

Year 2 – An introduction to quizzes

Unit introduction

This unit initially recaps on learning from the Year 1 Scratch Junior unit ‘Programming B - Introduction to animation’ 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.

There are two Year 2 programming units:

- Programming A – Robot algorithms
- Programming B – An introduction to quizzes

This is unit B which should be delivered before unit A.

Overview of lessons

Lesson	Brief overview	Learning objectives
1. 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 identify that sequences need to be started in ScratchJr. Learners will create programs and run them in full screen using the green flag.	To explain that a sequence of commands has a start • I can identify the start of a sequence • I can identify that a program needs to be started • I can show how to run my program

2. Outcomes	During this lesson learners will be taught 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 then use a set of blocks to create programs that produce different outcomes when run.	To explain that a sequence of commands has an outcome • 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
3. 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 then use a predefined design to create an animation based on the seasons. Learners will then be introduced to next week’s task and predict what a given algorithm might mean.	To create a program using a given design • I can tell 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
4. 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 then choose backgrounds and characters for their own quiz project. Learners will modify a given design sheet and then create their own quiz question in ScratchJr.	To change a given design • I can choose backgrounds for the design • I can choose characters for the design • I can create a program based on the new design

5. Designing and creating a program	During this lesson learners will create their own quiz question designs including their own choice of question, artwork and algorithms. They will increase the number of blocks used within their sequences to create more complex programs.	To create a program using my own design • I can choose the images for my own design • I can create an algorithm • I can build sequences of blocks to match my design
6. Evaluating	During this lesson learners will compare their project to their design. They will then think about how they could improve their design by adding additional features, modify their design and implement this on their devices. Learners will find and correct errors in programs (debug) and discuss whether they debugged errors in their own projects.	To decide how my project can be improved • I can compare my project to my design • I can improve my project by adding features • I can debug

Progression

This unit progresses learners' knowledge and understanding of

Please see the learning graph for this unit for more information about progression.

Curriculum links

[National curriculum links](#)

- 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

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.

Subject knowledge

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.

Enhance your subject knowledge to teach this unit through the following training opportunities:

Online training courses

- [Raspberry Pi Foundation online training courses](#)

Face-to-face courses

- [National Centre for Computing Education face-to-face training courses](#)

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

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](https://www.ncce.io/ogl).