using the Green flag .	To explain that a sequence of commands has a start <ul style="list-style-type: none">I can identify the start of a sequence

		<ul style="list-style-type: none"> • I can identify that a program needs to be started • I can show how to run my program
Outcomes	During this lesson, learners will discover that a sequence of commands has an 'outcome'. They will predict the outcomes of real-life scenarios and a range of small programs in ScratchJr. Learners will then match programs that produce the same outcome when run, and use a set of blocks to create programs that produce different outcomes when run.	<p>To explain that a sequence of commands has an outcome</p> <ul style="list-style-type: none"> • I can predict the outcome of a sequence of commands • I can match two sequences with the same outcome • I can change the outcome of a sequence of commands
Using a design	During this lesson, learners will be taught how to use the Start on tap and Go to page (Change background) blocks. They will use a predefined design to create an animation based on the seasons. Learners will then be introduced to the task for the next lesson. They will predict what a given algorithm might mean.	<p>To create a program using a given design</p> <ul style="list-style-type: none"> • I can work out the actions of a sprite in an algorithm • I can decide which blocks to use to meet the design • I can build the sequences of blocks I need
Changing a design	During this lesson, learners will look at an existing quiz design and think about how this can be realised within the ScratchJr app. They will choose backgrounds and characters for their own quiz projects. Learners will modify a given design sheet and create their own quiz questions in ScratchJr.	<p>To change a given design</p> <ul style="list-style-type: none"> • I can choose backgrounds for the design • I can choose characters for the design

		<ul style="list-style-type: none"> I can create a program based on the new design
Designing and creating a program	During this lesson, learners will create their own quiz question designs including their own choices of question, artwork, and algorithms. They will increase the number of blocks used within their sequences to create more complex programs.	<p>To create a program using my own design</p> <ul style="list-style-type: none"> I can choose the images for my own design I can create an algorithm I can build sequences of blocks to match my design
Evaluating	During this lesson, learners will compare their projects to their designs. They will think about how they could improve their designs by adding additional features. They will modify their designs and implement the changes on their devices. Learners will find and correct errors in programs (debug) and discuss whether they debugged errors in their own projects.	<p>To decide how my project can be improved</p> <ul style="list-style-type: none"> I can compare my project to my design I can improve my project by adding features I can debug my program

Subject knowledge and CPD opportunities

This unit focuses on developing learners' understanding of computer programming. It highlights that algorithms are a set of clear, precise, and ordered instructions, and that a computer program is the implementation of an algorithm on a digital device. The unit also introduces reading 'code' to predict what a program will do. Learners will engage in aspects of program design, including outlining the project task and creating algorithms.

When programming, there are four levels that can help describe a project, known as Levels of abstraction. Research suggests that this structure can support learners in understanding how to create a program and how it works:

Task – what is needed

Design – what it should do

Code – how it is done

Running the code – what it does

Spending time at the ‘task’ and ‘design’ levels before engaging in code-writing aids learners in assessing the achievability of their programs, and reduces a learner’s cognitive load during programming. Learners will move between the different levels throughout the unit, and this is highlighted within each lesson plan.

Continuing professional development

Enhance your subject knowledge to teach this unit through the following free CPD:

- [Getting started in Year 2 – short course](#)
- [Introduction to primary computing remote or face to face](#)
- [Teaching programming to 5- to 11-year-olds](#)
- [Teaching programming using Scratch and ScratchJr](#)

Teach primary computing certificate

To further enhance your subject knowledge, enrol on the [teach primary computing certificate](#). This will support you to develop your knowledge and skills in primary computing and gain the confidence to teach great lessons, all whilst earning a nationally recognised certificate!

Progression

This unit progresses learners’ knowledge and understanding of instructions in sequences and the use of logical reasoning to predict outcomes.

Common misconceptions

In this unit, pupils learn that different sequences can have the same outcome. Whilst this is an important teaching point, it is also important that pupils still understand that though the final outcome is the same, the order of the sequence can still be important. This is especially important in Scratch Jr where the blocks run one after the other with a break between, and each step of the code can be seen. Alongside showing pupils the sequences on the screen, demonstrate them running in Scratch Jr simultaneously to show that though the outcome is the same, the movement to get there is different.

Curriculum links

Computing

- Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs
- Use technology purposefully to create, organise, store, manipulate and retrieve digital content
- Use logical reasoning to predict the behaviour of simple programs

Maths

Measure

- sequence events in chronological order using language [for example, before and after, next, first, today, yesterday, tomorrow, morning, afternoon and evening]

Geometry - position and direction

- describe position, direction and movement, including whole, half, quarter and three-quarter turns

Assessment

Formative assessment

Assessment opportunities are detailed in each lesson plan. The learning objective and success criteria are introduced in the slide deck at the beginning of each lesson and then reviewed at the end. Learners are invited to assess how well they feel they have met the learning objective using thumbs up, thumbs sideways, or thumbs down.

Summative assessment

Please see the assessment rubric document for this unit. The rubric can be used to assess learning and highlights whether the pupil is approaching (emerging), achieving (expected), or exceeding the expectations in this unit.

Resources are updated regularly — please check that you are using the latest version.

Attribution statement

This resource was created by Raspberry Pi Foundation and updated by STEM Learning for the National Centre for Computing Education.

The contents of this resource are available for use under the [Open Government License](#) (OGL v3) meaning you can copy, adapt, distribute and publish the information. You must acknowledge the source of the Information in your product or application, by attributing Raspberry Pi Foundation and STEM Learning as stated here and are asked to provide a link to the [OGL v3](#).

The original version can be made available on request via info@teachcomputing.org.