UNIT
8.2

PYTHON MAGIC

Unit Overview

Target year group: Year 8
Projected Duration: 6-8 lessons
This scheme of work will give learners a basic understanding of the Python programming language. Through a series of six lessons learners will be introduced to the basic Python syntax and look at selection, loops, functions, lists and classes. Learners will be given the building blocks to extend their programming skills by adding to existing programs or extending their knowledge on Codecademy.

Relevant Prior Knowledge

Learners will have been introduced to visual programming languages in Year 7 in the form of Scratch and Kodu.

New Language for Learning

- ↘ *Syntax*
- ↘ *Variable*
- ↘ *String*
- ↘ *Selection*
- ↘ *Function*
- ↘ *Class*
- ↘ *Integer*
- ↘ *Float*
- ↘ *Loop*
- ↘ *Boolean*
- ↘ *List*

Resources

- ✓ *Python Magic - Teacher's PowerPoint*
- ✓ *Caesar Shift Wheel – cut out worksheet*
- ✓ *Text Adventure – guide sheet*
- ✓ *Animals.py – Example Python program*

Homework

Learners will need to create an account on www.codecademy.com and work through the exercises on Python. At a minimum they should complete the first section on Python Syntax.

Cross-Curricular Links

Numeracy:
- ↘ *Learners will create a calculator to carry out some basic arithmetical functions.*

Literacy:
- ↘ *Learners will create an interactive story in the form of a text-adventure game.*

History:
- ↘ *Reference to encryption using the Caesar shift – links can be made between this and the cryptography carried out in World War 2.*

Assessment

Assessment is based on the Progression Pathways Assessment Framework from Computing at School.

Pathway: Programming & Development

Level 4	☑ *Understands and uses if and if, then else statements.* ☑ *Uses a variable and relational operators within a loop to govern termination.* ☑ *Designs, writes and debugs modular programs using procedures.*
Level 5	☑ *Has practical experience of a high-level textual language, including using standard libraries.* ☑ *Uses a range of operators and expressions.* ☑ *Selects the appropriate data types.*
Level 6	☑ *Uses and manipulates one dimensional data structures.* ☑ *Detects and corrects syntactical errors.*

Support and Extension

All tasks are introduced with examples for learners to build upon, this should result in all learners being able to create their own simple programs.

***Low ability learners** may need tasks broken-down further by the classroom teacher with a higher emphasis on examples. Lesson 6 should be omitted for low ability groups.*

*Extension activities are built into individual lessons with opportunities to access Codecademy to further extend knowledge of **more able learners**. Teachers may wish to offer challenges from http://projecteuler.net to challenge exceptionally able and enthusiastic learners.*

Lesson 1

Activities:

- ↘ *Teacher introduces the topic – show video [5:44]*
- ↘ *Introduce the Learning Objective and explain what a String Variable is used for.*
- ↘ *Show the progression of code for the 'Hello World' program on the board – learners copy and create their own programs.*
- ↘ *Learners add to their programs so they can have a simple conversation with the computer.*

Extension:

- ★ *Learners should create an account on Codecademy (www.codecademy.com) – they should begin working through the Python course called 'Python Syntax'.*

Learning Objective:

Understand and be able to use basic Python syntax.

Learning Outcomes:

- ✓ *GOOD: Create the basic 'Hello' program and display different outcomes depending on different responses.*
- ✓ *BETTER: Add to the 'Hello' program so it can have a basic conversation with the user based on yes/no answers.*
- ✓ *BEST: Create an account on Codecademy and begin the 'Python Syntax' course.*

Lesson 2

Activities:

- ↘ *Recap on the data type 'String'. Explain what is meant by 'Integer' and 'Float'.*
- ↘ *Display code for the start of the 'Calculator' program on the board – learners to offer suggestions about what they think is happening in each line of code.*
- ↘ *Explain what is meant by 'Selection' using the example from the PowerPoint.*
- ↘ *Show the extended example of the 'Calculator' program with selection included for the operator.*
- ↘ *Learners should copy the example and add to it so that it is possible to carry out multiplication and division within the program.*

Learning Objective:

Understand how to use selection to create a simple calculator.

Learning Outcomes:

- ✓ *GOOD: Create the basic 'Calculator' program to allow the user to add and subtract.*
- ✓ *BETTER: Add to the 'Calculator' program to allow the user to carry-out multiplication and division.*
- ✓ *BEST: Complete additional levels of the 'Python Syntax' course on Codecademy.*

Lesson 3

Activities:

- ↘ *Explain what is meant by a 'Boolean' variable.*
- ↘ *Explain what is meant by a 'While Loop' and go through the example from the PowerPoint.*
- ↘ *Go through the code for the 'Guess the Number' game with learners giving an input on the purpose of each part.*
- ↘ *Learners create their own 'Guess the Number' game then play this to test it works.*

Learning Objective:

Understand how to create a loop in Python and use a Boolean variable to define when to stop.

Learning Outcomes:

- ✓ *GOOD: Create a working 'Guess the Number' game.*
- ✓ *BETTER: Complete the 'Python Syntax' course on Codecademy.*
- ✓ *BEST: Start working through the 'Strings and Console Output' course on Codecademy.*

Lesson 4

Activities:

- ↘ *Explain what is meant by a function.*
- ↘ *Display and discuss the code for the 'Abracadabra' program, teacher should question learners to give an input into what they think each part of the code does.*
- ↘ *Learners create their own 'Abracadabra' program and run it to test it works. Introduce the 'Reverse the Word' program, learners should copy the example from the PowerPoint then add to it such that the program displays an output indicating whether or not the word is a palindrome.*

Learning Objective:

Understand how to use and call functions.

Learning Outcomes:

- ✓ *GOOD: Create a program to call a function to pull a virtual rabbit from a hat.*
- ✓ *BETTER: Create a program that will reverse a given word.*
- ✓ *BEST: Add to the 'reverse the word' program such that the user is given feedback stating whether or not the word is a palindrome.*

Lesson 5 *(likely to take more than 1 lesson)*

Activities:

- ↘ *Explain what is meant by a list in Python.*
- ↘ *Go through the example of a list, question learners on what they think each print statement will do (don't tell them the answers). Learners copy the program and run it to discover what each print statement does and feed this back to the class.*
- ↘ *Explain what is meant by a Caesar Shift (perhaps give an example on the board if learners don't understand).*
- ↘ *Learners cut out and put together their own Caesar Shift wheel and use it to encode and decode messages.*
- ↘ *Learners should use their Caesar Shift wheel to decode the message from the PowerPoint (with no key given). If they get stuck perhaps discuss how many words have only one letter and use this as a starting point.*
  Learners copy the example Caesar Shift encryption program and try encrypting some text.

Learning Objective:

Understand how to store and look up values from a list.

Learning Outcomes:

- ✓ *GOOD: Create a list of students and carry out various functions relating to it.*
- ✓ *BETTER: Create a Caesar Cipher Wheel and use it to encode and decode messages (even when the key isn't given).*
- ✓ *BEST: Create a python program to encrypt text using the Caesar Cipher.*

Lesson 6 *(likely to take more than 1 lesson)*

Activities:

- ↘ *Explain what is meant by a class and go through the Animal class example. (Discuss attributes that are common to all animals before moving on.)*
- ↘ *Learners should open up the example Python program, Animals.py and look at the code (reassure them that it is okay if they don't understand the code as this is quite advanced.)*
- ↘ *Show learners the code to add new instances of animals to the program and display their details. (They can choose their own animals.)*
- ↘ *Explain that learners will next be using what they've learned about classes to create a text-adventure game. Play the Zork game with the class to give them an idea of how a text-adventure game works.*
- ↘ *Learners should use the design sheet to plan their own text-adventure game, they can then use the instructions to create their own game.*

Learning Objective:

Recognise how a class can be used to create custom objects.

Learning Outcomes:

- ✓ *GOOD: Add to the Animals program to create instances of new animals and display their details.*
- ✓ *BETTER: Design own text-adventure game and begin to build it using Python.*
- ✓ *BEST: Create and test a working text-adventure game.*