

Barefoot - wildlife animations

**Lessons:** 2

**Programming languages:** MakeCode

**Target age:** 7-11 yrs

**Subjects & topics:**

* Programming: Sequence, Iteration
* Computer systems: Input/output

# Unit of work summary

These lessons are aimed at students aged 7-9. Students familiarise themselves with the BBC micro:bit. They also create icons and animations by programming the micro:bit's LED display and using the forever block and count-controlled loops. You will ideally use physical micro:bits for these lessons, although you can also use the simulator.

## Overall key learning

* I can explain what the micro:bit's components do
* I can sequence code within the micro:bit MakeCode editor
* I can design and code images using the micro:bit's LEDs
* I can use repetition within the micro:bit MakeCode editor
* I can design and code an animation using the micro:bit’s LEDs
* I can evaluate my work and suggest improvements

## Additional skills

Creative thinking, evaluation

## Lesson 1: Barefoot - wildlife animations, lesson 1

In this lesson, pupils familiarise themselves with the micro:bit and identify various components on the device. Pupils examine how to transfer programs to the device and undertake a related unplugged activity. Pupils create their own programs using the predefined images, then go on to create their own images by programming individual LEDs.

**Key learning:**

* I can explain what the micro:bit's components do
* I can sequence code within the micro:bit MakeCode editor
* I can design and code images using the micro:bit’s LEDs

## Lesson 2: Barefoot - wildlife animations, lesson 2

In this lesson, pupils produce further programs with the micro:bit. Pupils produce an animation away from the device by creating a thaumatrope (an animation based on quickly moving between two images), which they share with the class. They then write a program containing their animation and a forever loop using MakeCode, before transferring it to the micro:bit. Pupils go on to use count controlled loops by producing a second animation, which they include within a program.

**Key learning:**

* I can use repetition within the micro:bit MakeCode editor
* I can design and code an animation using the micro:bit’s LEDs
* I can evaluate my work and suggest improvements

# Curriculum links

These lessons are mapped to the following learning objectives and standards:

## England National Curriculum

#### KS2 computing curriculum

Curriculum aims:

* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems

Students should be taught to:

* 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
* 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

[Read the full KS2 computing curriculum.](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf)

#### Year 3 science curriculum

Students should be taught to:

* Plants - identify and describe the functions of different parts of flowering plants: roots, stem/trunk, leaves and flowers.
* Animals - identify that humans and some other animals have skeletons and muscles for support, protection and movement.

[Read the full KS2 science curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/425618/PRIMARY_national_curriculum_-_Science.pdf)

## Scotland Curriculum for Excellence

#### Technologies

* I understand the instructions of a visual programming language and can predict the outcome of a program written using the language. (TCH 1-14a)
* I understand how computers process information. (TCH 1-14b)
* I can demonstrate a range of basic problem solving skills by building simple programs to carry out a given task, using an appropriate language. (TCH 1-15a)
* I can explore and experiment with digital technologies and can use what I learn to support and enhance my learning in different contexts. (TCH 1-01a)

[Read the full Curriculum for Excellence: technologies](https://education.gov.scot/Documents/Technologies-es-os.pdf).

## Northern Ireland Curriculum - Primary

#### Using ICT - Computational Thinking and Coding – Level 3

* Pupils should create a coding project such as an animation or make a pattern or shape.

#### Programmable devices (micro:bit)

* look at and talk about examples of coding projects, including the use of movement, speed, direction and loops;
* talk about how these projects are composed of different components and break the larger task into smaller manageable tasks (decomposition);
* plan what they want to happen in their own coding project, work out what different parts of the program must do and use logical reasoning to predict the commands that are required;
* use a range of commands, including movement with direction and speed, and light or sound commands;
* make code more efficient, for example by using loop control commands;
* test the program, checking that the commands are logical and debugging any errors, at regular intervals.

Pupils should:

* save using filenames into a given folder or location the teacher specifies;
* share their work with the teacher or others and discuss how they can improve it and
* make any agreed changes and save again.

[Read CCEA's Using ICT desirable features](https://ccea.org.uk/learning-resources/using-ict-desirable-features)

#### The Arts – Art and Design – KS1

Pupils should be enabled to:

* look at and talk about resource material to stimulate their own ideas (plants & nature)
* experiment with a range of media, materials, tools and processes such as drawing and three-dimensional construction.

[Read the CCEA Art and Design curriculum (page 72)](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/The%20Northern%20Ireland%20Curriculum%20-%20Primary.pdf)

## Curriculum for Wales

#### Science and technology

Progression step 2 - computation is the foundation for our digital world:

* I can create simple algorithms and am beginning to explain errors
* I can follow algorithms to determine their purpose and predict outcomes

Progression step 3 - computation is the foundation for our digital world:

* I can use conditional statements to add control and decision-making to algorithms
* I can identify repeating patterns and use loops to make my algorithms more concise
* I can explain and debug algorithms

Progression step 2 - design thinking and engineering offer technical and creative ways to meet society's needs and wants:

* I can produce designs to communicate my ideas in response to particular contexts
* I can explore how different component parts work together.

Progression step 3 - design thinking and engineering offer technical and creative ways to meet society's needs and wants:

* I can draw inspiration to design from historical, cultural and other sources.
* I can use design thinking to test and refine my design decisions without fear of failure.

[Read the full science and technology curriculum](https://hwb.gov.wales/curriculum-for-wales/science-and-technology/descriptions-of-learning/)

#### Digital competency framework

Progression step 2 - data and computational thinking - problem-solving and modelling:

* I can break down a problem to predict its outcome.
* I can detect and correct mistakes which cause instructions (a solution) to fail (debug)
* I can change instructions to achieve a different outcome.
* I can identify repetitions or loops in a sequence.

Progression step 3 - data and computational thinking - problem-solving and modelling:

* I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values and formulae.
* I can understand the importance of the order of statements within algorithms.

Progression step 2 - producing - evaluating and improving digital content:

* I can give an opinion about my own work and suggest improvements based on the success criteria

Progression step 3 - producing - evaluating and improving digital content

* I can explain reasons for the layout and content of my own work and the work of others
* I can ensure my output is appropriate for specific purposes

[Read the digital competence framework](https://hwb.gov.wales/curriculum-for-wales/cross-curricular-skills-frameworks/digital-competence-framework)

## USA CSTA Standards

#### Grades 3-5

* 1B-CS-01 - Describe how internal and external parts of computing devices function to form a system.
* 1B-CS-02 - Model how computer hardware and software work together as a system to accomplish tasks
* 1B-CS-03 - Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.
* 1B-AP-12 - Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
* 1B-AP-17 - Describe choices made during program development using code comments, presentations, and demonstrations.

[Read the CSTA Standards in full.](https://csteachers.org/k12standards/ )

This content is published under an [Open Government](https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/) licence.