# Lesson plan - “Slider bar graph”

## Using the MonkMakes Slider for micro:bit component

## Introduction

The particular focus of the lesson is for loops using range (). The students get the opportunity to learn about the for loops using range () in created contexts and then see it in action in the context of the slider interacting with the micro:bit. The program that they engage with draws a simple bar graph which lights up each row of LEDs depending on how far to the right the slider is positioned.

## Learning objectives

* To successfully set up a micro:bit/slider bar graph circuit and flash code to the micro:bit
* Apply knowledge of a for loop
* To be able to read and interpret a program containing user defined functions
* To be able to read and interpret a program containing a bar graph function
* To be able to read and interpret a program containing an if statement
* To be able to successfully modify a program containing multiple programming constructs

## Keywords

while, for, range, variable, function, in-built, user-defined, subroutine, looper

## Preparation

**Subject knowledge:**

This lesson is suitable for a class who already have a basic knowledge of programming using python. The coding for the micro:bit is written in a reduced version\* of python called micropython but for practical purposes the syntax will appear the same to the user. In addition the students should have already used the micro:bit, practising simple set up and smaller micro:bit specific programs so that they are used to building and flashing programs. Initially, building the circuit for the “slider bar graph” is a question of copying a diagram so a practical demonstration of handling the components gently and clipping with alligator clips would suffice.

**Pedagogical approach:**

The lesson is planned using the PRIMM pedagogy which stands for:

P - Predict

R - Run

I - Investigate

M - Modify

M - Make

For this reason it is important that the class have quick and easy access to the coding of the program which can be accessed via: https://tinyurl.com/ybc8crx8

The very first step is for the students to view the code and attempt to predict what the code does. We use a tool called a **code conversation** which provides the teacher with a conversation style **talkthrough** of the micropython code for you to support your students’ developing knowledge.

**Practical set-up and development environment:**

We recommend either using the on-line Python editor (https://python.microbit.org) or Mu (https://codewith.mu). Or you could manage your coding in micro:bit’s own environment <https://classroom.microbit.org/> which enables you to push code to the whole class and manage their activities.

**Resources that you need:**

* Presentation
* Starter worksheet questions
* Starter worksheet answers
* Code conversation full solution
* Code conversation template (level 1 - basic)
* Code conversation template (level 2 - intermediate)
* Code conversation template (level 3 - advanced)
* Plenary worksheet
* Plenary worksheet answers
* Hardware Per pair - 1 micro:bit, 1 USB/micro USB cable, 3 alligator clip leads, 1 slider
* Access to appropriate development environment (the on-line Python editor, Mu or micro:bit classroom - Mu is needed for the last lesson in the scheme)

## How the students’ progress is assessed

**Slider bar graph**

The presentation itself begins with a slide stating the difference between explicit and implicit type conversion followed by a worksheet checking if the students are already aware of the different data types in python. It is interspersed with pair or whole class discussion opportunities. As this is predominantly a practical lesson observing the students’ success in flashing the program, compiling the circuit and making small successful modifications to the program will also enable you to assess their progress. The learners also complete a worksheet at the end of the lesson which demonstrates their understanding of key principles conveyed during the lesson.

## Plan (with approximate timings)

|  |  |
| --- | --- |
| Starter activity 5 mins | **Range () revision**  The slide that greets the students as they arrive at the lesson has a simple range task on. The same question appears on a worksheet for them to complete individually. The answers are shared with the class and they are encouraged to share their explanation of what is happening “inside” the for x in range () loop. This process of not only being able to answer questions about the right code or output for a loop but to be able to explain the process further consolidates their learning. |
| Focus tasks 10 mins | **Range revision and range format**  In this section the students are given some explicit teaching about the use of range. Students often get confused about the “looper” so this is explained as is the syntax and use of negative numbers in range. The students are challenged to see if they can explain when a for loop should be used instead of a while loop. |
| PRIMM 8 mins | In order to put the lesson and the slider bar graph lesson in context the students need an introduction to the slider itself.  **Predict: students view code and attempt overview code conversation**  The students are shown the code and are asked to predict what the whole program does i.e. an overview in a couple of sentences.  For example: The program responds to the position of the slider and displays it as a bar graph on the micro:bit. The further to the right the slider is positioned the bigger the bar graph. This is contained in the bargraph function that expects a number between 0 and 5 as its parameter and then displays 0 to 5 lines of LEDs. |
| PRIMM 8 mins | **Run: students compile the circuit, flash and run the code**  Share the slide with instructions on the board to enable the students to build the circuit, flash and run the code. Ask them if the circuit behaves as they would have expected from the **predict** phase of the lesson. Encourage the students to move the slider up and down and see how the micro:bit LEDs respond. |
| PRIMM 13 mins | **Investigate: teacher and students use correct terminology to identify the syntax and features of the program**  Using the A4 worksheet, the learners attempt to complete the code conversation matching the correct explanation to each line of code.  There are three levels of the code conversation:  Level 1 - basic (some less challenging code lines to complete), Level 2 - intermediate (more challenging code lines to complete), Level 3 (all lines of code to complete)  Using the slides talk through the different steps in the program. |
| PRIMM 5 mins | **Modify: students modify the code to create a new program**  Students are invited to modify the program by first declaring what their new program should do, then by making the modification, saving the new program and testing it by flashing it to the micro:bit.  Suggestions for achievable alterations:-  Reverse the bar graph so that when the slider is at position 0 the whole microbit lights up and sliding it to the right makes less rows light up.  Changing the brightness any of the LEDs  Find out what happens if you don’t use display.clear()  Change the name of any of the variables or parameters |
| **Plenary**  6 mins | **Slider number plenary worksheet**  Students complete the plenary worksheet testing knowledge of bar graphs and for x in range () loop and function. |
| **Homework** | **Consider what other modifications you could make to the program, still using the slider component** |

## The Author

This lesson plan and all its parts were created by Dr. Paula Beer of Beer Academic Consultancy in collaboration with Monk Makes Ltd.



Dr Paula Beer has taught Computer Science and IT education to new and established teachers since 2007. Her own research has focused on the use of play and collaboration in computer science. She enjoys supporting practicing teachers by designing accessible lesson planning materials to get students engaged in computer science through play and collaboration. Paula has also produced educational materials for The Raspberry Pi Foundation, been a secondary school teacher, written a successful book (Hello App Inventor!) and has previously worked in IT project management for a media company and for the NHS.