**A black and white logo

Description automatically generated with low confidenceExploring** Financial Identity through Digital Technologies

Coding for animation

**Animate your money personality character using Scratch.**

This resource supports learners to animate their money personality character. Learners should develop their character first before starting this resource.

About this resource

A picture containing background pattern

Description automatically generatedView the Level 4 and 5 achievement objectives related to this learning experience [**here**](https://sortedinschools.org.nz/api/v1.0/download?files=312)**.**

# A person working on a computer Description automatically generated with medium confidence

A picture containing text, monitor, screen, screenshot

Description automatically generated



# SOLO taxonomy

The learning experiences in this resource are aligned to SOLO Taxonomy to ensure cohesiveness, constructive alignment and cognitive stretch for all learners. This gives you choices throughout the learning process.

|  |  |  |
| --- | --- | --- |
| **Need it/Know it** | A picture containing text  Description automatically generated **Think It/Link it** | A picture containing icon  Description automatically generated **Extend it/Defend it** |
| Make connections to what you aready know. This is the starting point for new learning. | Link your ideas and make connections to build new knowledge and understandings. Learn about the perspective and insights of others. | Extend your learning by applying it to new contexts. Find evidence, validate sources, summarise your thinking and present your findings to clarify. |

# Sorted themes

# Icon Description automatically generatedIcon Description automatically generatedIcon Description automatically generatedIcon Description automatically generated

Goals



A picture containing logo

Description automatically generated

Managing my money

Saving

Debt

A black and white logo

Description automatically generated with low confidence Learning experiences

|  |  |  |
| --- | --- | --- |
| **Need it/Know it** | **Think it/Link it** | **Extend it/Defend it** |
| **Define** *Computational concepts*   * **Sequence** – identify a series of steps for a task * **Loops** – run the same sequence multiple times * **Parallelism** – make things happen at the same time * **Events** – one thing causing another thing to happen * **Conditionals** –make decisions based on conditions * **Operators** – support for mathematical and logical expressions * **Data** – store, retrieve, and update values.   **View** a basic demo of Scratch, either through a live demo or through the [Scratch overview video](https://www.youtube.com/watch?v=ywG6lv9mFLI).  A large range of projects are available online to inspire you. Spend some time viewing sample projects. The [Scratch website](https://scratch.mit.edu/) has many interesting examples.  Begin your Scratch experience by working through the following simple steps, making the Scratch animated sprite, a cat, dance:   * Start by dragging the “move 10 steps” block from the “Motion” blocks palette to the scripting area. Every time you click on the block the cat moves a distance of 10 steps. You can change the number to make the cat move a greater or shorter distance. | **Something surprising**   * **Explore** the [**Scratch**](https://scratch.mit.edu/) interface in an open-ended way. Challenge yourself by taking just 10 minutes to make something surprising happen to a sprite. * **Explain** to a classmate what you are figuring out during the 10 minutes and ask others for help when you need it. * **Find out** whether anyone in the class or group has figured out:   + how to add sound   + how to change the background   + how to access the help screens for particular blocks. * **Explain** your code to others and ask questions about unfamiliar code constructs that other groups have used.   Physical programming through the Scratch interface  Write some Scratch code on paper and get a classmate to physically walk it through to show understanding of a certain part of Scratch. | **Share** two strategies that can be used when you get stuck while designing.  Reflect and evaluate testing and debugging strategies   * What was the problem? * How did you identify the problem? * How did you fix the problem? * Did others have alternative approaches to fixing the problem?   Set up a class “helpdesk” where you can log problems that others might be able to help you with.  **Tip: Pick different learning experiences from each column to build a framework of lessons that differentiates your classroom** |

A black and white logo

Description automatically generated with low confidence Learning experiences *continued*

|  |  |  |
| --- | --- | --- |
| **Need it/Know it** | A picture containing text  Description automatically generated **Think it/Link it** | A picture containing icon  Description automatically generated **Extend it/Defend it** |
| * **Define** From the “Sound” palette, drag the “Play drum” block. Click on the block to hear its drum sound. Drag and snap the “play drum” block below the “move“ block. When you click on this stack of two blocks, the cat will move and then play the drum sound. * Copy this stack of blocks (either using the Duplicate toolbar item or by right clicking the stack and selecting “duplicate”) and snap the copy to the already placed blocks. * Change the second “move” block to -10 steps, so the cat moves backward. Every time the stack of four blocks is clicked, the cat does a little dance forward and back. * Go to the “Control” blocks palette and grab the “repeat” block. Wrap the “repeat” block around the other blocks in the scripting area. Now when you click on the stack, the cat dances forward and back 10 times. * Finally, drag the “when Sprite clicked” block and snap it to the top of the stack. Click on the cat (instead of the blocks stack) to make the cat dance.   Based on: [CREATIVE COMPUTING a design-based introduction to computational thinking](http://scratched.gse.harvard.edu/sites/default/files/curriculumguide-v20110923.pdf) | Apply these instructions to highlight *parallelism* (things happening at the same time) and *events* (one thing causing another thing to happen):   * Have one person do one thing (like walk across the room). * Have that person “reset”. * Have that person do two things simultaneously (like walk across the room and talk). * Add the second person, by having the second person simultaneously (but independently) do a task, like talking. * Have the second person do a dependent task, like responding to the first person instead of talking over them.   Apply your Pass-it-on Story - Work in pairs.  The pass-it-on story is a Scratch project that is started by a pair of people and then passed on to two other pairs to extend and reimagine it. You can start your story however you want to, focusing on characters, scene, or plot. Each pair has 10 minutes to work on their contribution to the collaborative project before the groups rotate.  Based on: [CREATIVE COMPUTING a design-based introduction to computational thinking](http://scratched.gse.harvard.edu/sites/default/files/curriculumguide-v20110923.pdf) |  |

A picture containing pool ball, vector graphics, table

Description automatically generated