# Lesson 3:

Introduction

The class will be introduced to capturing user input and using that input to control a simple application made in Turtle. They will learn about data types and what each type can be used for.

## Learning Objectives

How to use edublocks

How to capture user input

How to identify different data types.

## Key Vocabulary

Sequence, selection and iteration, data types (string, integer, float)

## Preparation

#### Subject Knowledge

An understanding of creating code in a block based environment and understanding how key coding concepts are across all languages and can be illustrated using different languages.

Possible Misconceptions

User input is just keyboard input

That data types are not important.

Logical thinking is difficult.

#### Pedagogy

Ideally the class will each have access to a computer and complete the tasks individually. The lesson can be completed with 1 computer per 2 children.

#### You will need

A computer running Windows / Mac or Linux or Chromebook

A web browser (Firefox, Edge, Google Chrome, Safari)

## Assessment Opportunities

Understand that coding concepts are not tied to just one language and that skills learnt in one language can be applied to others.

Understanding the logical operators (slide 8).

Understanding data types and where they are best used.

## Outline Plan

This is a quick overview of the accompanying slide deck.

| Introduction  Group Activity  (Slide 3)  5 Minutes | Challenge the class to identify as many forms of input as possible. Including newer forms of input.  Write them down on the board.  For example  Keyboard, mouse, voice, joypad, XBOX Kinect, Wiimote etc.  In Python, user input is typically referring to keyboard input. |
| --- | --- |
| Group activity  (Slide 4)  5 Minutes | To start Edublocks, ask the class to open a web browser and type in  app.edublocks.org/editor  Select Python 3 as the mode, give the project a file name and click Create  The class are tasked with creating the code from this slide and then running the code.  The for loop (for i in range) is deliberately going to fail. As the user input is captured as a string, which cannot be used to define the number of times that the loop iterates. |
| Group Activity  5 Minutes  Slides (5 - 6) | Here we debug the error.  This is a type error, identifying that we used an incorrect data type for the loop. But the Python error is quite scary.  The three key data types we are focusing on are   * String: A sequence of characters, including punctuation, numbers and letters. * Integer: A number with no decimal place. * Float: A number with a decimal place.   Challenge the class to identify the best data type for a situation.  For example  “Which data type would be useful for currencies / temperature / scientific data?” = float  “Which data type would be useful for a message?” = string  “Which data type would be useful to count the number of buses on a road?” = integer |
| Group Activity  5 Minutes  (Slide 7) | Here the class will fix the issue in their code.  The easiest method to fix this is to pull the input() block out, then drop the int(1) block from Statements into the now vacant space. Then drop the input() block into the int(1) block. |
| Group Activity  5 Minutes  (Slide 8) | Here we introduce conditional statements, tests that will use logic to determine an outcome.  In this example the class will see that if the user types in Ed, or Aijaz, then the code will greet them by name. But if they use another name, then the else condition is used, and a generic hello is printed. |
| Group Activity  5 Minutes  (Slide 9) | Conditional statements / tests use logical operators for comparison.  ==  Values must be equal.  !=  Values must NOT be equal.  <  Left value is less than the right value.  <=  Left value is less than or equal to right value.  >  Left value is greater than the right value.  >=  Left value is greater than or equal to right value.  These logical operators will provide lots of functionality for future projects. |
| Group Activity  10 Minutes  (Slide 10) | The class should work through this example to get used to how conditional statements work and how logic is applied. |
| Group Activity  10 Minutes  (Slides 11-12) | In this challenge the class are tasked with fixing this broken sequence of code (slide 10).  The solution is shown on slide 12.  The class is asked to make one change at a time.   * What happened? * Does it work as they expected? * How can they fix it? |
| Plenary  2 Minutes  (Slide 13) | Here we recap the learning from this lesson.   * How we captured user input. * How we modified the data type for use in specific tasks. * How we used conditional statements to change the outcome of the code. * How logical operators are used in conditional statements. |
| Next Time  1 Minute  (Slide 14) | We will continue our learning with edublocks, and learn how to create variables that will store data, and then use the data with conditional statements. |