# Lesson 1 – What is micro:pyhon?

|  |  |
| --- | --- |
| The Big Picture – Why Is This Relevant? | Learning Objectives |
| This introductory lesson will take learners through what they will be learning for the next 24 lessons and will introduce some of the concepts around project based learning which may be new to them.  It will introduce IDEs which enable you to create, edit and test code. This lesson looks at the main features and introduces two basic features, scroll text and using images. | * Learners will understand what the course involves, what they will be learning and why it is important * Learner will understand the context of programming physical devices and how it relates to careers in STEM * Learners will know what an IDE is * Learners will be able to scroll text and display images on the micro:bit |
| Engagement – How Can I Engage Learners? | Assessment for Learning |
| This is the first lesson of a large course and so learners should be interested in what is to come as well as what they will be doing. For some this may be their first experience of a textually derived language and this may motivate them. They will be able to create their first text based program which will display an image on the micro:bit. | **Expected Progress:**   * All learners will be familiar with the layout of the course and the difference between the theory and project lesson. Learners will understand what the ‘success criteria’ is for * Learners will be able to scroll the ‘Hello, World’ message and display a demonstration image   **Good Progress:**   * Learners will understand the differences and similarities between the examples of block based and text based code * Learners will be able to scroll their own message and display an image of their choice   **Exceptional Progress:**   * Learners will confidently translate block based code examples into Python * Learners will be able to display and scroll two or more messages |
| Links to KS3 Programme of Study | |
| * use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems | |
| Key concepts | Key words |
| * Code can be written in both text and blocks, they do the same thing but more can be done with actual code * Text based languages have a specific syntax that must be used, this is not as forgiving as blocks but the concepts are the same * Indentation really matters! * Case sensitivity * What is an IDE * Scrolling text and displaying images | * Python * MicroPython * Text based language vs blocks * Indentation * Case sensitivity * Power cycle (turn it off and on again) * IDE |
| Differentiation | Resources |
| Some learners will be anxious about the transition to text based languages and so will require more support initially. It is typical for learners to struggle to translate the techniques that they may have been comfortable with in MakeCode into Python.  Most Learners will be able to follow the instructions  however, adding a program to a microcontroller  maybe a new concept to some Learners and they may  need support with getting the files onto the micro:bit  initially. | * Access to the micro:bit IDE or equivalent IDE * Lesson 1 ppt * Lesson 1 Activity Sheet * L1\_scroll\_text python file * 1 micro:bit per Learner * 1 USB cable to connect the micro:bit to a PC * A PC * Access to <https://python.microbit.org/v/1.1> |
| Lesson Flow | |
| * Introduce the course to the class and explain that it is a mix of theory, activities and project * Explain that the theory lessons will have tasks and activities and will always us the same format * Go through the activity sheet and explain what each section is for. Ensure learners understand what the success criteria are as this is vital throughout the course. Explain that the course has several projects that will be done in groups and that communication, collaboration and discipline will be required to complete the tasks successfully * Introduce the concept of a textually derived language and see if learners have heard of any others * Ensure that learners understand the difference between mark-up (HTML, CSS) and actual programming languages such as Python, C++, PHP etc. Explain that the same things (and more) can be done in Python that can be done with blocks and look at the comparisons in the slides * Get learners to complete the first task on the activity sheet * Introduction to what an IDE is * Show Learners how to log on and open the IDE * Teacher demonstration of scrolling ‘Hello, World’ program * Learners try and download to their micro:bit * Teacher demonstration of displaying images on the micro:bit * Learners try and download to their micro:bit * Learners work through the remainder of the activity sheet independently; teacher intervenes where appropriate * Encourage more advanced Learners to attempt the stretch tasks once they complete main task * Learners complete Activity Sheet * Teacher recap of the features of an IDE | |
| Making | |
| There is no making in this lesson | |