Activity 3: Fixing the TARDIS

Teacher note – Curriculum links:

- Understanding the need for care and precision to avoid errors and being able to find and correct errors (bugs) in algorithms and programs

In our **Doctor Who and the micro:bit Live Lesson**, we'll be asking you to help us fix the TARDIS by putting your debugging skills to the test.

During the Live Lesson, we'll be introducing you to three different types of errors – **syntax errors**, **runtime errors** and **logic errors**.

Syntax errors: Syntax errors are mistakes in the way that the code is written. Common syntax errors include spelling mistakes, incorrect use of punctuation and the use of capital letters.

Runtime errors: Runtime errors are errors that will cause the program to crash even if there appears to be nothing wrong with the code itself. This is often when the computer is asked to perform an action that cannot produce a result. For example, you could have written the instructions in the wrong order, or asked a computer to divide a number by 0, which gives you an infinitely large result. In most cases, the computer will tell you that a runtime error has occurred.

Logic errors: Logic errors are those where the program works but produces different results from what you expected.

To help us fix the TARDIS, try to identify three errors in the block of MicroPython code below. There is one of each type of error.

What the code is supposed to do:

- When you start up the micro:bit, the LEDs display an image of a happy face
- When you press button A, the micro:bit's LEDs flashes an image of a heart four times (holding it for one second each time), before showing the message 'CONNECTION ACTIVE'

Simply use a pen or highlighter to underline or highlight the errors you find, and try and see if you classify them into syntax, runtime and logic errors.

```
from microbit import *
TARDIS_signal_Strength = 10
index = 0
while True:
    if button_a.is_pressed():
        TARDIS_signal_Strength = TARDIS_signal_Strength / 0
    if TARDIS_signal_Strength > 99:

        while index < 4:
            display.show(Image.HEART)
            sleep(1000)
            display.clear()
            sleep(1000
            index = index + 1
        display.scroll('CONNECTION ACTIVE')
    else:
        display.show(Image.SAD)
```

Correctly identified the errors? Send in your answers and you could see them featured on our Live Lesson on 28th March 2017.

Simply ask your teachers to email **live.lessons@bbc.co.uk** or use the hashtag **#bbclivelessons**.