Introduction to EduBlocks

EduBlocks lesson plan – Code Playground

Lesson overview

In this lesson, students will be introduced to EduBlocks, a visual programming tool that helps bridge the gap between Scratch and Python by showing both blocks and code together. Students will explore the Python turtle library, using blocks to create a colourful spiral pattern. They'll learn to use loops, variables, RGB colour values, and mathematical operations to control a turtle on screen and will begin to make the connection between blocks and Python syntax.

Time	Key learning outcomes	Resources
45 mins	• Understand what EduBlocks is and how it bridges block-based and text-based coding • Learn how to create a project using Python turtle in EduBlocks • Understand how RGB colours work in Python and how to set screen colours • Use loops, variables, and functions to control a turtle to draw a spiral • Experiment with code changes to observe effects on output.	• Laptops or desktop computers • Access to EduBlocks website - https://edublocks.org/

Content

Introduction

Begin with a brief discussion:

"Have you used Scratch before?"

"What did you like about using blocks to code?"

Real-life connection

Explain how programmers use text-based languages like Python in real life, and how EduBlocks helps us move from Scratch style blocks to real Python code.

Demonstrate the EduBlocks website, guiding students through steps to start a new project. Show them how to click on 'Start coding' then select 'Python 3'. Explain how to give their project a name before clicking the blue 'Create' button to begin coding. Explain the turtle concept: like a pen on a drawing board that follows instructions.

Scratch practical

Ask the children to log into EduBlocks and create a new project as described in the workbook.
Show the introduction to EduBlocks video as a guided lesson pausing regularly when the pupils need to catch up.

- Children should be able to follow along with the workbook or the guided lesson video
- By the end of the lesson children should be able to create a working program showcasing the introduction to EduBlocks project.

Activity – Introduction to EduBlocks

This project uses a turtle, which is like a pen on a drawing board that follows instructions.

Walk the children through importing the turtle and setting up the screen.

Show how Python uses RGB colour codes:

- 255, 0, 0 is Red
- 0, 255, 0 is Green
- 0, 0, 255 is Blue.

This project uses variables. Support the students to create a list of colours which are assigned to the colour's variable.

Scratch practical

Using the video and workbooks support the children to follow the instructions and complete the coding project. Have them think of other ways to enhance the project if they have extra time.

Activity wrap up

Prepare to share your project with the class

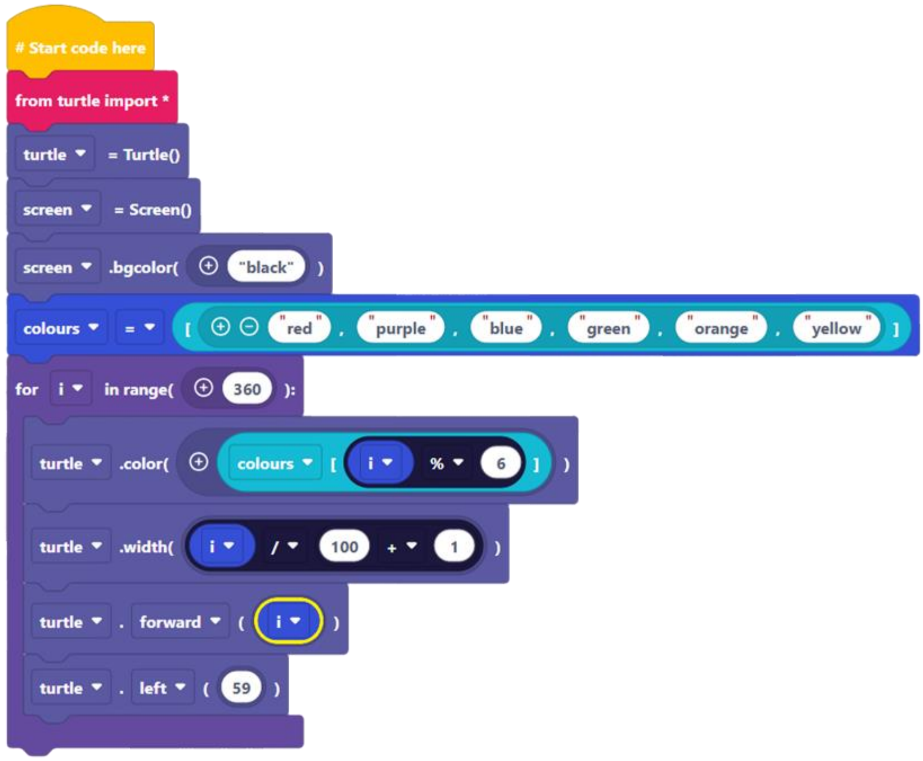
- What does 'i % 6' mean? (Explain modulus using division and remainder)
- How does changing the angle affect the shape?

Encourage customisation:

- Change the colours within the list or add more colours
- Experiment by changing the angle and the speed of the turtle.

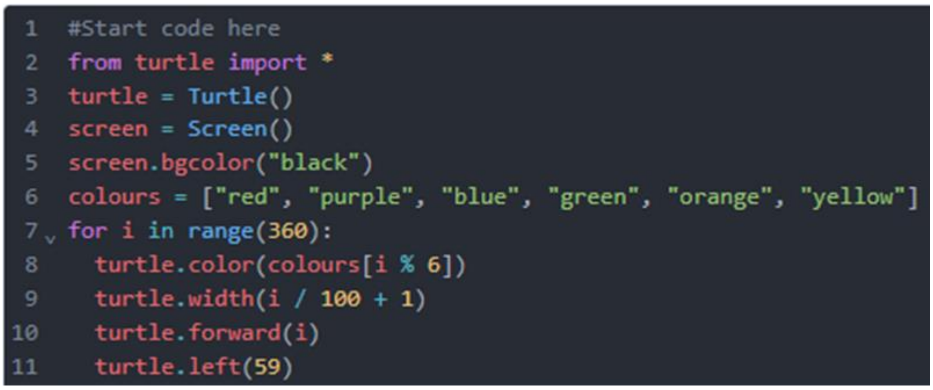
Code snippets

Block code



```
# Start code here
from turtle import *
turtle = Turtle()
screen = Screen()
screen .bgcolor( "black" )
colours = [ "red" , "purple" , "blue" , "green" , "orange" , "yellow" ]
for i in range( 360 ):
  turtle .color( colours [ i % 6 ] )
  turtle .width( i / 100 + 1 )
  turtle . forward ( i )
  turtle . left ( 59 )
```

Python code



```
#Start code here
from turtle import *
turtle = Turtle()
screen = Screen()
screen.bgcolor("black")
colours = ["red", "purple", "blue", "green", "orange", "yellow"]
for i in range(360):
  turtle.color(colours[i % 6])
  turtle.width(i / 100 + 1)
  turtle.forward(i)
  turtle.left(59)
```

Summary

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

Assessment guidance

Differentiation – Lower ability/ASN

- Focus on assembling blocks in the right order
- Support with recognising Python keywords like import, for, and range
- Used fixed values rather than exploring RGB or angle changes
- Allow paired work for additional support.

Differentiation – Higher ability/extension

- Challenge students to create their own shape using loops and angles
- Encourage editing the Python text to try coding without blocks
- Explore adding pen up and pen down features to create patterns with gaps.

Plenary

- What does the turtle do when we run the code?
- How does changing one number affect the whole drawing?
- What did you notice about the blocks and the Python code side by side?

Assessment questions

- Why do we use quotation marks around the colour names like “black” in our code?
- What does the modulus operator % do in the code block?
- What will happen if you change the turtle’s turn angle from 59 degrees to 90 degrees?