

<u>Unit: Introducing Python</u>				
<p>The unit will focus around the understanding and use of Python to create programs. The students will cover a number of areas within the unit including variables, input, print and IF statements. The unit will end in two different forms of assessment.</p>				
	Lesson Outline	Activities	Keywords	H/W
<u>Lesson</u> <u>1</u>	<p>“Introduction”</p> <p>Lesson Objectives:</p> <p>LO1 – Create simple code including the input and print scripts.</p>	<p>Starter Video about programming</p> <p>Main Simple coding activity</p> <p>Plenary Recap the key words that have been introduced at the start of the lesson</p>	<p>Python Input Print</p>	
<u>Lesson</u> <u>2</u>	<p>“Variables/Inputs”</p> <p>Lesson Objectives:</p> <p>LO1 – Create variables and declare the data types for variables.</p>	<p>Starter Recap on simple programming skills from previous lesson</p> <p>Main Students to complete their first full program making use of numerous input scripts and outputting the information in a sentence using placeholders</p> <p>Plenary Fixing errors worksheet</p>	<p>Data Types Variables Strings Input</p>	<p>Complete section 1 of the python section on the code academy website</p>
<u>Lesson</u> <u>3</u>	<p>“Using Numbers”</p> <p>Lesson Objectives:</p> <p>LO1 – Use operators to effectively create programs to solve a number of</p>	<p>Starter Find out what data types are and list 3 different data types</p> <p>Main Complete the Integer Tasks worksheet</p>	<p>Integers Data Types</p>	<p>Complete section 2 of the python section on the code academy website</p>

	scenarios.	Plenary Kahoot Quiz		
<u>Lesson 4</u>	<p>“Inputs and Integers”</p> <p>Lesson Objectives:</p> <p>LO1 – Combine strings and integers to create a small program</p>	<p>Starter Python errors worksheet</p> <p>Main Moon Pig birthday card message</p> <p>Plenary Tell me three things</p>	Input Integers	
<u>Lesson 5</u>	<p>“IF Statements”</p> <p>Lesson Objectives:</p> <p>LO1 – Understand what an IF statement does</p> <p>LO2 – Make use of IF statements in your programs</p>	<p>Starter Complete research into what an IF statement is and what they are used for</p> <p>Main IF statement worksheet</p> <p>Plenary Q & A</p>	If Else If Else Selection	Complete section 3 of the python section on the code academy website
<u>Lesson 6</u>	<p>“Loop the Loop”</p> <p>Lesson Objectives:</p> <p>LO1 – Understand what a loop is used for when programming</p> <p>LO2 – Be able to make effective use of the WHILE statement</p>	<p>Starter Complete research into what a loop is and how a while loop is used</p> <p>Main Alton Towers task While loops worksheet</p> <p>Plenary Peer testing of program</p>	Loops While Iteration	Complete section 4 of the python section on the code academy website
<u>Lesson 7</u>	<p>“Selection and Iteration”</p> <p>Lesson Objectives:</p> <p>LO1 – Understand what is meant by selection and Iteration</p> <p>LO2 – Combine all programming techniques</p>	<p>Starter Make improvement from test table completed in previous lesson</p> <p>Main Alton Towers task While loops worksheet</p> <p>Plenary Kahoot Quiz</p>	Selection Iteration	

Year 8 Scheme of Work – Python

	used so far.			
<u>Lesson 8</u>	<p>“Procedures”</p> <p>Lesson Objectives:</p> <p>LO1 – Understand what a procedure is</p> <p>LO2 – Be able to make effective use of a number of different procedures</p>	<p>Starter Think, Pair, Share about skills the students have learnt so far</p> <p>Main Alton Towers task While loops worksheet</p> <p>Plenary Revision Quiz</p>	Procedures Organisation Selection Iteration	Revise
<u>Lesson 9</u>	<p>“Assessment”</p> <p>Lesson Objectives:</p> <p>LO1 – To complete the end of cycle assessment based making use of the programming techniques previously used</p>	<p>Starter Introduce to programming test Rules of programming assessment</p> <p>Main Complete the end of unit programming assessment</p> <p>Plenary Hand in Test</p>		
<u>Lesson 10</u>	<p>“End of Unit test”</p> <p>Lesson Objectives:</p> <p>LO1 – To complete the end of unit written test to check theory understanding</p>	<p>Starter Rules of written test</p> <p>Main Complete the end of unit test</p> <p>Plenary Mark and hand in tests</p>		

LEVELS AND PROGRESS

Excellent	Secure	Developing
<ul style="list-style-type: none"> • Recognise pieces of code that can be used to solve two or more similar problems • Can appropriately select pieces of pre-prepared code, combining them with my own code, to solve a problem • Can translate a specification into the code required to produce a solution • Can rationalise if Python is an appropriate piece of software to solve a given problem • Can justify when and why I would Python as a solution, and outline the benefits and limitations of the software • Can explain the meaning of text in an Python program, without necessarily using the Python interface • Be able to write programs in a text-based language and be able to create your own data structures. 	<ul style="list-style-type: none"> • Can use and manipulate numeric variables and string variables in my own programs. • Can use and manipulate lists and/or tuples in my programs. • Can use Python code conventions (including importing modules) accurately, and vary the rules within the programs. • Can write a short program and then use Python to code and execute the program, knowing the limitations/suitability of the software. • Can write procedures with parameters and functions that returns a value • Can justify when and why I would Python as a solution • Correctly use variables, lists and simple procedures in your programs. • Use selection and repetition correctly in your programs. 	<ul style="list-style-type: none"> • Can recognise different types of data: Integer, Float, Long and Complex • Can code using syntax and typography, carefully and precisely. (i.e. text, language and symbols the computer will understand) • Can code using selection and iteration, such as for and while loops • Can describe the difference between numeric and string variables • Can describe what a simple program given to me will do, and be able to describe some of the coding that I will need in my programs • Can run and test a simple program given to me, and then make a similar one of my own using the same ideas • Be able to plan a sequence of instructions for something that you want to happen.