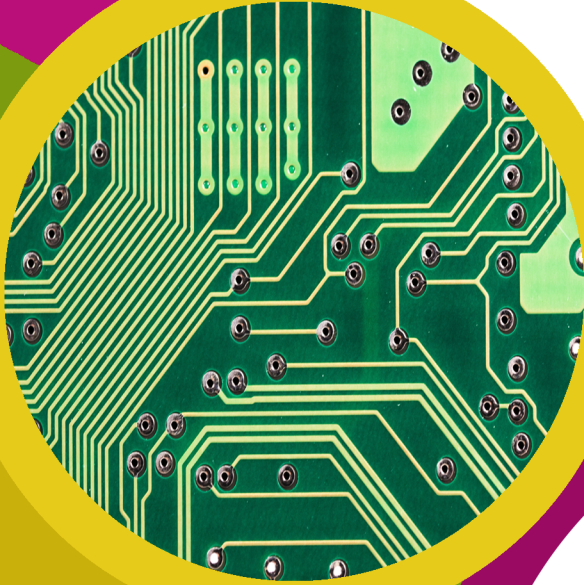
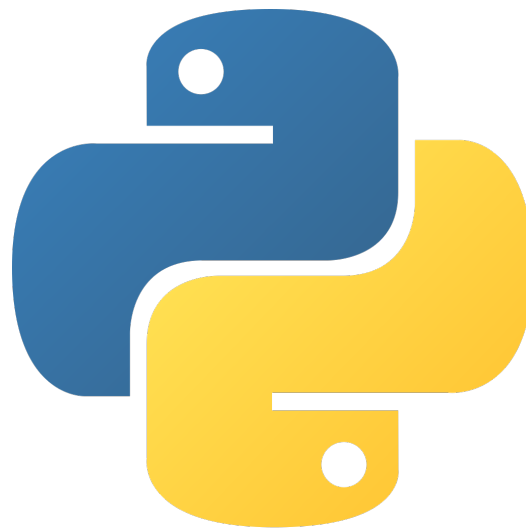


technocamps

Micro:bit Helping Animals Teacher Guidance



01010100
01100101011
0001101101000
01101110011011
1011000110110
000101101101
0111000001
110011



Prifysgol
Metropolitan
Caerdydd

it.wales



University of
South Wales
Prifysgol
De Cymru

Links to Science and Technology AoLE

Computation:

(PS2) I can create simple algorithms and am beginning to explain errors.

(PS3) I can use conditional statements to add control and decision-making to algorithms.

(PS3) I can use sensors and actuators in systems that gather and process data about the systems' environment.

The World Around Us:

(PS2) I can recognise that what I do, and the things I use, can have an impact on my environment and on living things.

(PS2) I can use my knowledge and understanding to predict effects as part of my scientific exploration.

Links to Other AoLEs

Humanities:

(PS2) I can describe how people and the natural world may impact each other.

(PS2) I can describe how and where some places and environments are similar, and others are different.

Mathematics and Numeracy:

(PS2) I can collect and organise data to ask and answer questions in relevant situations.

The Four Purposes and Cross-Curricular Skills

This resource provides **Critical Thinking and Problem-Solving** opportunities. Students are required to follow and design an algorithm using block-based programming. They are able to analyse errors in the code, identify solutions, and deduce the next steps in the code.

Learners will also use **Creativity and Innovation**. They are encouraged to discuss and implement strategies to improve their program. Learners are also taught about problems facing animals and the environment and are encouraged to think of possible solutions to different scenarios.

The **Data and Computational Thinking** section of the **DCF** applies to this resource. Students will learn to code using loops, events, and conditionals to create an algorithm that takes input from the environment and displays an appropriate output. The **Numeracy** section of the **LNF** also applies to this resource as students will learn to collect data from their environment and base decisions on this data.

Why Is Learning This Important?

This resource provides learners with the opportunity to create simple algorithms with a demonstrable application, using a block-based programming language. It introduces concepts such as conditionals, loops, and event-based programming which are critical to most common programming languages. The resource also explores interactions between humans and animals, the threats facing certain species and how we can work towards solutions to these problems using programming. This can be expanded to introduce students to text-based programming such as Python.

Suggested Approaches Key

In this suggested approach we use the following colours to differentiate the types of activities:

- **Yellow - Explain.** Teachers should explain the slide/example to the class.
- **Green - Discuss.** Teachers should start an open discussion with the class to get them to feedback some answers/ideas.
- **Purple - Activity.** Students are expected to complete an activity whether it be in their workbooks or on the computer, followed by a discussion of their solutions.
- **Green - Introduction/Conclusion.** The introduction/conclusion is also colour coded green. Teachers should hand out materials in the introduction and conclude the session and collect materials at the end.

Introduction

Begin with introductions, and a brief explanation of the Technocamps programme, before handing out any resources required by learners and any additional aids for learners with additional learning needs.

Explain: Topics Covered Today

We will be learning about human-animal interactions and how we can use micro:bits and programming to create solutions to animal problems.

Discuss: Micro:bit

Provide each student with a micro:bit and ask them what they know about it. Have they used them before? Can they tell you what some of the components are (e.g. buttons, LED lights, USB connector)?

Explain: What is micro:bit?

The micro:bit is a very small computer that is used to teach how hardware and software work together.

It has several components: 25 Led lights that can be used to display images, sensors that can detect light/temperature/movement, buttons, and radio and bluetooth antenna.

We can program the micro:bit to take input, display output, process information, communicate with other micro:bits and many more things.

Explain: What is programming?

Programming is telling a computer what to do using a set of ordered instructions. The set of ordered instructions is called an **algorithm**. The language used to tell the computer what to do is called a **programming language**.

Introduce the students to the MakeCode editor and explain how to connect their micro:bit devices.

Explain: Downloading Programs

Each time you make changes to your program you need to click on the download button before the micro:bit can run the program.

Activity: Animal Icons

Program the micro:bit to display an image of an animal. Drag the **show icon** block into the **forever** loop and choose the icon from the dropdown menu.

Explain: LED lights

The **show icon** blocks allow us to display default images on the micro:bit. Using the **show leds** block, we can customise the image that is displayed.

Activity: Custom Icons

Ask the students to create their own animal icons using the **show leds** block. Clicking on each individual square turns that light on or off. You can show them the examples on the slide.

Explain: Loops

Loops can be used to repeat commands in a program without typing out each action every time. They can be repeated forever, for a certain number of times, or for a given condition.

In MakeCode, these are found in the **Loops** section.

For a loop to have a purpose, an action command needs to be placed inside it.

All MakeCode programs start with a default **forever** loop. This loop will run a set of commands until the micro:bit is unplugged or reset. You can only have one **forever** loop in a micro:bit code.

Activity: Animation

Ask the students to create a program that displays two alternating animal images. They can do this in the default **forever** loop.

Suggest that they add a **pause** command between the two icons.

Discuss: Forever vs. On Start

Explain the difference between the **on start** block and the **forever** loop. Ask the students what they think would happen if they put the code from the animation activity in the **on start** block instead of the **forever** block.

Answer: The code would run once and then stop without repeating.

Activity: Custom Animation

Ask the students to create their own animation using the **forever** block and the **show leds** command. They can use the animal icons they made earlier to try and get them to move. The slide example is a fish. Remind them to click download when they are done.

Explain: Micro:bit senses

Each micro:bit has several sensors in it. They can detect light, sound, movement, and temperature. We can use this input data to tell the micro:bit to carry out different actions under different conditions.

For example, if it gets too dark we can tell the micro:bit light to turn on.

To do this, we use conditionals.

Explain: Logic and If-statements

Conditions help us perform different actions based on different conditions. These conditions should always result in Yes/No or True/False.

For example:

If my homework is done, **then** I can go outside to play.

If I have eaten my dinner, **then** I can have dessert.

Ask the students for more examples.

In MakeCode, the if-statements are found in the **Logic** section.

Explain: Shake Sensor

The micro:bit can sense several different types of movement such as shaking the device, tilting it to the right or left, or even dropping the device into “free fall”.

Activity: Shake Sensor

Ask the students to program a shake sensor. You can do this with them, step-by-step using the slides.

1. Drag the **if true then** block from the **Logic** section in MakeCode.
2. Drag the **is shake gesture** command from the **Input** section into the **if** statement. Ask the students what they think the next step will be.
3. Add the **show leds** command you saw earlier into the first indent of the **if** statement. The students can choose any pattern they want to create.
4. Download and run the code. Ask the students if they can see a problem with this program? (The lights do not turn off when they stop shaking the micro:bit)
5. Add a **clear screen** command after the **else** statement. Try running the code again - it should work as expected.

Discuss: Sea Turtles

Ask the students what they know about sea turtles. Where do they live? What do they eat? Are they endangered? Do they know what endangered means?

Explain: Sea Turtles

Sea turtles are marine (salt water) reptiles that live in oceans all around the world.

There are seven species.

They can eat seaweed, seagrass, jellyfish, fish, etc.

Young turtles spend their time in the open ocean but when they get older, they move closer to the shores. Adult female will return to land to lay their eggs on beaches.

Discuss: Sea Turtle Problems

Many sea turtle species are in trouble and their populations are declining. Ask the students why they think this may be? How do you think humans have affected sea turtle populations?

- Habitat loss - Tourism and urbanisation making beaches uninhabitable and destruction of coral reefs.
- Fishing - Either fishing for turtles, or accidentally catching turtles when fishing for other animals.
- Plastics in the ocean
- Climate change - changing ocean temperature, and changing the male:female ratio in offspring (more females born when temperature is warmer).

Explain: Sea Turtles Hatchlings

When sea turtle eggs hatch on the beach, the young turtles make their way to the sea in search of food and shelter.

They use the moonlight and stars reflecting off the water to guide them to the ocean. Bright lights used by humans, such as lamp posts and torches, can confuse the turtles causing them to get lost. Ask the students if they can think of any solutions to this?

Instead of strong lights we can use turtle-safe lights to guide humans around turtle nesting beaches. We can program the micro:bit to display a red low wattage light that can be placed along the ground. They are bright enough to guide humans without confusing the turtles.

Activity: Turtle-Safe Lights

Ask the students to create a program that gets the micro:bit to light up when the environment gets dark. This will be a similar algorithm to the one used to make a shake sensor. The students will just need to change the input requirement.

Step-by-step instructions are provided on the slides. As you go through the slides, ask the students if they can work out what the next step in the code will be.

1. Start by creating a turtle icon using the **show leds** command.
2. Drag and drop the **if true then** block from the **Logic** section into the **forever** loop.
3. Next we need to program the micro:bit to react to the light level. Add the **0 < 0** command to the **if** statement. Add the **light level** command and change the number to 50.
4. Add the **show leds** and **clear screen** commands.
5. Try running the code - can the students get their micro:bits to light up? If not, the light intensity number (50) may need to be changed.

Discuss: Bycatch

Ask the students if they know what this word means.

Big fishing boats send out huge nets that can catch thousands of fish. Sometimes the wrong species get caught in the nets - this is called bycatch. Animals such as dolphins, birds, and turtles can get accidentally caught in fishing nets.

Ask the students if they can think of any ways of preventing this.

Some answers:

- Using specially designed nets that can allow larger animals to escape.
- Alerting other animals to the presence of nets so they can avoid them - using light or sound.

Discuss: Fishing Net Lights

We want to program a micro:bit to light up when underwater so that fishermen can see into the nets to make sure there is not bycatch. The lights can also deter some species (birds and turtles).

Ask the students which sensors on the micro:bit we can use to determine whether the device is underwater.

Light - the deeper the nets, the less light they will receive.

Movement - the waves and currents can shake the nets. Fish and bycatch entering the net can also trigger a movement sensor.

Temperature - the temperature will be different in and out of water.

Activity: Fishing Net Lights

Ask the students to program the micro:bit to light up when it senses a decrease in light intensity and an increase in movement. This will be a combination of the shake sensor and light sensor they made earlier.

They will need to use an **and** command found in the **Logic** section and add it to the **if statement**. Add the **light level < 50** and **is shake gesture** commands to the **if** block.

The completed program is shown on the slides. See if the students can get the micro:bit to light up.

Explain: Fishing Net Sounds

The lights on the fishing nets can deter some bycatch such as turtles and birds. Other animals such as dolphins and porpoises can be deterred using sound alerts in the water.

We can add a sound output to the micro:bit program that makes a noise when the device lights up.

Sound output can be found in the **Music** section.

Extension Activity: Fishing Net Sounds

Ask the students to add a sound output to their code. See if they can do it before revealing the completed code.

Drag the **play tone Middle C for 1 beat** into the code, after the **show leds** command.

Discuss: Land Animals

Ask the students what they know about land animals.

What different types are there?

How many species do you think there are?

Where can you find the most animals?

Explain: Land Animals

There is a huge variety of animals - mammals, birds, insects, reptiles, amphibians, etc. Which of these is the most common (insects)?

There are an estimated 6.5 million species of land animals, of which only 25,000 can be found in the UK (only 0.4%).

Rainforests have the greatest species diversity but animals can be found in almost every environment you can think of (deserts, polar regions, etc.)

Discuss: Animal Problems

Ask the students if they can think of any problems that land animals face that are caused by humans.

- Climate change
- Habitat loss
- Poaching
- Plastics and other garbage
- Pollution

Can the students think of any more? Which of these apply to animals in the UK?

Explain: Local Wildlife

Even though the UK only has a small fraction of the world's animals, they still face some serious threats - climate change, pollution, pesticides, and plastics all affect UK animals.

We don't always know which animals are at threat and need help. Therefore, it is very important to keep track of populations. Can the students think of a way to do this? Can we use the micro:bit for this?

Explain: Species Counter

A species counter is simply a device you can use to count how many individuals of a certain species you can see. This will give us an estimate of the number of animals in the area. If we do this over a longer period, we can see how the populations change over time and which animals are declining. This tells us which species of animals need our help most to fight the threats of humans.

We can program our micro:bits to act as a counter, and then you can use this to count species in your local area.

Activity: Species Counter

Instruct the students on how to make a micro:bit species counter. Ask the students for ideas at each step.

1. Create a variable for the animal we want to investigate. This is found in the **Variables** section. Give the variable the name of your animal.
2. Drag the **set ... to 0** command into the **on start** block. How can we get the micro:bit to display this number?
3. Add the **show number** block from the **Basic** section. Drag and drop the animal variable into the **show number** block. What happens when we run the code now?
4. Next we need to program the micro:bit so that you can change the number when you see your animal. For this, we need events...

Explain: Events

Each micro:bit has two buttons: A on the left and B on the right.

These buttons allow us to choose which action to take without reprogramming the micro:bit each time.

For example, we can increase the number on the species counter by one every time we press button A

These commands are found in the **input** section.

Activity: Species Counter

Continue instructing the students on how to make a micro:bit species counter. Ask the students for ideas at each step.

5. Drag a **on button A pressed** block from the **Input** section to your code. What do you think we can add to this block to make our species counter (hint - see what other commands are in the **Variables** section).
6. Add the **change ... by 1** command and another **show number** command.
7. Run the code and see what happens.

Differentiating for Learners

- Extension tasks have been provided to challenge students to improve their code using different output signals. They can add around output as well as the light.
- Some learner may need more guidance in assembling the code than others. Since this workshop is aimed at ages 9-11, there might be several learners with little coding experience so will require step-by-step instructions in assembling the blocks. Other learners can use their previous coding experience to write the algorithm without any code provision.
- The micro:bit website and editor allow for navigation using accessibility features such as a screenreader, or speech recognition software.

Where To Go Next

- When testing different input sensors on the micro:bit, learners can be given time to explore the different possibilities to test the possibilities and limitations of the micro:bits.
- Micro:bit also has a user-friendly Python editor. This workshop can be adapted to introduce learners to a text-based programming language.
- There are several other activities that can be conducted with micro:bit to explore interactions between humans and animals. For instance using the radio feature to track animals.



technocamps



@Technocamps



Find us on
Facebook