CODE KALEIDOSCOPE

Creative Code. Colourful Thinking.

Teacher Guide

A concise guide for running the Creative Coding unit with Mu Editor, Python Turtle, the Code Kaleidoscope drawing workspace and the student learning journal.

Unit Overview

Stage	Focus	Teaching purpose
Introduction	Code Kaleidoscope workspace	Predict, test and explain changes in a visual coding environment.
Lesson 1	Sequence	Connect Scratch-style commands to Python Turtle syntax.
Lesson 2	Variables	Change stored values to control creative output.
Lesson 3	Selection	Use if/else decisions to alter what happens.
Lesson 4	Iteration	Use loops to create repeated patterns and symmetry.
Lesson 5	Data types and random	Use integers, strings and random values in generative designs.
Lesson 6	Debugging	Find, explain, fix and improve creative code.
Final Project	Creative showcase	Combine concepts in a planned original Turtle artwork.

Recommended Lesson Flow

Lesson part	Suggested time	Teacher action
Do Now	3-5 minutes	Short retrieval or visual thinking task as students arrive.
Retrieval	5 minutes	Reactivate prior vocabulary and connect back to earlier lessons.
Predict	5-8 minutes	Students inspect code or output before running it.
Run	10-15 minutes	Students type code into Mu Editor and run it.
Investigate	8-10 minutes	Students identify which lines control the output.
Modify	10-15 minutes	Students change values with support and compare outcomes.
Make	15+ minutes	Students create a small independent outcome.
Plenary	5 minutes	Students explain what changed and which concept they used.

Using the Resources

Creative Coding Learning Journal

Print as the main student workbook. It includes Do Now tasks, PRIMM response areas, WAGOLL annotation, final project planning, assessment and DIRT response.

Lesson Slides

Use the PowerPoint deck to structure teacher modelling, retrieval, PRIMM stages and plenaries. Slides are deliberately concise for projection.

Code Kaleidoscope Workspace

Use the drawing workspace as a bridge into variables, input, selection, iteration and output before students type code in Mu Editor.

Teacher Evaluation Questionnaire

Use only during prototype testing. Keep consent paperwork separate, following university procedures.

Scaffolding Routes

Route	Teacher guidance
Supported	Use starter code, change one variable at a time, and use sentence stems such as "When I changed ___, the output ___".
Core	Complete PRIMM activities in order, compare outputs, and use vocabulary such as variable, loop, input and output.
Stretch	Combine shapes, variables, selection, loops and random values. Predict before changing code and justify design choices.

Cold Call / Check for Understanding Examples

- Which line of code controls the output?
- What do you predict will change if this value is larger?
- Which part of the program is repeated?
- What error did you fix, and how do you know it worked?
- Which concept did you use in your final design?

Final Project Guidance

The final project should feel like framed freedom: Bronze students adapt a starter, Silver students combine previous techniques, and Gold students design independently using variables, loops, functions, selection or input where appropriate.