



Purple Mash Computing Scheme

Introduction to Python

Contents

Overview

- Key Learning
- Knowledge Organiser and Learning Mat
- Concept Map and Quiz

Lesson 1: Text Output

- Aim
- Success Criteria
- Resources
- Preparation
- Teaching Notes

Lesson 2: Python and Mathematical Calculations

- Aim
- Success Criteria
- Resources
- Preparation
- Teaching Notes

Lesson 3: Using Repeat

- Aim
- Success Criteria
- Resources
- Preparation
- Teaching Notes

Lesson 4: Moving Sprites

- Aim
- Success Criteria
- Resources
- Preparation
- Teaching Notes

Assessment

Need more support? Contact us:

Tel: +44(0)208 203 1781 | Email: support@2simple.com | X: [@2simplesoftware](https://twitter.com/2simplesoftware)

Overview

This unit introduces children to text-based coding through the use of Python. Children will learn some of the key language rules for Python. By the end of the unit, children will have outputted text to the screen, used loops to repeat code and made a sprite move by coding it.

Key Learning

- To understand that Python is a text-based coding language and enter simple commands.
- To use Python to perform mathematical calculations.
- To be able to use repetition and Python library functions.
- To understand how Python in Pieces can be used to program a graphical user interface.

Knowledge Organiser and Learning Mat

These resources can be used to support learning across the unit.

- [Knowledge Organiser](#)
- [Learning Mat](#)

Concept Map and Quiz

Use these resources to gauge children's understanding before or after the unit has been taught.

- [Concept Map](#)
- [Quiz](#)

Lesson 1: Text Output

Aim

- To understand that Python is a text-based coding language and enter simple commands.

Success Criteria

- I can explain what Python is and compare it to block-based coding.
- I know Python has strict rules that must be followed for code to work.
- I can use quotes, brackets and capitals correctly in my code when printing messages to screen.
- I can debug errors in simple commands.

Preparation

- For all activities, children to navigate to the **Python in Pieces** application found in the tools area of Purple Mash.

Please find all lesson resources and additional teaching notes below.

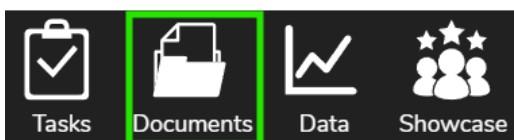
Resources

- [Slideshow - Lesson 1](#)
- [Python in Pieces](#)

Teaching Notes

For slide 14, 19 and 21, ensure that children open Freecode Project within Python in Pieces and save any work in a class folder with a suitable name.

All work children save in a class folder within Python in Pieces can be accessed through the documents icon found in the toolbar of Python in Pieces.



Lesson 2: Python and Mathematical Calculations

Aim

- To use Python to perform mathematical calculations.

Success Criteria

- I can type simple number calculations in Python.
- I can use Python to solve maths problems.
- I can debug errors in mathematical commands.

Preparation

- For all activities, children to navigate to the **Python in Pieces** application found in the tools area of Purple Mash.

Please find all lesson resources and additional teaching notes below.

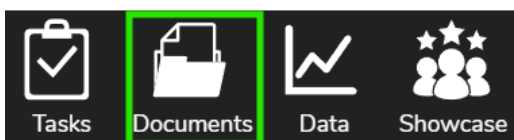
Resources

- [Slideshow - Lesson 2](#)
- [Python in Pieces](#)

Teaching Notes

For slides 14 and 17, ensure that children open Freecode Project within Python in Pieces and save any work in a class folder with a suitable name.

All work children save in a class folder within Python in Pieces can be accessed through the documents icon found in the toolbar of Python in Pieces.



Lesson 3: Using Repeat

Aim

- To be able to use repetition and Python library functions.

Success Criteria

- I can explain why repetition is used in some computer programs.
- I understand what a 'for loop' is and can use 'for loops' in my programs.
- I can import a function from the Python library.
- I can use the sleep function to control the speed of my code.

Preparation

- For all activities, children to navigate to the **Python in Pieces** application found in the tools area of Purple Mash.

Please find all lesson resources and additional teaching notes below.

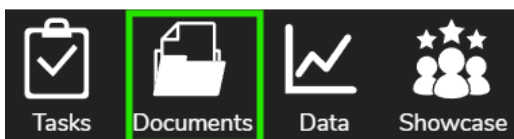
Resources

- [Slideshow - Lesson 3](#)
- [Python in Pieces](#)

Teaching Notes

For slides 12, 16 and 18, ensure that children open Freecode Project within Python in Pieces and save any work in a class folder with a suitable name.

All work children save in a class folder within Python in Pieces can be accessed through the documents icon found in the toolbar of Python in Pieces.



Lesson 4: Moving Sprites

Aim

- To understand how Python in Pieces can be used to program a graphical user interface.

Success Criteria

- I can explain what a graphical user interface is.
- I know what a sprite is and how to change its position on screen.
- I can use 'sleep' function to control the timing of movement.

Preparation

- For all activities, children to navigate to the **Python in Pieces** application found in the tools area of Purple Mash.

Please find all lesson resources and additional teaching notes below.

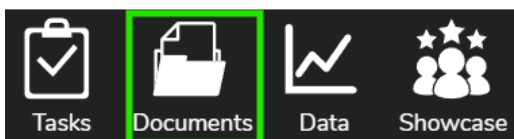
Resources

- [Slideshow - Lesson 4](#)
- [Python in Pieces](#)

Teaching Notes

For slides 9 and 17, ensure that children open Freecode Project within Python in Pieces and save any work in a class folder with a suitable name.

All work children save in a class folder within Python in Pieces can be accessed through the documents icon found in the toolbar of Python in Pieces.



Assessment

The following information is an exemplar of what a child at an expected level would be able to demonstrate when completing this unit with additional exemplars to demonstrate how this would vary for a child with emerging or exceeding achievements.

Emerging

- Children can compare the main visual differences between text and block-based coding. They recall some of the software that the Python coding language is used to create in the everyday world. Children can write simple commands that print outputs to the screen. With support, they can debug syntax errors in their lines of code such as failing to use quotes for strings or forgetting to close brackets. They can write code for a looping code structure and can determine how many times it loops. With support, children can print concatenations of variable values and strings to control the presentation of output on the screen. With support, children can import the sleep function and use it to add pauses in the program.

Expected

- Children can explain the difference between text and block-based coding languages. They recognise the uses of Python and can explain why it is more adaptable and flexible than using block-based coding. Children know that Python has strict text formatting rules that must be followed for programs to run without errors. Children can use some of the key syntax conventions to enter simple commands that produce outputs. They can write code that prints strings, variable values and calculations. Children explore writing the code for simple loops and control the speed of execution using sleep commands. They can control aspects of the graphical user interface using Python coding.

Exceeding

- Children can explain the difference between text and block-based coding languages. They can identify the advantages of using a text-based coding language such as Python over block-based coding. Children can use the syntax rules of Python to write commands that produce outputs. They can demonstrate how functions including print(), range() and sleep() can be modified to produce the required coding logic. Children are experimental with their coding and purposefully manipulate functions to write code that produces varied outputs. They write loops that make use of sleep commands. They can control aspects of the graphical user interface using Python coding and extend themselves to create novel programs.