# Lesson plan - “Slider number”

## Using the MonkMakes Slider for micro:bit component

## Introduction

This lesson gives you time to introduce the slider to the learners. The particular focus of the lesson is comparison of implicit and explicit data casting. In addition they will learn how to program the LED display on the micro:bit in response to input from the slider.

## Learning objectives

* Understand the purpose and functionality of the slider
* To successfully set up a micro:bit/slider number circuit and flash code to the micro:bit
* Apply knowledge of type conversion
* To be able to read and interpret a program containing a while loop
* To be able to read and interpret a program containing data type casting
* To be able to successfully modify a program containing multiple programming constructs

## Keywords

while, for, variable, function, in-built, user-defined, subroutine, explicit, implicit, type, type-casting, integer, float, string

## 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 number” 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/239dtxk3

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)
* 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 number**

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 8 mins | **Data type revision**  The slide that greets the student puts the focus of the lesson in context i.e. it is about data type conversion. The students are then given a scaffolded worksheet which enables them to share what they already know about data types, with a challenge question to determine if any have prior knowledge of type addition. After the students have completed the worksheet quickly go through the solutions and ensure that they correct any wrong answers on their starter worksheets. |
| Focus tasks 10 mins | **Data type consolidation**  In this section the students are given some explicit teaching about checking data types as well as implicit versus explicit data casting. The students are also asked if it is possible to cast a string and given the opportunity to discuss the decision to cast a number as a string. |
| Activity 1 8 mins | In order to put the lesson and the slider number lesson in context the students need an introduction to the slider itself. This is done at quite a simple level and introduces the idea that the number signal output by the slider always needs some intervention by the programmer to make it usable in output.  **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:  This program displays a single digit between 0 and 9 depending on the position of the slider. |
| **Activity 2**  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. |
| Activity 3 10 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.  As it is a very short program there are just two levels of the code conversation:  Level 1 - basic (some less challenging code lines to complete), Level 2 - intermediate (whole program to interpret)  Using the slides talk through the different steps in the program. |
| Activity 4 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:-  Change the name of any of the variables or parameters  Produce a happy picture on the LED display for a number less than 5 and a surprised picture on the LED display for a number 5 or more  Produce a number for any readings less than 8 or a surprised picture for 9.  If you have any students who want to explore the use of images on the micro:bit in response to the microphone input this page provides a list of images: https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/images.html |
| **Plenary**  6 mins | **Slider number plenary worksheet**  Students complete the plenary worksheet testing knowledge of type casting. |
| **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.