

INTRODUCTION TO PYTHON – OVERVIEW

Overview

This unit will continue to develop students' understanding of programming whilst introducing them to the programming language Python. The unit progresses concepts such as inputs and outputs, events, variables, logical tests and loops through one of the world's most popular programming languages.

In this unit, students will learn what Python is and how this text-based type of programming differs from the block-coding they are used to. They will learn how to create a series of different programs that achieve different results and become more familiar with the nuances of working with text-based programming.

National Curriculum

The areas of the National Curriculum and the key computing concepts covered in this unit are outlined in the table below:

Knowledge and understanding	Computing concepts
To write and debug programs that accomplish specific goals.	Debugging
To solve problems by decomposing them into smaller parts.	Decomposition
To use sequencing and repetition in programs.	Sequencing, repetition and looping
To work with variables.	Variables
To work with inputs and outputs.	Inputs and outputs
To use logical reasoning to explain how simple algorithms work.	Reasoning and algorithms
To detect and correct errors in algorithms and programs.	Debugging

The Computing Curriculum

Where the knowledge and understanding developed in this unit fits into the computing curriculum is outlined in the table below:

Prior Learning	Future Learning
EYFS Unit: Introduction to programmable robots	Key Stage 3 Computing.
Year 1 Unit: Introduction to algorithms unplugged	
Year 1 Unit: Programming virtual Beebots	
Year 2 Unit: Debugging Algorithms	
Year 2 Unit: Programming using Scratch Jnr	
Year 3 Unit: Introduction to Scratch	
Year 4 Unit: Introduction to micro:bits	
Year 5 Unit: Advanced micro:bits	
Year 6 Unit: Programming a computer game in Scratch	

Cross-curricular links and extension activities

This unit provides opportunities for cross-curricular links through maths creating simple calculators and solving simple calculations using BODMAS.

Required resources

There are plenty of free Python editors that can be installed to enable you to write and test Python code.

By far the easiest editor to install is Python IDLE as it requires little to no additional system configuration.

You can find Python IDLE in the Microsoft store if you're using a Windows operating system or alternatively at the following link [Download Python | Python.org](https://www.python.org/downloads/) if you need it for other operating systems.

Python IDLE is designed for beginners and small scripts making it highly suitable for this unit of work.

Other alternatives can be found in the table below.

Editor	Notes
<i>Visual Studio Code</i>	A generic code editor maintained by Microsoft but available free of charge (don't confuse it with Visual Studio, which is a completely different program – see below).
<i>PyCharm</i>	An editor devoted to writing Python code. Reviews online suggest that it can be very slow to work with, and some users will need to upgrade to the paid Premium edition.
<i>Visual Studio</i>	If you already spend time working in Visual Studio, you may find it easiest to use this as your development environment (although it's a bit of a big beast!).
<i>Jupyter Notebooks</i>	If you work in machine learning or AI you may well choose this powerful coding environment.