



Raspberry Pi

## Year 3 – Programming A - Sequencing sounds

### Unit introduction

This unit explores the concept of sequencing in programming through Scratch. It begins with an introduction to the programming environment, which will be new to most learners. They will be introduced to a selection of motion, sound, and event blocks which they will use to create their own programs, featuring sequences. The final project is to make a representation of a piano. The unit is paced to focus on all aspects of sequences, and make sure that knowledge is built in a structured manner. Learners also apply stages of program design through this unit.

There are two Year 3 programming units:

- Programming A - Sequencing sounds
- Programming B - Events and actions in programs

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

### Overview of lessons

Lesson	Brief overview	Learning objectives
1. Introduction to Scratch	This lesson introduces learners to a new programming environment: Scratch. Learners will begin by comparing Scratch to other programming environments they may have experienced, before familiarising themselves with the basic layout of the screen.	To explore a new programming environment <ul style="list-style-type: none"> <li>• I can identify the objects in a Scratch project (sprites, backdrops)</li> <li>• I can explain that objects in Scratch have attributes (linked to)</li> </ul>

		<ul style="list-style-type: none"> <li>I can recognise that commands in Scratch are represented as blocks</li> </ul>
2. Programming sprites	In this lesson, learners will create movement for more than one sprite. In doing this, they will design and implement their code, and then will create code to replicate a given outcome. Finally, they will experiment with new motion blocks.	<p>To identify that commands have an outcome</p> <ul style="list-style-type: none"> <li>I can identify that each sprite is controlled by the commands I choose</li> <li>I can choose a word which describes an on-screen action for my plan</li> <li>I can create a program following a design</li> </ul>
3. Sequences	In this lesson, learners will be introduced to the concept of sequences by joining blocks of code together. They will also learn how event blocks can be used to start a project in a variety of different ways. In doing this, they will apply principles of design to plan and create a project.	<p>To explain that a program has a start</p> <ul style="list-style-type: none"> <li>I can start a program in different ways</li> <li>I can create a sequence of connected commands</li> <li>I can explain that the objects in my project will respond exactly to the code</li> </ul>
4. Ordering commands	This lesson explores sequences, and how they are implemented in a simple program. Learners have the opportunity to experiment with sequences where order is and is not important. They will create their own sequences from given designs.	<p>To recognise that a sequence of commands can have an order</p> <ul style="list-style-type: none"> <li>I can explain what a sequence is</li> <li>I can combine sound commands</li> <li>I can order notes into a sequence</li> </ul>
5. Looking good	This lesson develops learners' understanding of sequences by giving them the opportunity to combine motion and sounds in one sequence. They will also learn how to use costumes to change the appearance of a sprite, and backdrops to	<p>To change the appearance of my project</p> <ul style="list-style-type: none"> <li>I can build a sequence of commands</li> </ul>

	change the appearance of the stage. They will apply the skills in Activity 1 and 2 to design and create their own project, including sequences, sprites with costumes, and multiple backdrops.	<ul style="list-style-type: none"> <li>• I can decide the actions for each sprite in a program</li> <li>• I can make design choices for my artwork</li> </ul>
6. Making an instrument	In this lesson, learners will create a musical instrument in Scratch. They will apply the concept of design to help develop programs and use programming blocks — which they have been introduced to throughout the unit. They will learn that code can be copied from one sprite to another, and that projects should be tested to see if they perform as expected.	<p>To create a project from a task description</p> <ul style="list-style-type: none"> <li>• I can identify and name the objects I will need for a project</li> <li>• I can relate a task description to a design</li> <li>• I can implement my algorithm as code</li> </ul>

## Progression

This unit assumes that learners will have some prior experience of programming; the KS1 NCCE units cover floor robots and ScratchJr. However, experience of other languages or environments may also be useful.

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

## Curriculum links

### [National curriculum links](#)

- Design, write, and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- 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, and to detect and correct errors in algorithms and programs

- Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

## Assessment

### Formative assessment

Assessment opportunities are detailed in each lesson plan. The learning objectives and success criteria are introduced in the slide decks 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 student's work from lesson 6.

We recommend the use of teacher accounts in Scratch to help with assessment throughout this unit. For guidance on setting up teacher accounts, please visit [the Scratch website](https://scratch.mit.edu/educators/faq). (<https://scratch.mit.edu/educators/faq>)

## Subject knowledge

This unit focuses on developing learners' understanding of sequences in a new programming language. It highlights that the order of sequences is important. This unit also develops learners' understanding of design in programming, using the approach outlined below.

When programming, there are four levels which 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 can aid learners in assessing the 'do-ability' of their programs. It also 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 free CPD:**

- [Getting started in Year 3 – short course](#)
- [Introduction to primary computing remote or face to face](#)
- [Teaching programming to 5- to 11-year olds](#)
- [Teaching programming using Scratch and Scratch Jr](#)

To further enhance your subject knowledge, enrol on the [primary 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!

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 [nccpe.io/ogl](https://www.nccpe.io/ogl).