First Steps Together
Curriculum Unit Overview / Intent

Year Group: KS3 (Year B)	Term: Summer 2

Computing: Programming Python

Key Concepts

Sequence: instructions run in a precise order.
Variables: storing and updating values.
Input and output: making programs interactive.
Selection: using `if`, `elif` and `else` to make decisions.
Repetition: using loops to repeat actions efficiently.
Boolean logic: using `and`, `or` and `not` in decisions.
Algorithms: following and comparing step-by-step solutions.
Debugging: identifying, testing and correcting errors.

Prior Knowledge

Basic understanding of what a program is.
Awareness that computers follow instructions exactly.
Experience of block-based coding or simple digital instructions.
Ability to follow a sequence of steps in a logical order.
Some familiarity with typing, using a keyboard and saving work.
Basic understanding of variables as labels for information, if previously introduced.
Awareness that errors can happen and programs can be improved.

Cross-Curricular Links

Maths: sequences, logic, variables, comparison operators, data handling.
English: clear instructions, precise vocabulary, reading code carefully.
Science: modelling systems, testing ideas, cause and effect.
Design and Technology: solving problems through iterative design and testing.
PSHE: resilience, teamwork, safe and responsible use of technology.

SMSC & British Values

Spiritual: recognising patterns, problem-solving and the satisfaction of improving code.
Moral: understanding responsible use of technology and the impact of digital actions.
Social: collaborating, sharing ideas and giving constructive feedback.
Cultural: exploring how programming supports different industries and communities.
Democracy: listening to others' ideas when comparing solutions and making choices.

Assessment

Teacher observation of program creation, testing and refinement.
Retrieval practice at the start of lessons to revisit prior learning.
Exit tasks to identify whether pupils can explain what their code does.
A practical end-of-unit task requiring a short program using sequence, selection, repetition, variables, input/output and simple Boolean logic.

Adaptations

Chunk instructions into small, manageable steps with one task at a time.
Provide code templates, word banks and sentence starters.
Use worked examples and partially completed programs.
Reduce cognitive load by limiting new syntax per lesson.
Pre-teach key vocabulary and symbols.
Offer visual supports, colour coding and annotated code.
Use paired programming with carefully matched partners.

Lesson Objectives

1. To explain how Python instructions are stored and executed, and to write a simple sequence of commands that produces output.
2. To use variables, input and output in Python to store and change information in a program.
3. To use selection in Python to create decisions using comparison operators and simple Boolean logic.
4. To use repetition in Python to reduce duplicated code and make a program more efficient.
5. To create, test and debug a Python program by applying a systematic problem-solving approach.
6. To design, evaluate and improve a Python program that combines sequence, selection, repetition, variables, input/output and logical reasoning about alternative solutions.

Links to Future Learning

Building more complex Python programs with nested selection and loops.
Developing games and interactive stories.
Using functions and modular programming.
Working with lists, strings and simple data structures.
Understanding algorithms in greater depth, including sorting and searching.

Key Vocabulary

Program
Instruction
Sequence
Variable
Value
Input
Output
Selection
Condition
Boolean
`and`
`or`
`not`
Repetition
Loop
Count-controlled loop
Debug
Error
Test
Algorithm
Abstraction
Comparison operator
`if`
`elif`
`else`
Syntax

Possible Enrichment:

Use Python to model a real-world scenario, such as a pet feeding reminder or simple calculator. Add challenge extensions such as score tracking or multiple decision paths. Evaluate two different algorithms for the same problem and justify the better choice.