

Aim

This document provides guidance on the four Starter Lessons which provide an introduction and foundation to the Learn to Code course for students and teachers.

The Starter Lessons guide you through the main elements of the Learn to Code course.

We advise that they are followed in order:

- 1 Introduction to SAM Blockly: the online block-based coding software
- 2 Introduction to SAM Blocks: connecting hardware blocks to SAM Blockly
- 3 Introduction to micro:bit
- 4 What is Computational Thinking?

Lesson Format

Each Starter Lesson consists of Lesson Slides and a Mission Journal for students. They are designed to be completed in a 30 minute session and structured into four sections, including offline ('unplugged') and online.

The structure consists of:

Learn: Key information

Do: Let's Build! > Challenge > Chili Challenges

Reflect: Consolidation, extend and evidence of learning in the Mission Journal



Overview

Starter Lesson 1 Introduction to SAM Blockly

Hardware blocks none

Software blocks

- 'program start' block
- 'print (" ")' block

Overview This lesson will introduce our character Sam and the missions she is sent on which form the link between lessons. Students will be introduced to Cyberspace where Sam finds herself. Students will also be introduced to the software platform SAM Blockly, and code a program to learn about the different blocks within the platform.

Objectives Students will:

- **become familiar** with the character Sam.
- **understand** how Sam found herself in Cyberspace.
- **understand** where general blocks are within SAM Blockly.
- **create** a code that will output a question in the Console to learn more about fellow classmates.



Overview

Starter Lesson 2

Introduction to SAM Blocks

Hardware blocks

Light Sensor (If hardware block unavailable, a virtual Light Sensor may be used)

Software blocks

- 'when Light Sensor value changes'
- 'get Light Sensor value'
- 'wait for [] seconds" block
- 'print (" ")' block

Overview

This lesson will introduce pairing the SAM hardware blocks within SAM Blockly — the platform that allows block-based programs to be built. Students will learn how to create a program in SAM Blockly and extend understanding to programming the output to show the change in light detected by the Light Sensor. Students will extend their learning through differentiated Chili Challenges.

Objectives

Students will:

- **become familiar** with 'Blockly' coding in SAM Blockly
- **understand** how the blocks are used on the workspace.
- **understand** the purpose of different areas of the workspace.
- **create** a simple program to output the changing Light Sensor values on the Console.
- **develop** the program by utilizing the 'wait' block to slow down the output.

Overview

Starter Lesson 3 Introduction to micro:bit

Hardware blocks Micro:bit (If physical micro:bit unavailable, a virtual micro:bit may be used)

Software blocks

- 'when micro:bit [A] is [pressed]' block
- 'on micro:bit display' block
- 'on micro:bit display ("word")' block

Overview This lesson will introduce the BBC micro:bit. Students will connect the micro:bit to SAM Blockly, learning how to use the programming blocks linked to the micro:bit.. Students will be introduced to the key computing terms '**events**', '**actions**' and '**values**.' Students will code a system that displays their names scrolling across the micro:bit. Differentiated Chili Challenges extend students' learning further.

Objectives Students will:

- **become familiar** with the micro:bit
- **understand** how to connect the micro:bit to SAM Blockly
- **understand** that programming blocks become available when a device is connected to SAM Blockly
- **create** a simple program to output an image on the micro:bit
- **develop** the program by utilizing drop-down settings within the blocks.
- **develop** the program by changing the output to scroll the students' names

Overview

Starter Lesson 4 Introduction to Computational Thinking

Hardware blocks none

Software blocks none

Overview This lesson will introduce the four main components of computational thinking: decomposition, pattern recognition, abstraction, and algorithmic thinking. Students will be guided through the definitions and examples of each stage. Additionally, students will be tasked with using all four components through the design brief of creating a new SAM hardware block.

Objectives Students will:

- **describe** all four stages of computational thinking.
- **identify** examples of computational thinking in their world.
- **apply** computational thinking to new problems and challenges.