

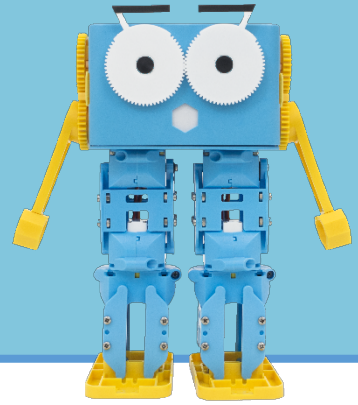
Lesson 2.2 – Scratch vs. Python

Education Level: Second/Third Level (Age 9-13)

Lesson Duration: 45 minutes

Prerequisite Knowledge: Lesson 2.1

Device Compatibility: Laptop or PC



Lesson Overview

In this lesson, students will be comparing Scratch to Python – a text-based programming language that is widely used in many disciplines including web, machine learning and robotics. Students will be drawing links between the two languages as they begin to prepare for writing their first Python script in future lessons.

Learning Objectives

- Describe what the main differences are between Scratch and Python
- Draw links between Scratch blocks and Python commands
- Put together printed out programming blocks to guide someone out of a maze

Key Vocabulary

- Python
- Text-Based Programming
- Scratch
- Block-Based Programming
- Commands
- Algorithm

Resources & Equipment

- Marty Workbook (Lesson 2)
- Maze worksheet
- Scratch print out blocks
- Python print out commands
- Counter/marker for maze worksheet

Additional Reading

- Educator's Guide

Learning Plan & Activities

1. Start by doing a recap on Scratch as a programming language before introducing Python and showing some sample commands/short example scripts
 - a. Ask students to comment on what they notice immediately to be different about Python compared to Scratch
 - b. Ask students if they can name any other programming languages and whether it's more similar to Scratch or Python
2. In small groups, hand out the maze worksheet along with the Scratch and Python print out blocks/commands and a counter/marker to keep track of where in the maze students are
 - a. Since some blocks may need to be repeated, students should use the blocks as a guide and write down the instructions in the workbooks
3. Ask students to use the printed Scratch blocks to make an algorithm to get them out of the maze from the starting point
4. Then ask student groups to use the printed Python commands to do the same thing – if it helps, they can keep their Scratch algorithm to compare

- Ask students to reflect on the activity – what did they have to do differently when using the Python commands, which was more difficult, etc.

Additional Challenges

- Use the printed-out control blocks such as repeat/loops and ask students to repeat the task but using this new block and command for Scratch and Python
- Ask students to create their own maze for other student groups to try using the instructions to get around

Curriculum Benchmarks

Curriculum for Excellence – Technologies Benchmark Guide

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 2.2
Digital Literacy	TCH 0-01a	●
	TCH 0-02a	●
Technological Developments in Society & Business	TCH 0-05a	●
Craft, Design, Engineering and Graphics	TCH 0-09a	●
	TCH 1-09a	○
	TCH 0-11a	●
	TCH 1-11a	●
	TCH 2-11a	○
Computing Science	TCH 0-13a	●
	TCH 1-13a	○
	TCH 2-13a	○
	TCH 3-13b	○
	TCH 4-13a	○
	TCH 0-14a	●
	TCH 0-14b	●
	TCH 1-14a	○
	TCH 1-14b	○
	TCH 2-14a	○
	TCH 0-15a	○
	TCH 1-15a	○
	TCH 2-15a	○

National Curriculum – Computing, Design & Technology

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 2.2
Computing	1-a	●
	1-b	●
	1-c	●
	2-a	○
	2-b	●

	2-c	●
	3-a	○
	3-b	●
	4-a	○
	4-b	●
Design & Technology	1.1-a	●
	1.1-b	●
	1.3-b	●
	2.3-b	●
	3.3-c	●

Australian F-10 Curriculum – Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 2.2
Digital Technologies	ACTDIK001	○
	ACTDIK002	●
	ACTDIP004	●
	ACTDIK008	●
	ACTDIP010	●
	ACTDIP011	○
	ACTDIP013	●
	ACTDIP017	○
	ACTDIP019	○
	ACTDIP020	○
	ACTDIP029	○
	ACTDIP040	●
Design & Technologies	ACTDEP006	●
	ACTDEP009	●
	ACTDEP018	●