

Arnika Macphail @ArnikaMac

Jo Nicol @jonicolnz

@CycloneNZ



# DIGITAL TECHNOLOGIES CURRICULUM 2018

What might it look like  
and how you could  
implement it into your  
classroom?



The following information is based on the information that has been released so far regarding the draft digital technologies curriculum. Some minor aspects of what we present may be different from what is finally released next year.

# Outline

- ▶ What is the difference between digital technologies, digital fluency and eLearning?
- ▶ What form will the new digital technologies curriculum take?
- ▶ What level will students be expected to reach by the end of Year 10?
- ▶ Computational thinking
- ▶ Designing and developing digital outcomes
- ▶ Resources for further learning

# Important Distinction

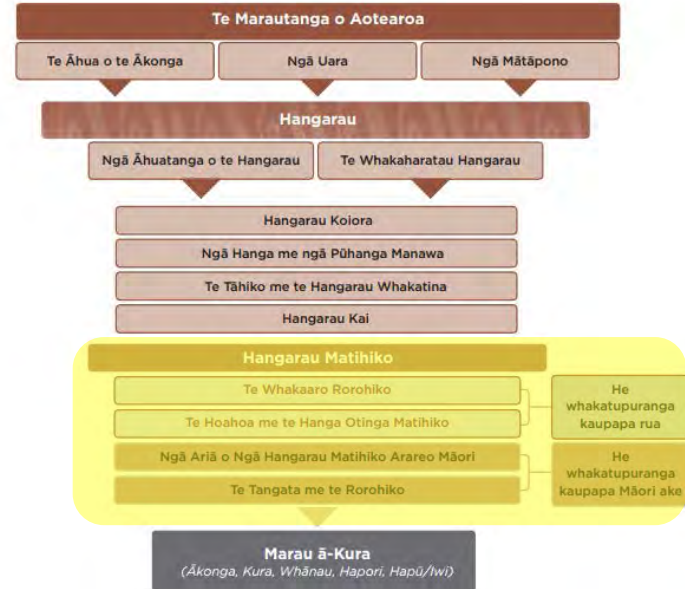
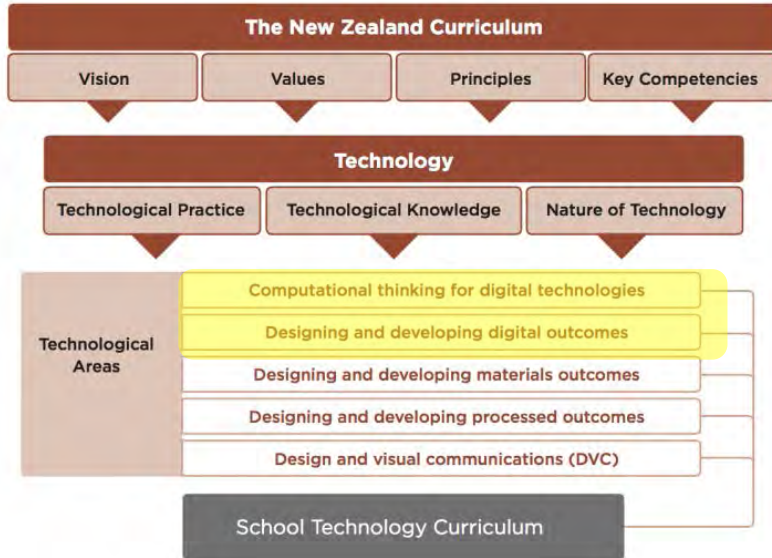
eLearning	Digital Fluency	Digital Technologies Curriculum
Learning through the using technology	Learning how to use technology, when to use technology and why they have chosen to use particular technology	Learning about technology and how to be a creator in the digital world, not just learning to use systems

# What do we know about what it will look like?

- ▶ From 2018, Digital Technologies | Hangarau Matihiko (DT | HM) will be integrated into the Technology
- ▶ Rewriting “essence statement” for the digital technologies curriculum area.
- ▶ The three strands of the traditional technologies curriculum will remain the same (technological practice, technological knowledge and nature of technology)
- ▶ However, where there were previously 3 technologies areas (processed outcomes, material outcomes and DVC outcomes) there will now be 5 areas ,two of which are specifically focused on digital technologies (computational thinking and designing and developing digital outcomes)
- ▶ A Learning Progression Framework (similar to what has just been introduced in reading, writing and maths) will be produced to signpost a learner’s development in key skills, knowledge and attitudes, as their understanding grows throughout their education

# The New Curriculum Strand

## → Digital Technologies | Hangarau Matihiko 2018



By the end  
of Year 10  
students will  
be able to..

## Computational Thinking

### End of Year 10

Students understand that there can be multiple algorithms for the same problem, some are better than others, and by recognising patterns between problems they can generalise known algorithms so they can re-apply these.

Students can independently decompose problems into an algorithm that is articulated in such a way that a computing device can understand.

They can implement the algorithm by creating a program which uses inputs, outputs, sequencing, loops and selection using comparative operators and logical operators.

They can take these skills and work in a collaborative environment to solve computing problems.

Students can explain/document their programs and use an organised approach for testing and debugging.

Students understand how computers store different types of data using binary digits and can use variables of different data types within their programs.

## Designing and Developing Digital Outcomes

### End of Year 10

By the end of year 10 students will be able to make decisions (largely independently) about the best tools/techniques to solve the problem.

They can work through an iterative process to design, develop, create, store, test and evaluate digital content that meets its purpose.

They can take these skills to select appropriate software and file types for particular purposes based on key features, and justify the selection.

Use selected software to develop and combine digital content to create an outcome.

Students can understand the role of operating systems in managing personal computer hardware, security, and application software.

Students can explain the conventions of file management procedures and use of storage devices.



# Computational Thinking

# Computational Thinking

<http://core-ed.org/research-and-innovation/ten-trends/2016/computational-thinking/>

"Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."

Cuny, Snyder, Wing, 2010

<https://cs-unplugged.appspot.com/en-gb/computational-thinking/>



# BINARY NUMBERS

---

Link to activity we did:

<https://cs-unplugged.appspot.com/en-gb/topics/binary-numbers/unit-plan/how-binary-digits-work-junior/>

# CS Unplugged

<https://cs-unplugged.appspot.com/en-gb/>



Computer Science without a computer

View a sneak peek of CS Unplugged 2.0!

# Kidbots

## New Words!

### Algorithm

Say it with me: Al-go-ri-thm

*A list of steps that you can follow to finish a task*

---

### Program

Say it with me: Pro-gram

*An algorithm that has been coded  
into something that can be run by a machine*

# Kidbot Roles

*Developer:* The person writing the instructions.

*Tester:* The person checking the instructions work.



Code.org

# KidBots



Two hands on  
side of shoulders



Two hands on  
top of shoulders



One hand on  
back



One hand on right  
shoulder



One hand on left shoulder



Code.org

Kidbot says "Syntax Error" if  
they don't understand

## Fast finishers: fitness unplugged

In your groups add some exercise symbols into your programme (e.g. star jumps/hopping). Programme your kidbot to complete a short exercise routine.



# BeeBots

- Spelling words
- Learning Te Reo
- Basic Facts
- Story telling
- Story Coding



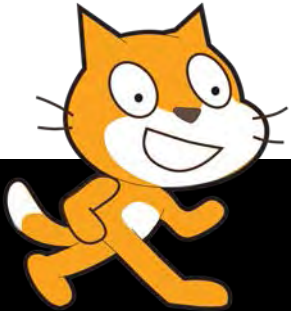


## Plugging In

Free BeeBot and BlueBot Apps on iPads available.



# Getting Started with ScratchJr



2017

Credit :Mandy Dempsey

# Getting Started With ScratchJr



Website:

<http://www.scratchjr.org/>

App Store:

<https://itunes.apple.com/nz/app/scratchjr/id895485086?mt=8>

Beyond Jr:

<https://scratch.mit.edu/>



# Getting Started With ScratchJr



## → Introduction to Motion Blocks (Unplugged)

- ◆ Robot Toys
- ◆ KidBots
- ◆ Simon Says



## → Introducing Scratch the Cat (Plugging it in)

- ◆ Make Scratch Move (Motion Blocks)
  - Using backgrounds
  - Make a shape
- ◆ Scratch Dance Party! (Exploring Size, Speech and Sound)

# Getting Started With ScratchJr



## → Scratch Collages

- ◆ Make a Fish Tank! (Multiple Characters, Loops, Creating Own characters)
- ◆ Spooky Forest!



# Crossy Road?



# Project-based learning

## Your challenge!



Ladbrooks School  
Tours

# Your task

Design a Scratch Animation that highlights one of your  
**School Values**  
(eg *Respect, Responsibility, Creative Thinkers etc*)





# Role play

Role play what you want to program.

This could be using:

- A whiteboard
- Recording from a device as if they are reporting
- Using toys



# Setting up your sprites and backdrop

First we will set up our character. We are only going to program one character to start with, but you can add more later.

1. Select a sprite with no face
2. Click on the camera
3. Double click in the sprite and take your photo
4. Now take a picture for your backdrop



# Your program needs to have

**An output** - this could be written words or a recording. Write 1 sentence per block.

**A loop** - that means two or more steps need to be repeated a number of times.

**Sequential programming** - especially using the red end block so you don't have to click to make the background change.

# Sharing ideas

Take a movie using a second tablet of the project

Email the project and airplay/share to a laptop to take a screencast and import into a movie making tool (this could be an older buddy class doing this, with the juniors being their “product owners”)

Run a wet day video arcade where students get to learn through your programs.

Have a rotation of Drop Everything and Read using technology (programs created for others to use)

# Scratch - Step by Step

<https://scratch.mit.edu/>

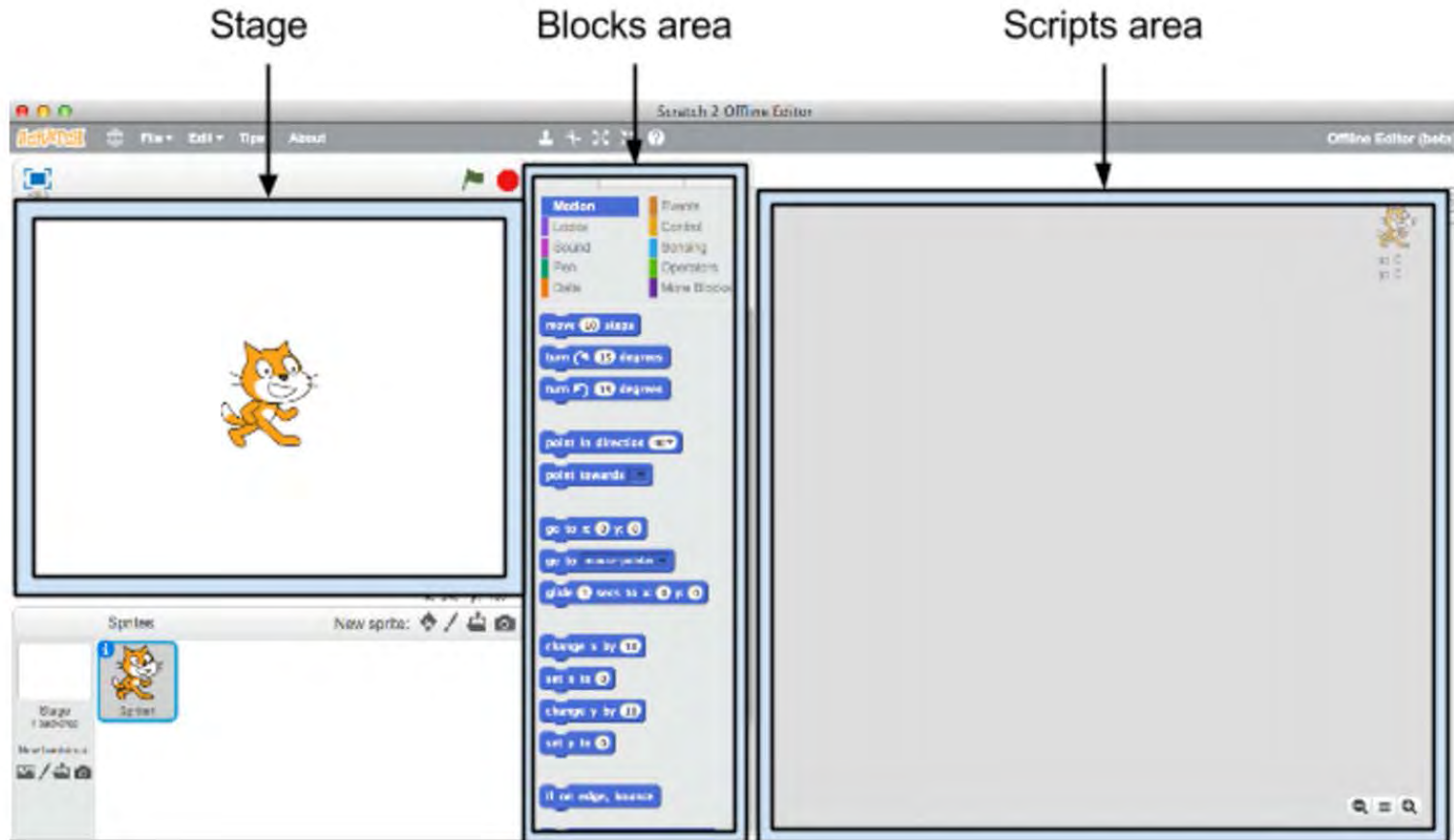
From: CS4PS

# What we will aim to cover

1. Inputs and Outputs
2. Conditionals
3. Loops
4. Variables

Through all of this you will also be practicing sequencing instructions, testing, and debugging

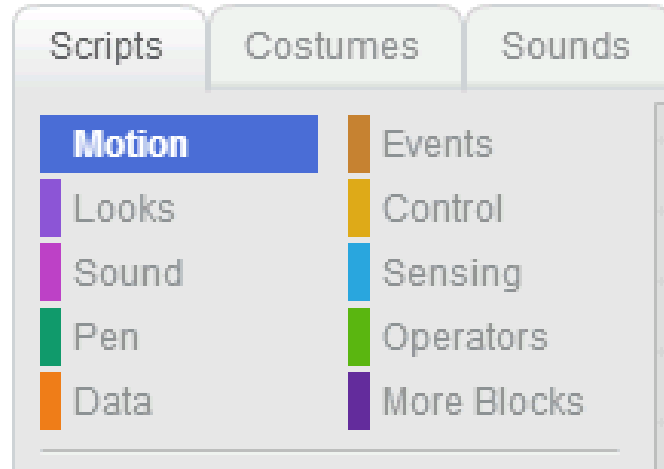
# The Scratch workspace...





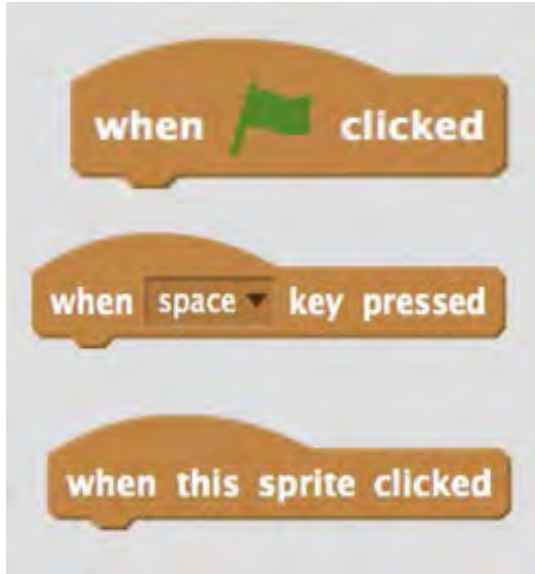
# Scratch Blocks

- Blocks are grouped together into colour coded categories
- A lot of blocks can seem confusing at first, so don't worry! We will cover some of them one-by-one
- Take 5 mins to glance through the categories. At the end of this time try and find how to make a sprite say "Hello!"





# Making things happen



- When we write a program we need to tell it when to run, otherwise the computer will never know when you want the program to happen!
- Look at the blocks in the **Events** category. The main one we will be using is the “*when <green flag> clicked*” block

# Hello World

→ Drag the “*when <green flag> clicked*” block, and then a “*say*” block into the Scripts area and connect them together to create your first program!



# Tasks - Inputs and outputs

- Modify your program so the Sprite says Hello to you in a different language
- Change your program so the sprite asks your name, and then says it.
  - ◆ Hint: look in the “sensing” tab
- Change your program so the sprite asks what your name is, and then says “Hello ----”, where ---- is your name. You’ll need this block to do it:





# Variables

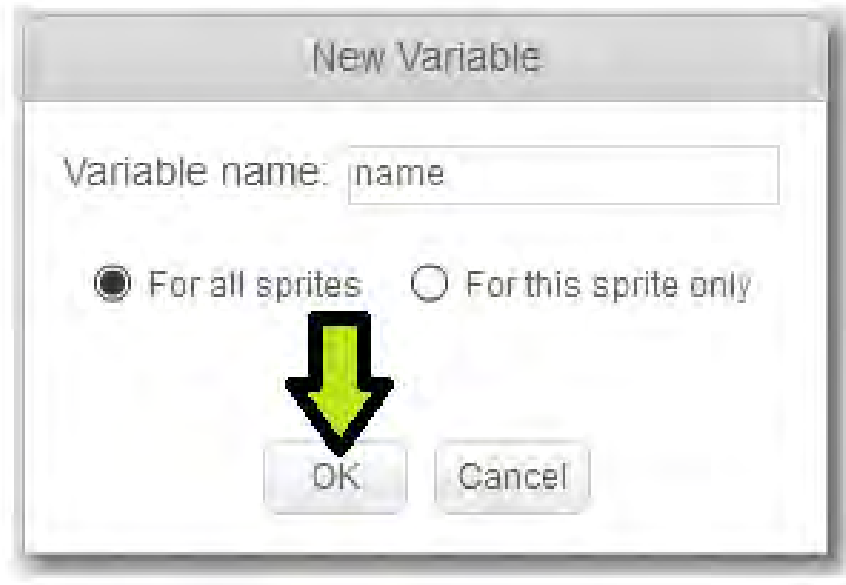
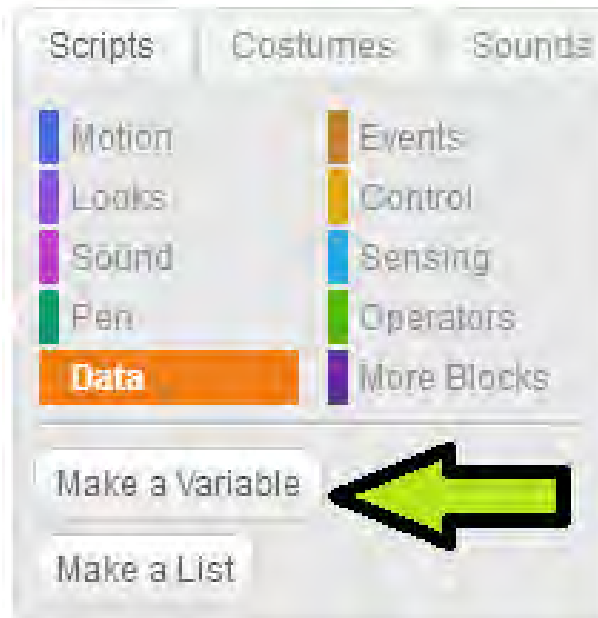


We use variables for storing...

1. Words and sentences
  - a. We call these “strings”
2. Numbers
  - a. E.g. The score in a game
  - b. The results of calculations
3. Files



# Creating Variables



# Inputs and outputs

There are more ways to give a computer inputs than words:

→ Typing in numbers, using the arrow keys, clicking with the mouse...

And there are more ways to get outputs that having the Sprite say words!

→ The sprite moves places, changes colour, makes sounds...

**Save your program and  
start a new one**

# Tasks - Inputs and outputs

- ◆ Hint: You'll need blocks from the Events tab and the Looks tab for these tasks
- Write a program that makes the Sprite change colour when the space bar is clicked
- Add more to your program, so that when you click on the screen the Sprite moves to where you clicked
- Add more so that when you press the up arrow key the sprite gets a bit bigger, and when you click the down key it gets smaller



**Save your program and  
start a new one**

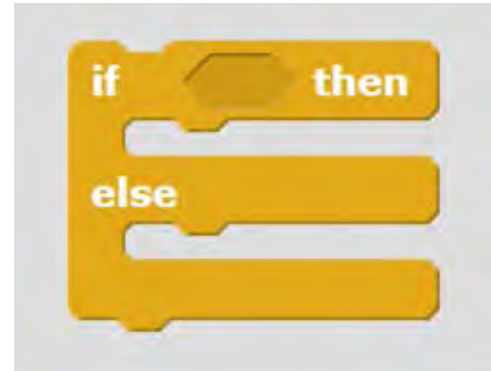
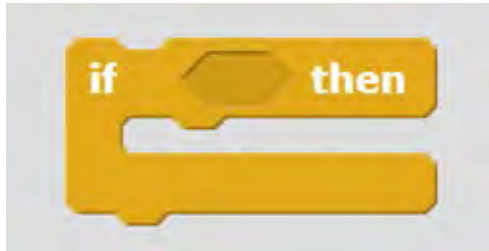


# Conditionals

We use conditionals to test if things are True or False. They are in the **Operators** tab. These are the ones we will use:



To use conditionals we need these blocks from the **Control** tab

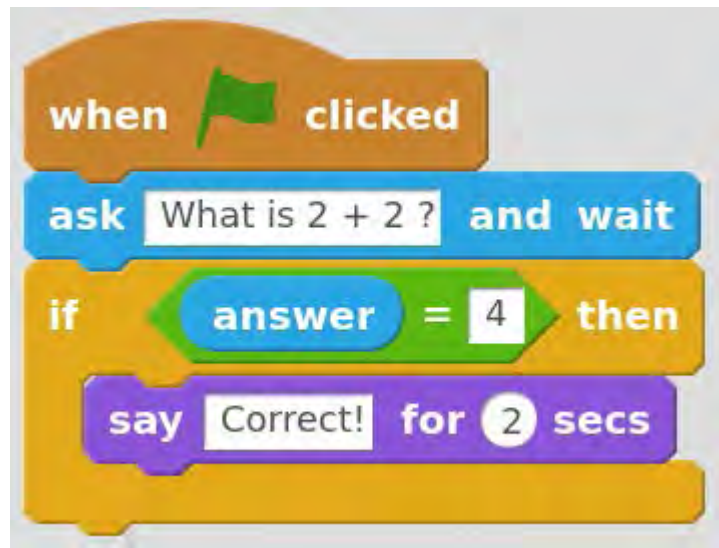


# Using Conditionals

Make this program and test it  
(always test your code!)

Try changing it so it asks a  
different maths question.

Does it tell you you're correct when you type in  
the correct answer?

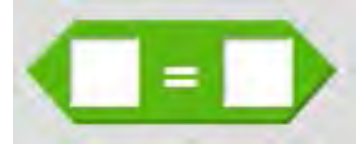
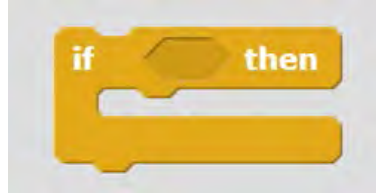


**Save your program and  
start a new one**

# Tasks - Conditionals

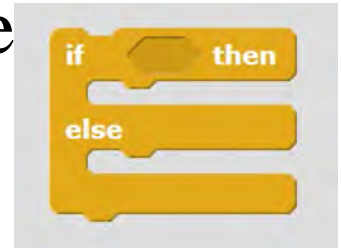
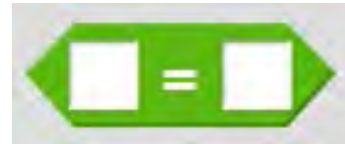
→ Make a program that asks the user “Do you like pancakes?”. If the user says “Yes” then have the sprite say “Me too!”

◆ Let’s run some tests...



→ Modify your program so that if the user doesn’t say “Yes” the Sprite says “Ok, let’s make something else!”

◆ Let’s run some tests...



**Save your program and  
start a new one**

# Tasks - Conditionals

- Write a program that asks an addition question and tells the user if they answered it right or wrong, and then asks a subtraction question and tells the user if they are right or wrong.
- Try modifying this program so that it only asks the second question if the user got the first one correct.

# Tasks - Conditionals

- Write a program that asks the user how old they are, and then tells them if they are old enough to vote.
- Change the program so if the user is not old enough, the sprite will tell them how many years it will be until they are old enough

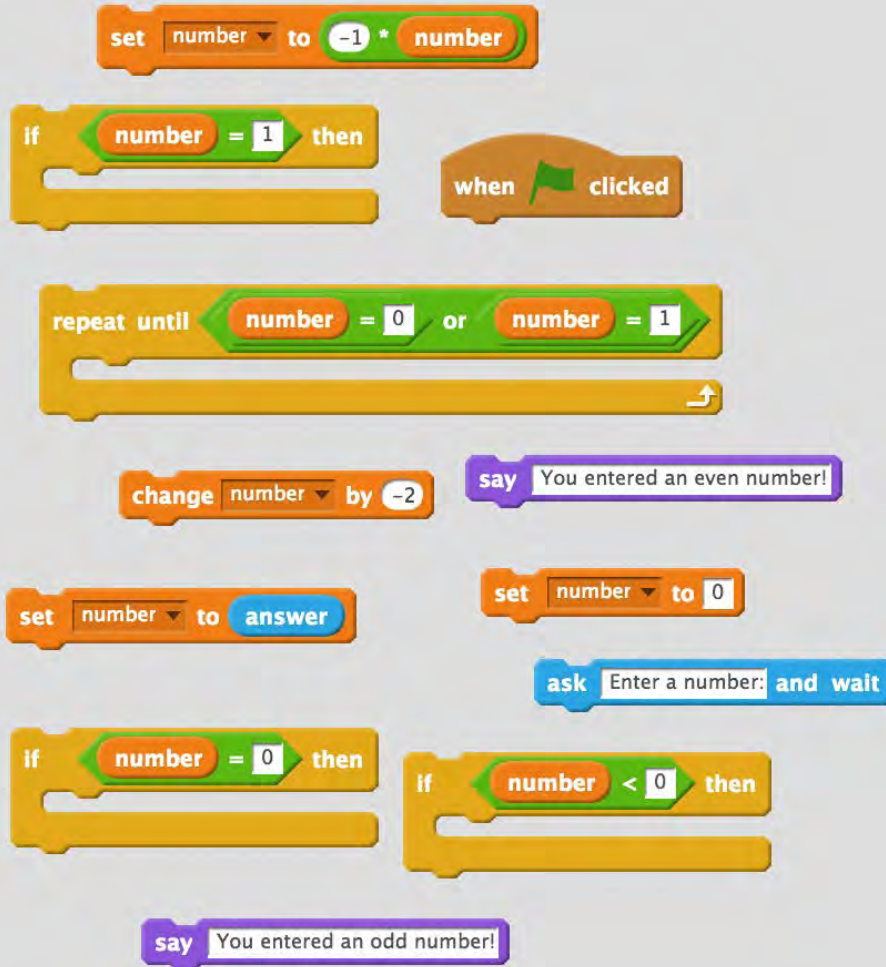


# Task - Odds and evens

→ Using the given blocks, make a program that tests if a number is odd or even...

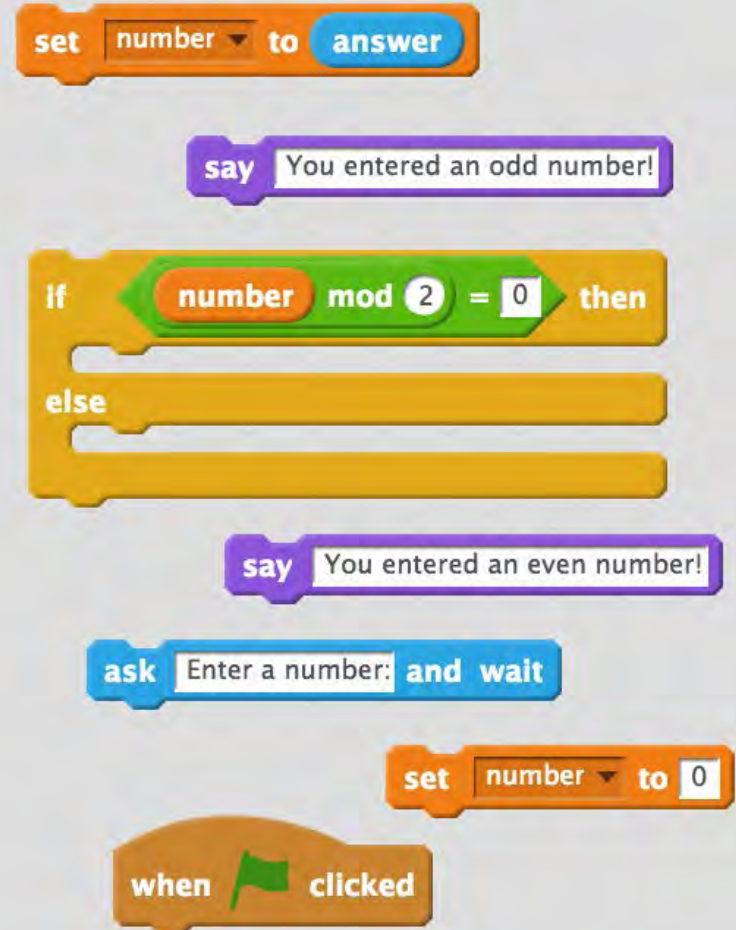
# Version 1

## Using subtraction



# Version 2

Using the mod operator



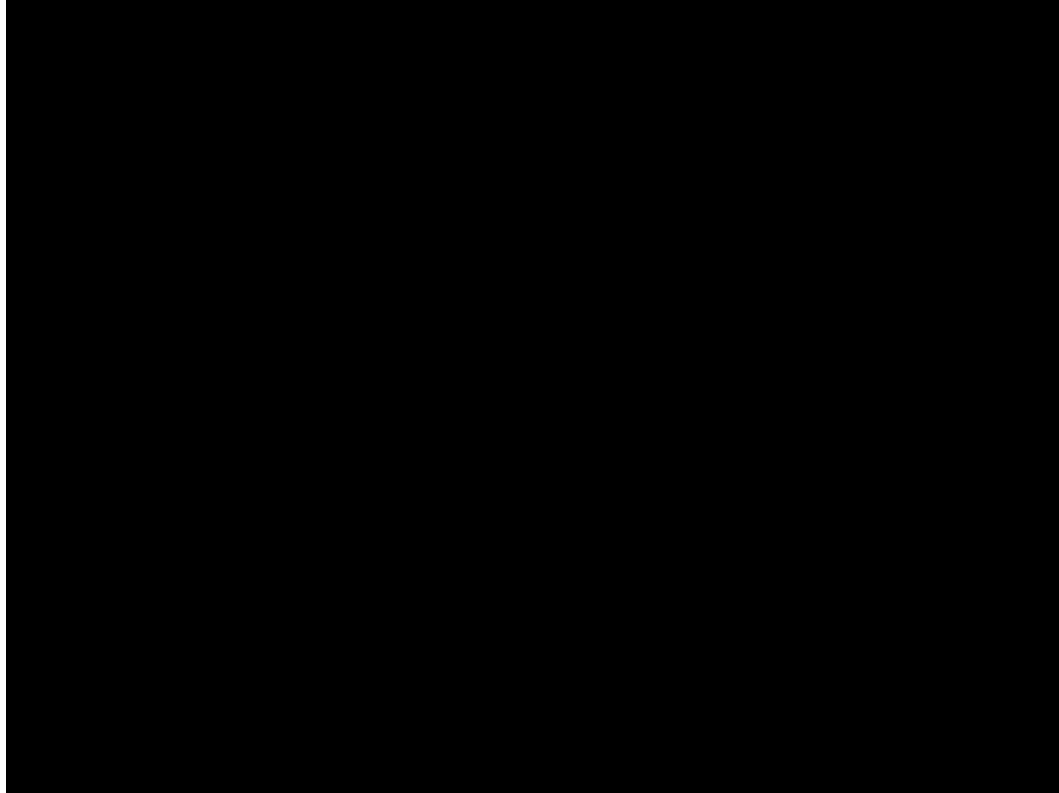
The background of the slide on the left features a stylized, semi-transparent image of an Angry Birds game level. It shows a green pig character, a red bird character, and various wooden blocks and structures. The text 'Online programmes to develop computational thinking' is overlaid on this image.

# Online programmes to develop computational thinking

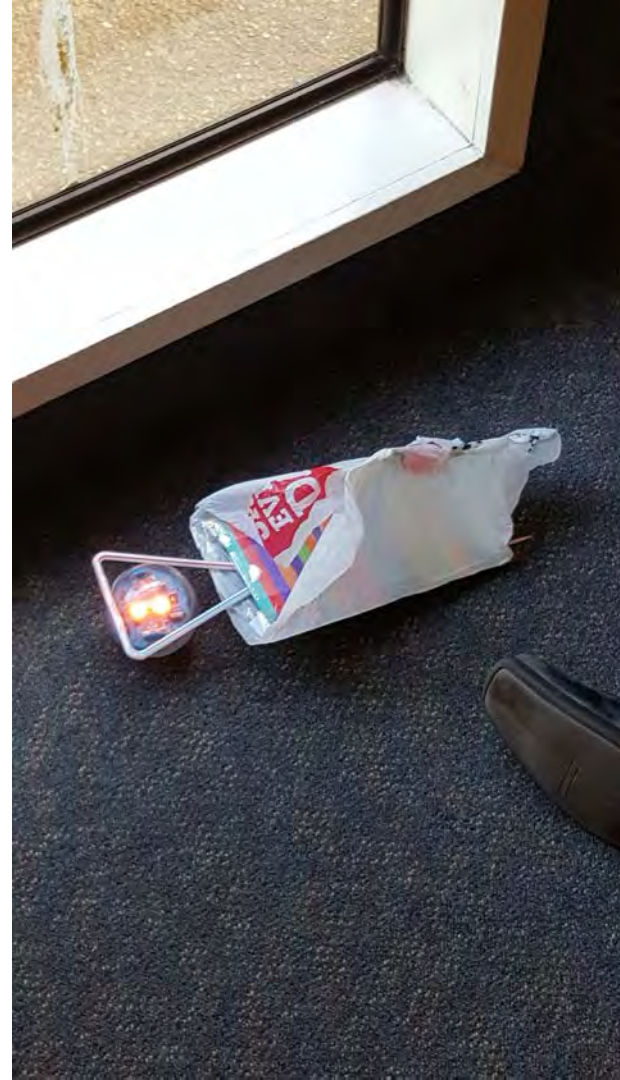
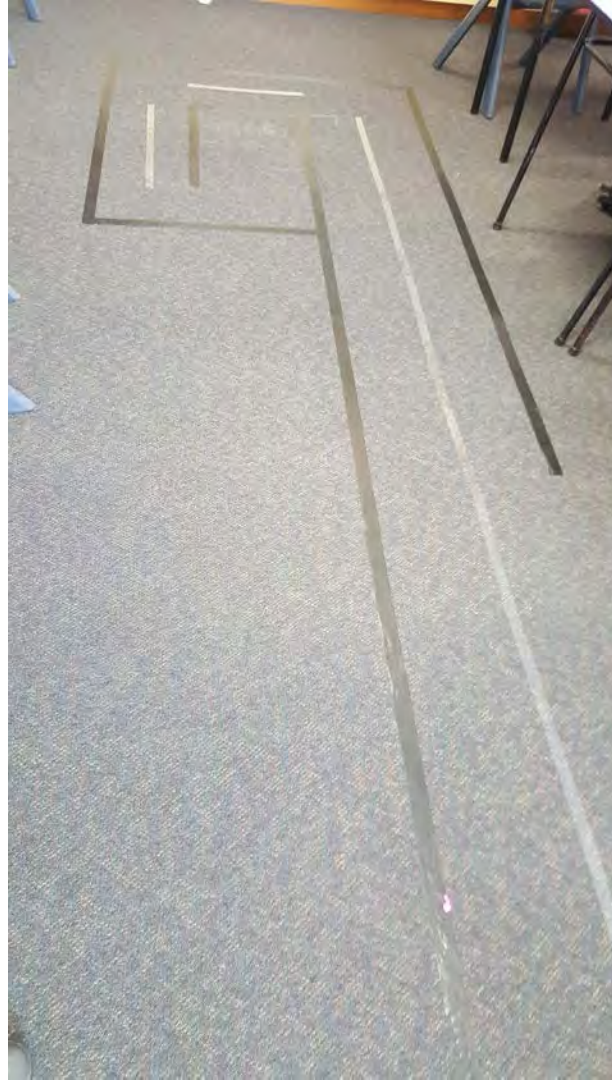
- ▶ <https://code.org/learn> - "Write your first computer programme". We recommend that you watch the introduction.
- ▶ <https://scratch.mit.edu/> - Watch video in top left corner, click into project to see how it is done, then give it a go.
- ▶ <https://www.touchdevelop.com/ccga>

# SPHERO

<https://www.sphero.com/education>









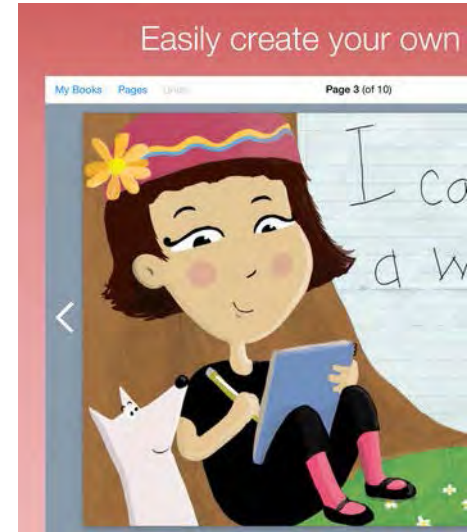


# Designing and Delivering Digital Outcomes

- Students understand that digital applications and systems are **created for humans by humans**. They develop increasingly sophisticated understandings and skills related to designing and **producing quality, fit-for-purpose, digital outcomes**.
- They develop their understanding of the digital Information technologies that people need in **order to locate, analyse, evaluate, and present digital information efficiently, effectively, and ethically**.
- Students also become more aware of how **to build, install, maintain, and support computers, networks, and systems** so that they are secure and efficient.
- Students develop knowledge and skills in using **different creative digital technologies to create digital content for the web, interactive digital platforms, and print**. They construct digital media outcomes that integrate media types and incorporate original content.



# Designing and developing digital outcomes



# Progress Outcome 1

Can participate in a teacher-led development to create, manipulate, store, retrieve and share content. This will include:

- Identifying digital devices and their purpose and knowing that humans made them.
- Identify the inputs and outputs of a system.
- An awareness of some applications and their purpose (face-time, draw studio).
- An understanding that computers store content and we can retrieve it later.

# Want to keep learning?

- Computer Science Unplugged: Computer Science without a computer, <http://csunplugged.org/>
- Digital Technology in the national curriculum, <http://elearning.tki.org.nz/Teaching/Curriculum-areas/Digital-Technologies-in-the-curriculum>