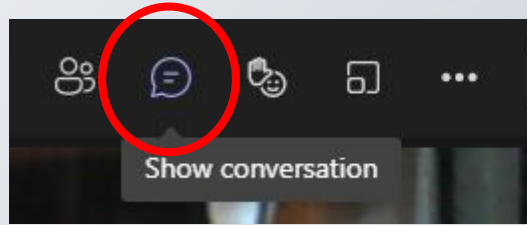


# While we wait to get started ...

Open the chat



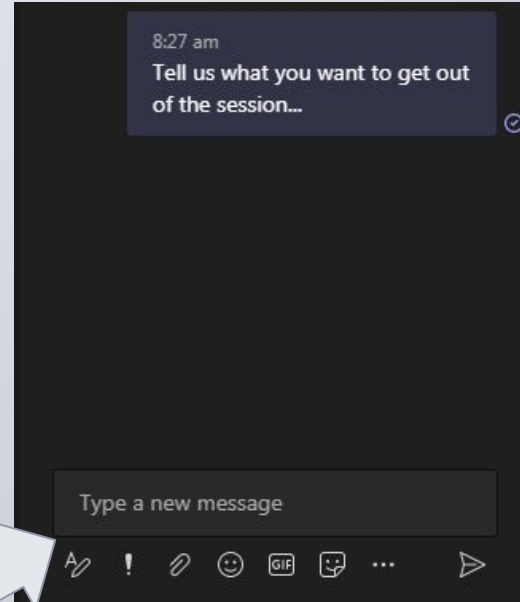
Your mic is on mute  
... and camera disabled

Tell us what you want to  
get out of the session.

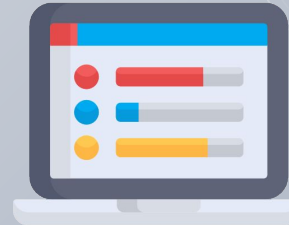
## NOTE:

your name will appear  
with your comment.

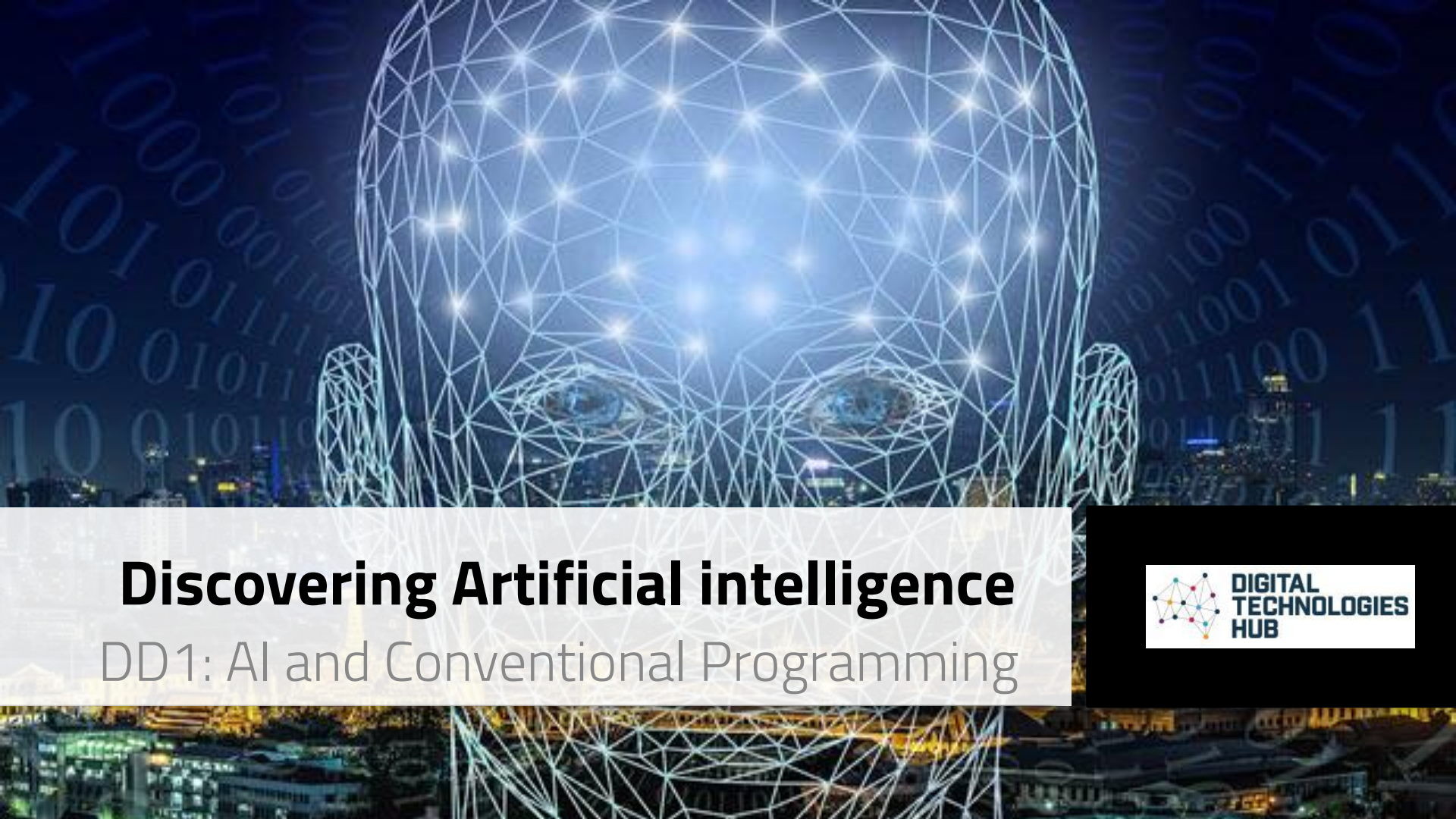
The chat won't be part  
of the recorded version.



Take our poll ... you will  
find it in the chat...



DIGITAL  
TECHNOLOGIES  
HUB



# Discovering Artificial intelligence

DD1: AI and Conventional Programming



# By the end of this session...

You should be able to:

Create a program that includes user input.

Build your teaching strategies to help your students program.

Assess student's programs and identify inputs, branching and repetition.

Undertake a risk assessment to ensure AI tools are safe to use



# Achievement standards: starting point

## Achievement Standard

By the end of Year 2, students identify digital systems (hardware and software) and their purposes. They use digital systems patterns in data in different ways.

Students design solutions to simple problems, defining the problem in terms of data and functional requirements and design solutions by developing algorithms to address the problems. They incorporate decision-making, repetition and user interface design into their designs and implement their digital solutions, including a visual program. They explain how information systems and their solutions meet needs and consider sustainability. Students manage the creation and communication of ideas and information in collaborative digital projects using validated data and agreed protocols.

## Achievement Standard

By the end of Year 6, students explain the fundamentals of digital system components (hardware, software and networks) and how digital systems are connected to form networks. They explain how digital systems use whole numbers as a basis for representing a variety of data types.

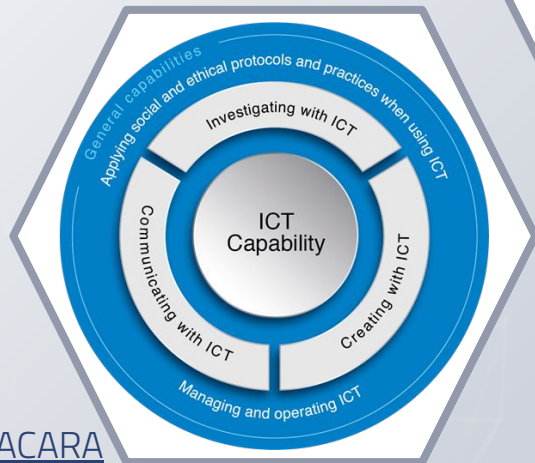
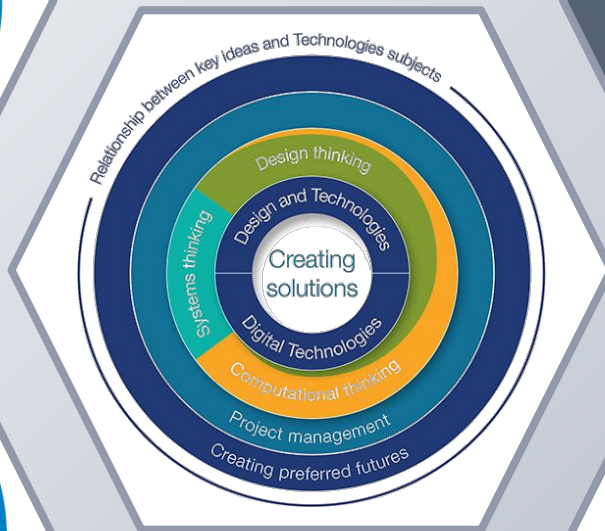
Students define problems in terms of data and functional requirements and design solutions by developing algorithms to address the problems. They incorporate decision-making, repetition and user interface design into their designs and implement their digital solutions, including a visual program.

They explain how information systems and their solutions meet needs and consider sustainability. Students manage the creation and communication of ideas and information in collaborative digital projects using validated data and agreed protocols.

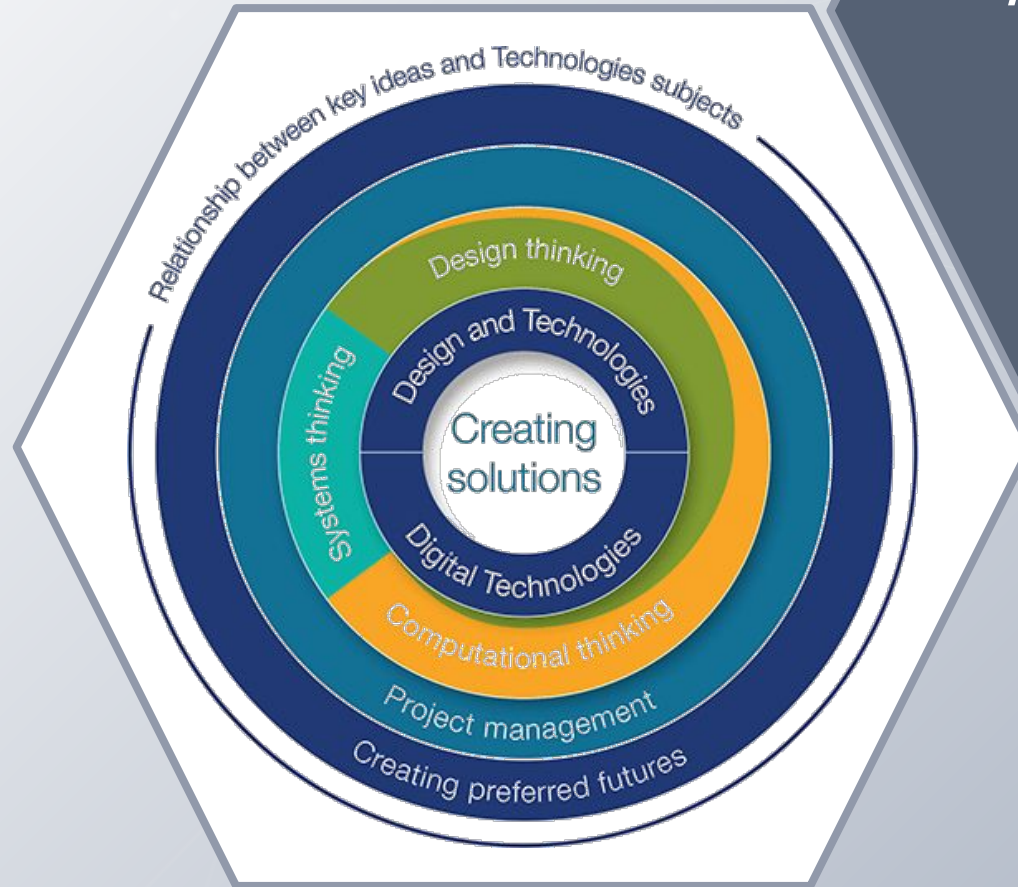
Students describe how a range of digital systems (hardware and software) and their peripheral devices are connected to form networks. They explain how the same digital systems can be used in different ways.

Students design and implement digital solutions that involve decision-making and communication. They explain how the solutions meet their purposes. They use different data when creating digital solutions. They safely use and manage digital systems and agreed protocols. They explain how information systems and their solutions meet needs and consider sustainability. Students manage the creation and communication of ideas and information in collaborative digital projects using validated data and agreed protocols.

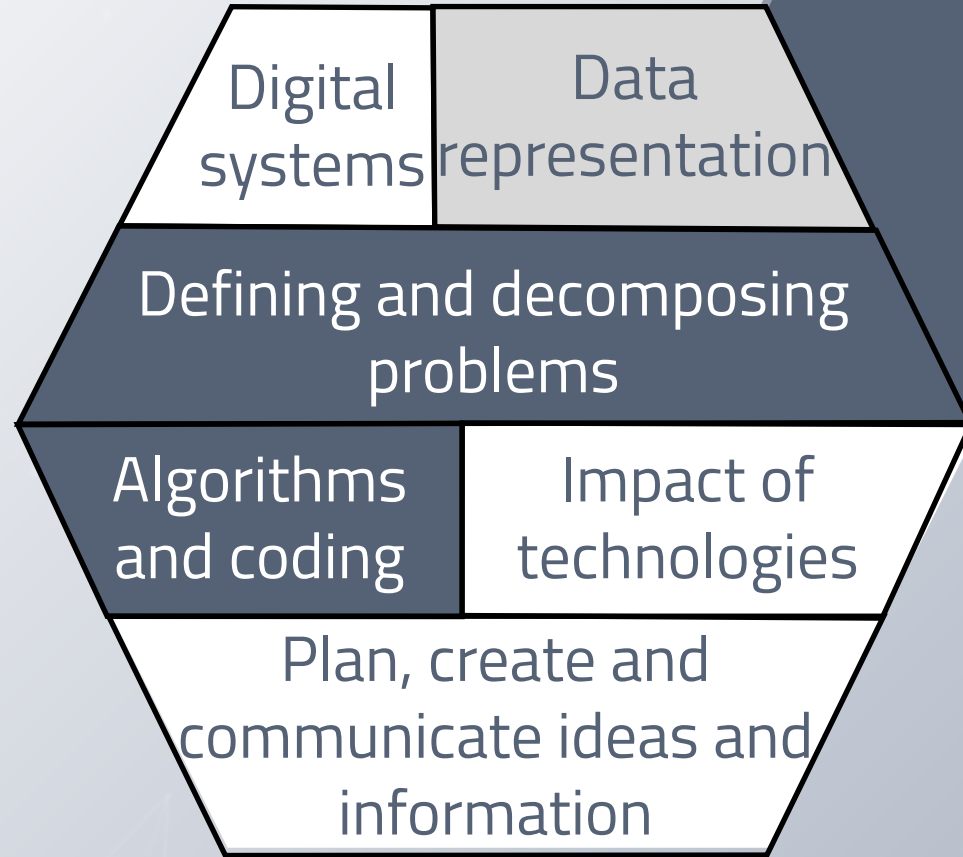
# AI topics



# AI topics



Foci of this  
deep dive





# Systems Thinking



## Design Thinking

## Computational Thinking



**DIGITAL  
TECHNOLOGIES  
HUB**



How is AI different from  
conventional programming?

# Ai vs conventional programming

What to do:

Pre ...?

During ...?

Post ...?

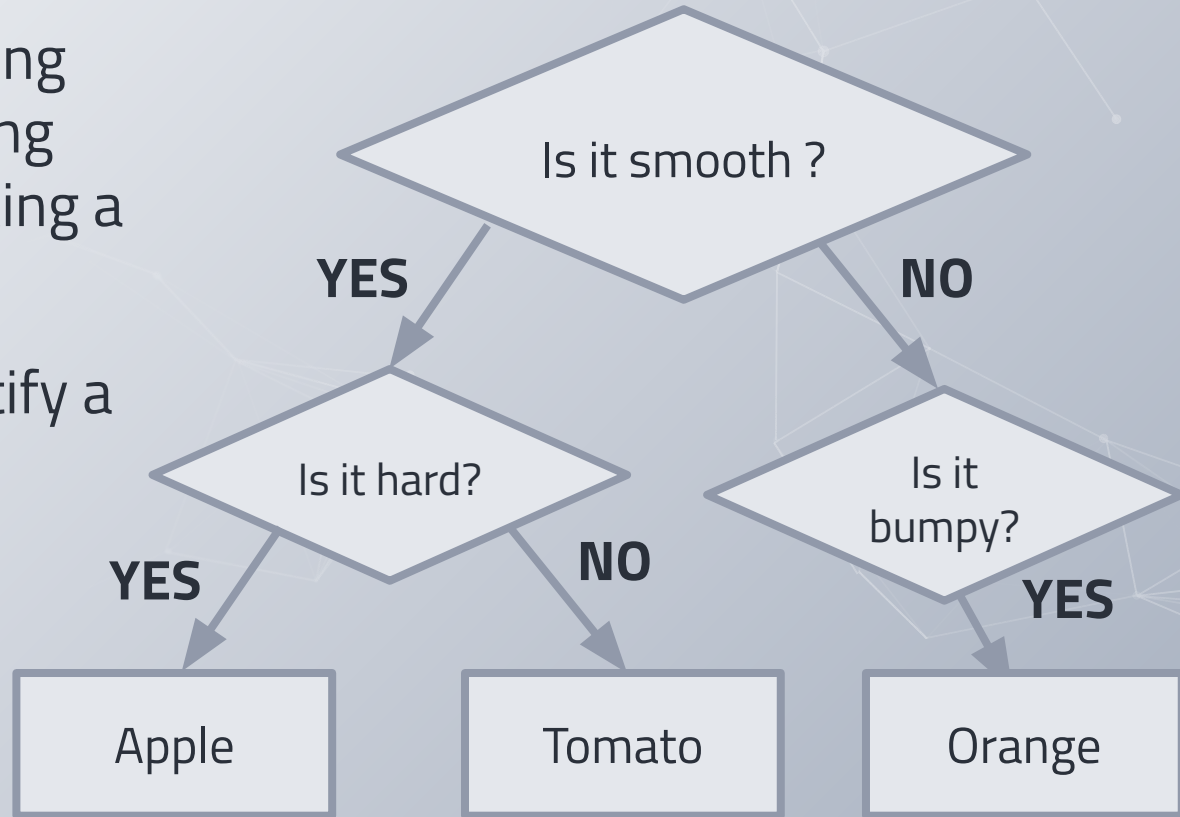


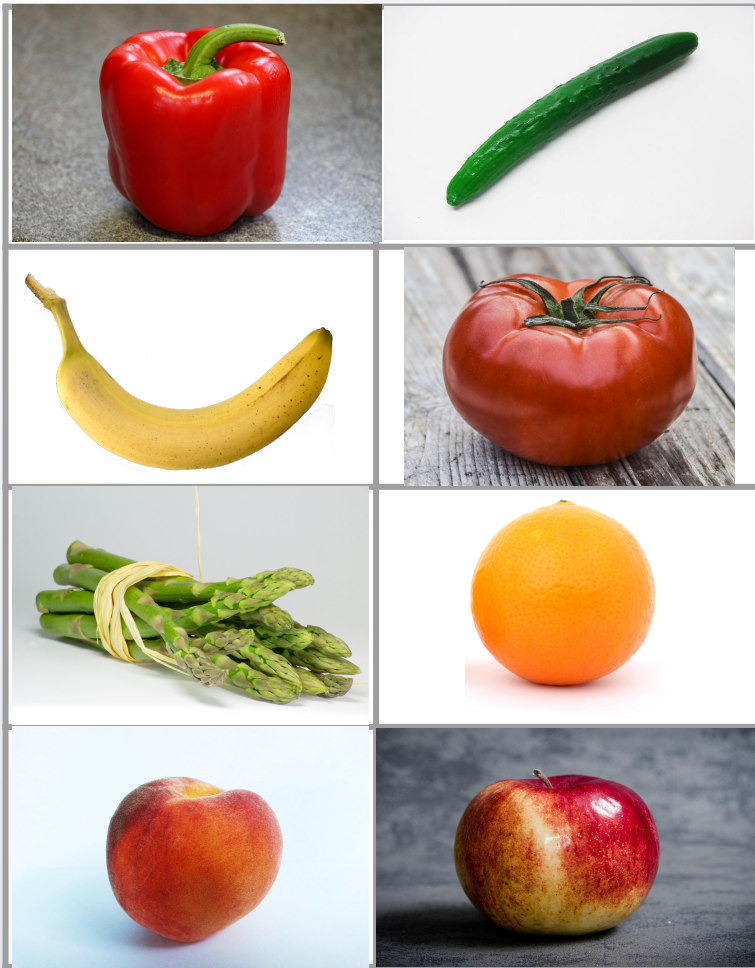
Artificial Intelligence Explainers: [Video 1: Introduction to AI & machine learning](#)

# Conventional programming

A traditional programming approach works by asking YES/NO questions, making a decision tree.

The task here is to identify a fruit or vegetable.





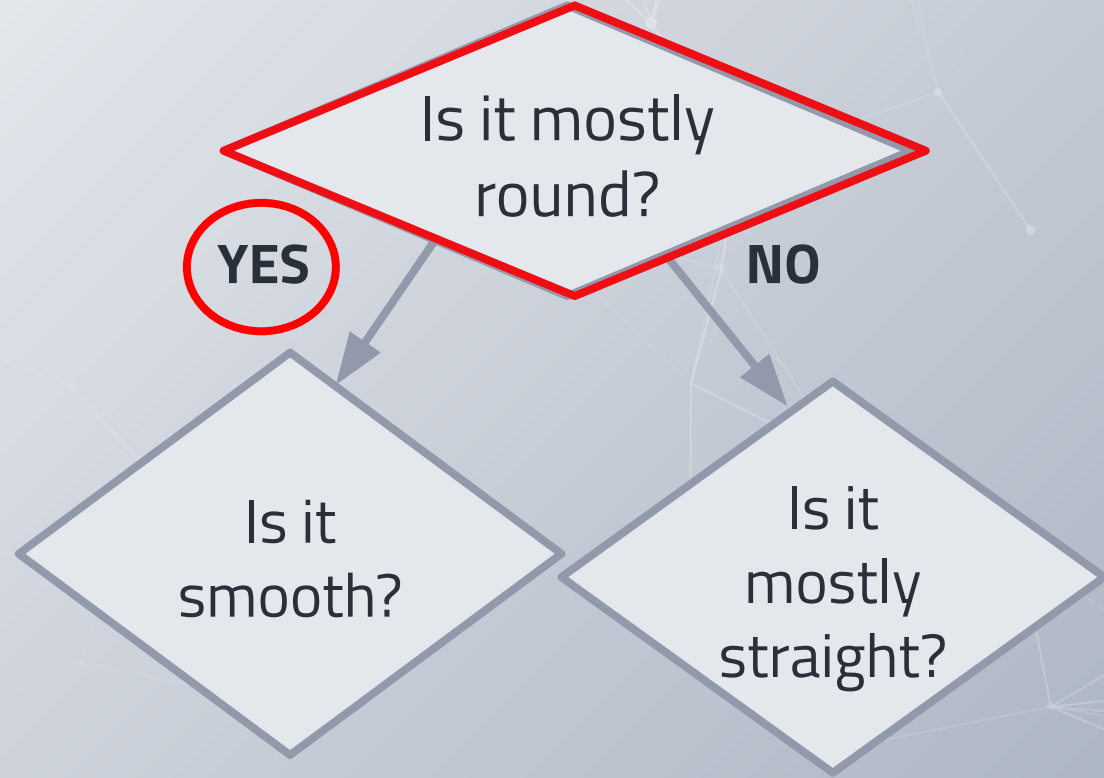
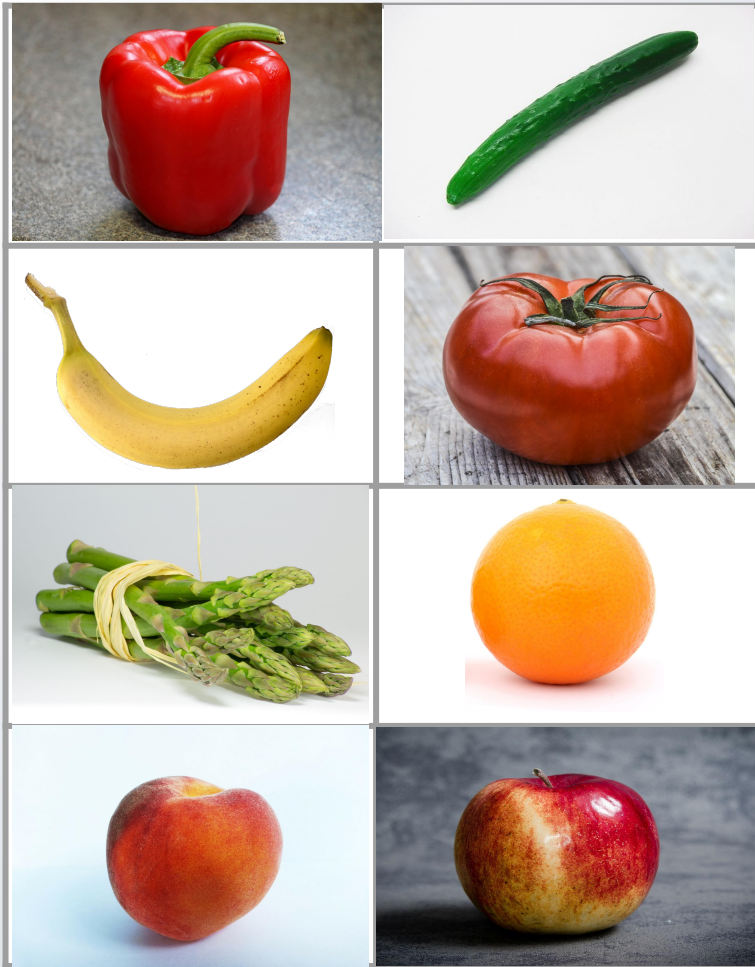
The task here **for the computer** is to identify a fruit or vegetable.

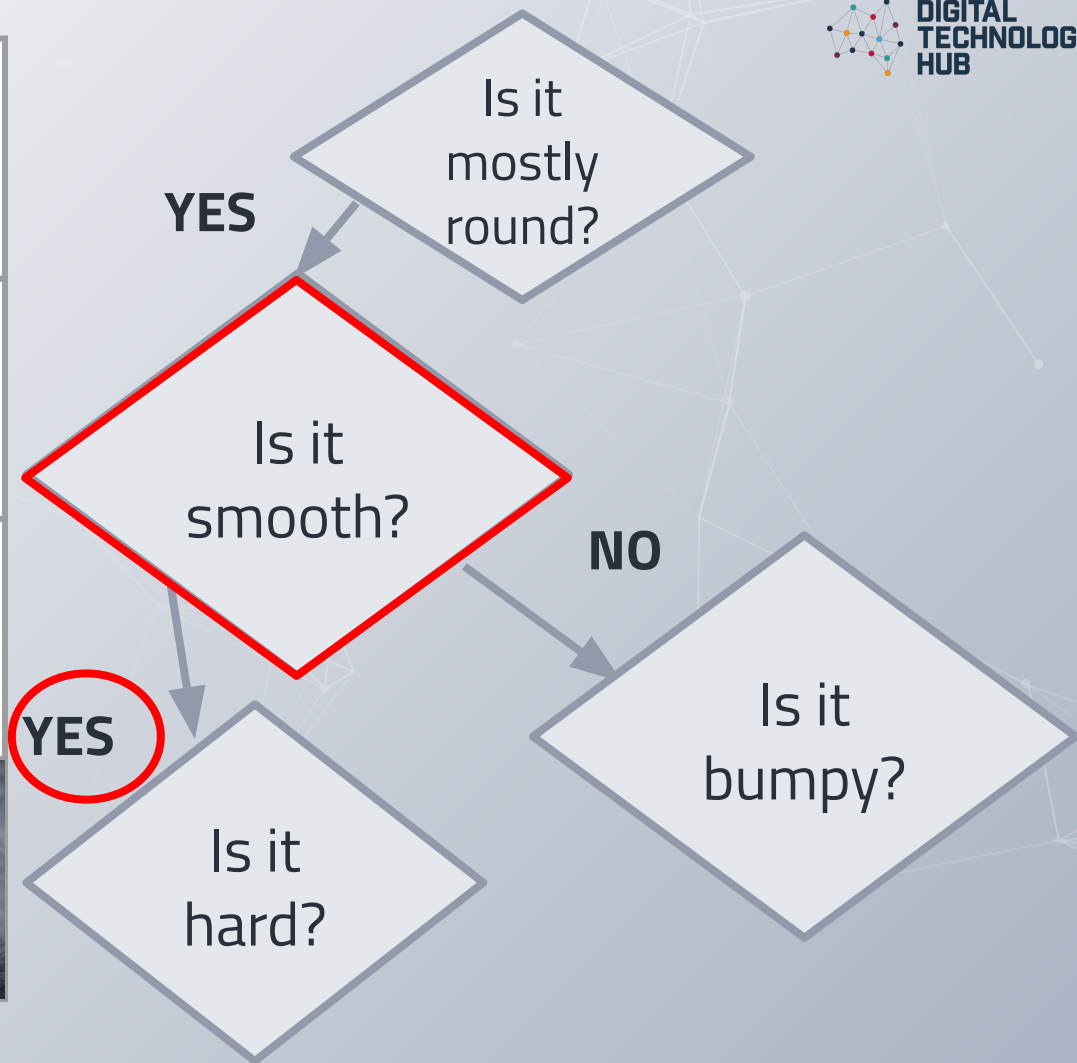
One way to program the computer is to eliminate others by their attributes.

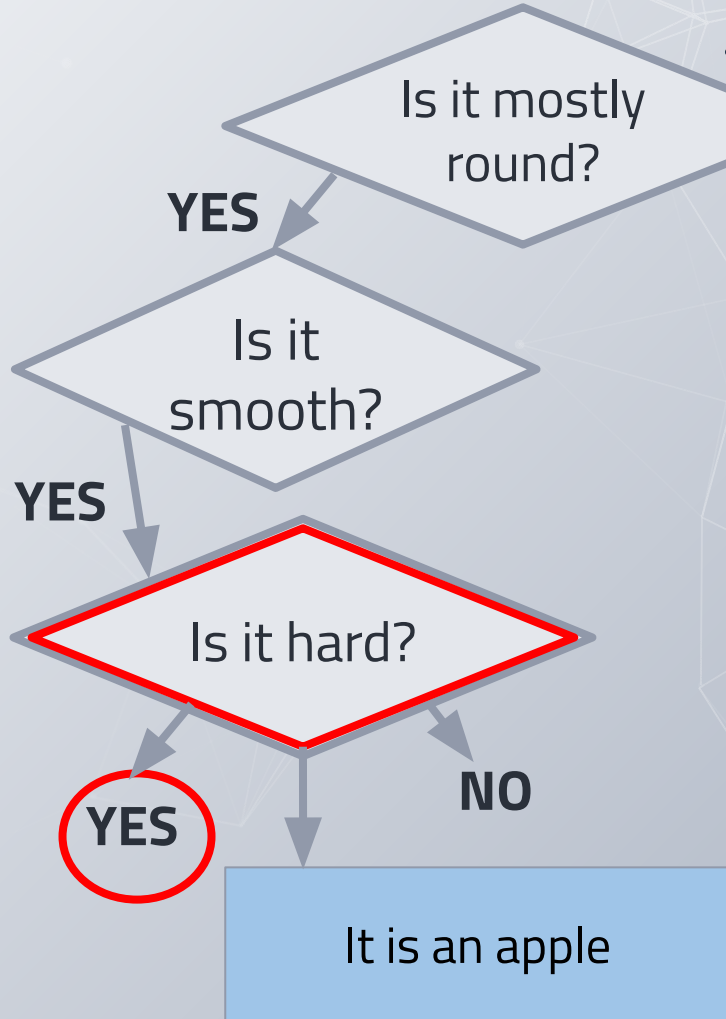
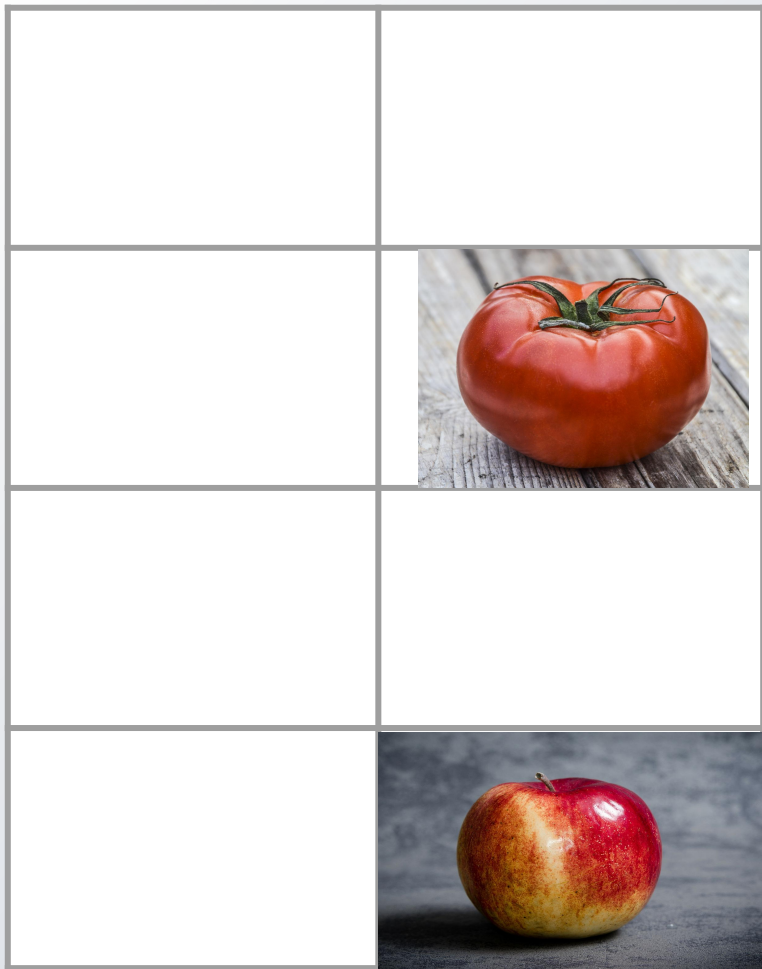
Shape

Texture









# Conventional programming

A programmer writes a computer program that precisely instructs a computer what to do to solve a particular problem.

Full of branching (if statements)



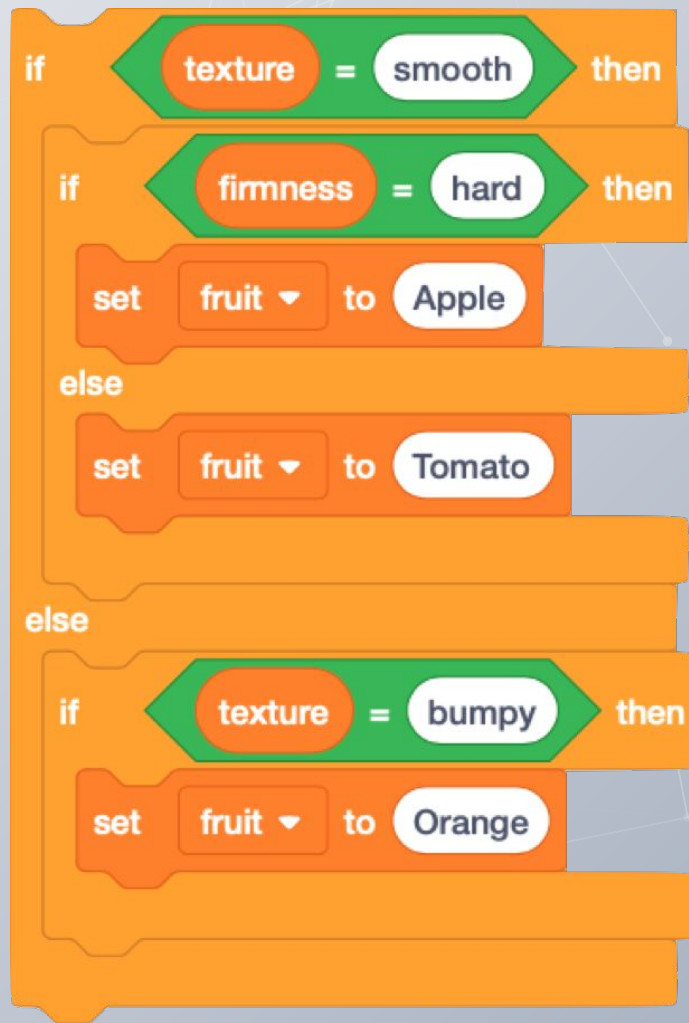


## Conventional programming

Decision trees can quickly grow out of control when we need to check multiple things.

And we also need to explain to the computer what the adjectives *smooth*, *hard*, *bumpy* mean.

And what if we need to add something else ... Pumpkin?



# AI Programming

But with an AI, our code gets so much simpler

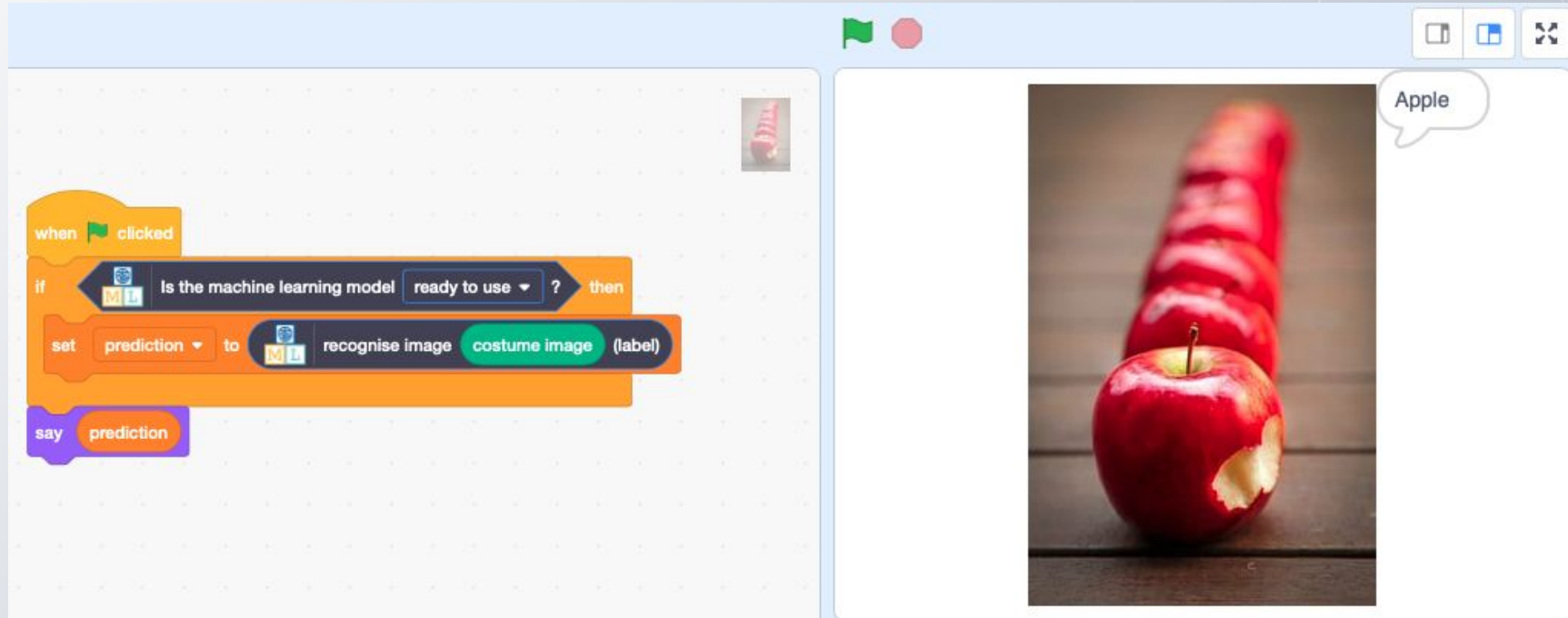
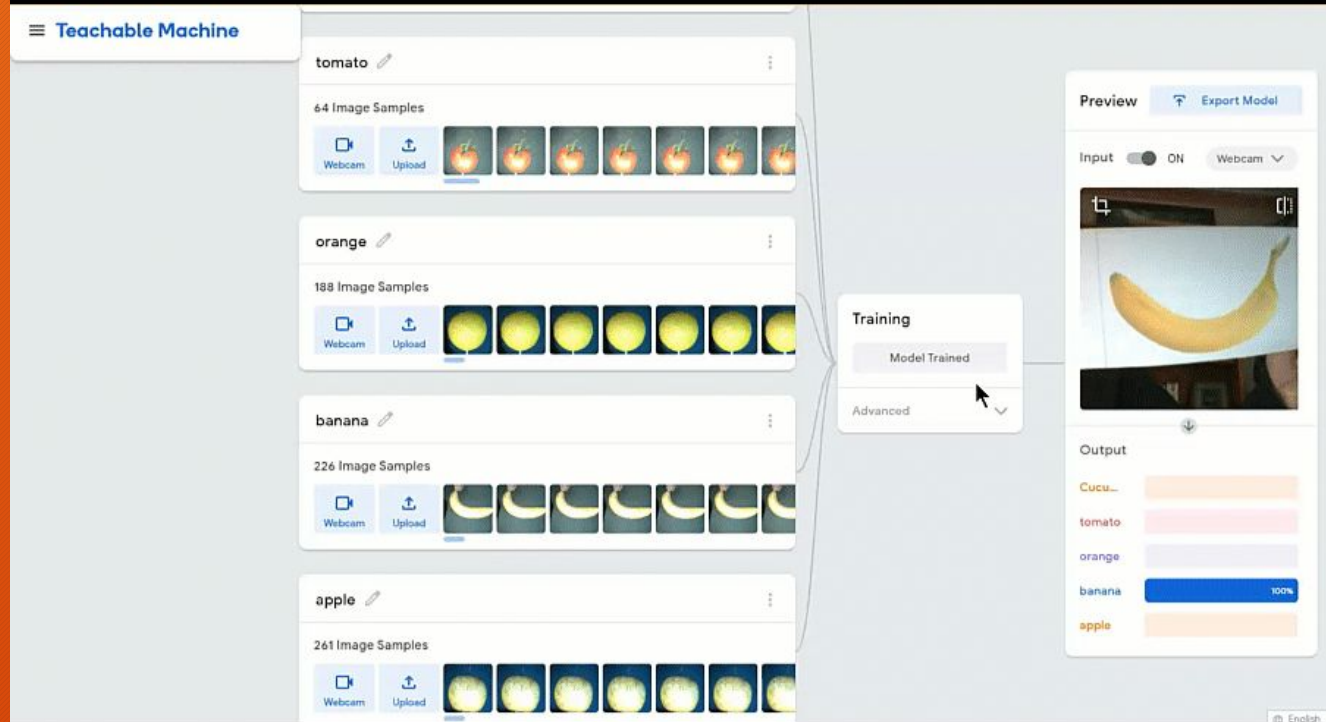


Image: Pixabay

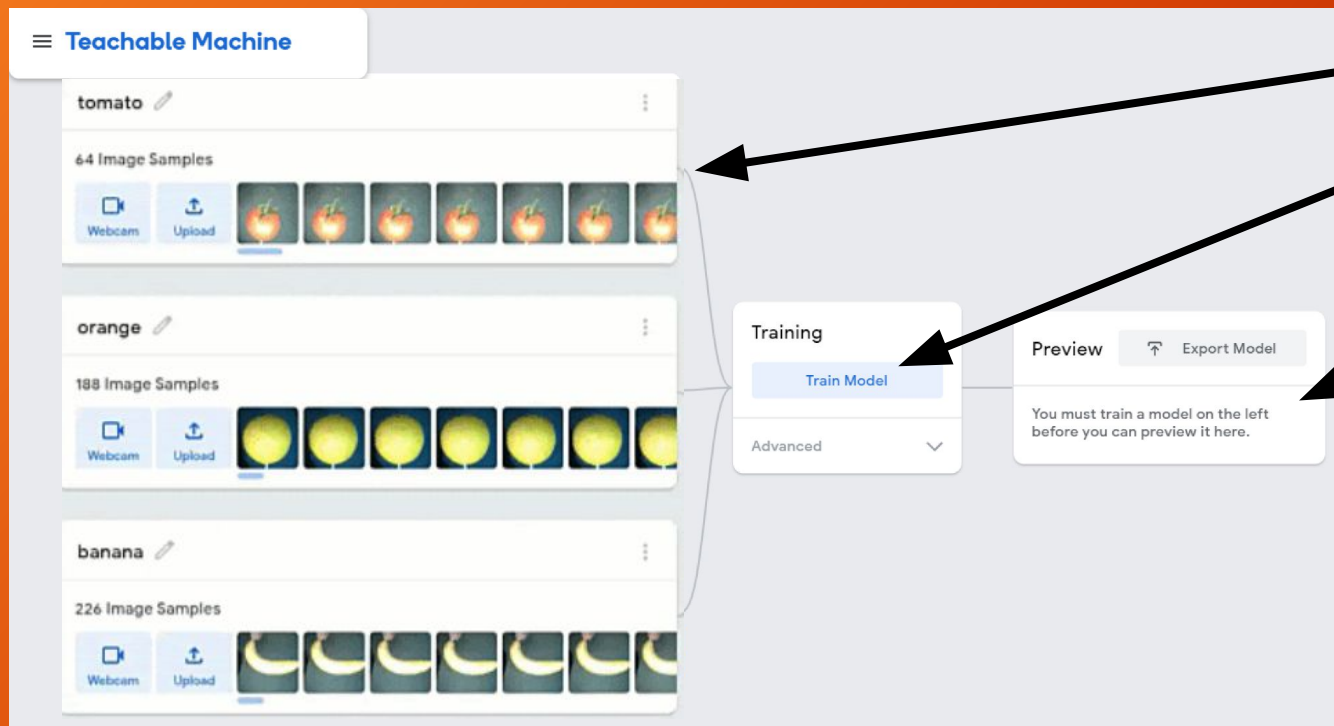
# Creating & training an AI model



<https://teachablemachine.withgoogle.com/models/oE7da2vzM/>

Use this [pre-made model](#) to test the AI to see how well it recognises fruit and vegetables. (You will need a device with a webcam).

## 2. Prepare an AI model: train the AI



1. **Collect data.**

2. **Train** the AI model.

3. **Test** the trained model.

Google Teachable Machine



# Conventional Programming

Algorithms

Implementation

- Programs involving branching, iterations

# Artificial Intelligence

Data

- Representation, Collection, Interpretation

Impact

Recognising voice, images, and filling in the data gaps.

# Considerations when teaching the curriculum

What questions you might ask yourself ...

What does covering Artificial Intelligence add to my teaching and learning program?

When and how to can I combine AI and conventional programming?

# Student projects

Scratch 3.0

Teachable Machine

My Computer Brain

Scratch 3.0  
Extensions

Pure conventional programming

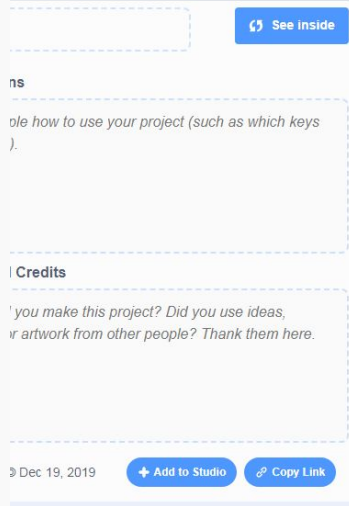
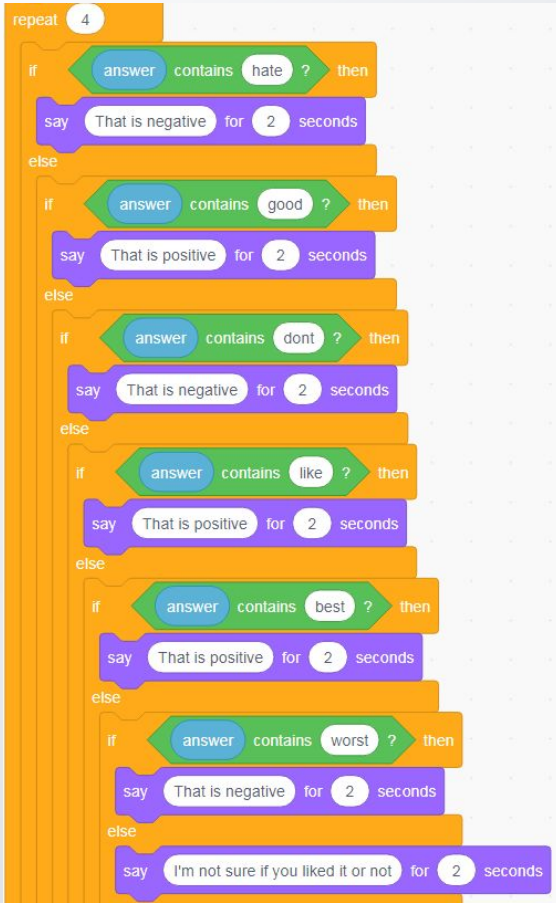
Pure AI

In practice, most classroom projects from Year 3 and up will be somewhere on the spectrum between pure conventional programming and pure AI.

For F-2, pure AI projects are highly engaging and require no implementation (coding)

# Conventional programs: an AI context

A real AI

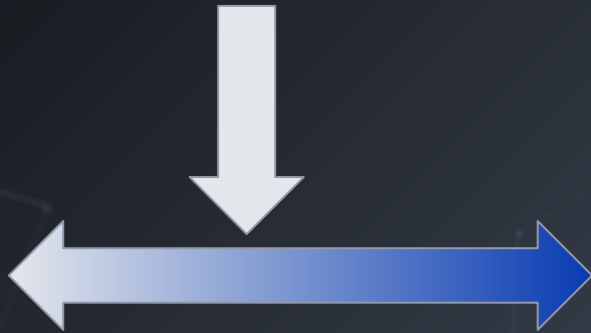


**LESSON:** CAN A COMPUTER RECOGNISE YOUR SENTIMENT? (Years 5-6)

**VIDEO:** Artificial Intelligence Explainers: Video 2: AI in our everyday life



# Conventional programming +AI



Using Scratch 3.0 to create  
computer program

And starting to add  
extensions

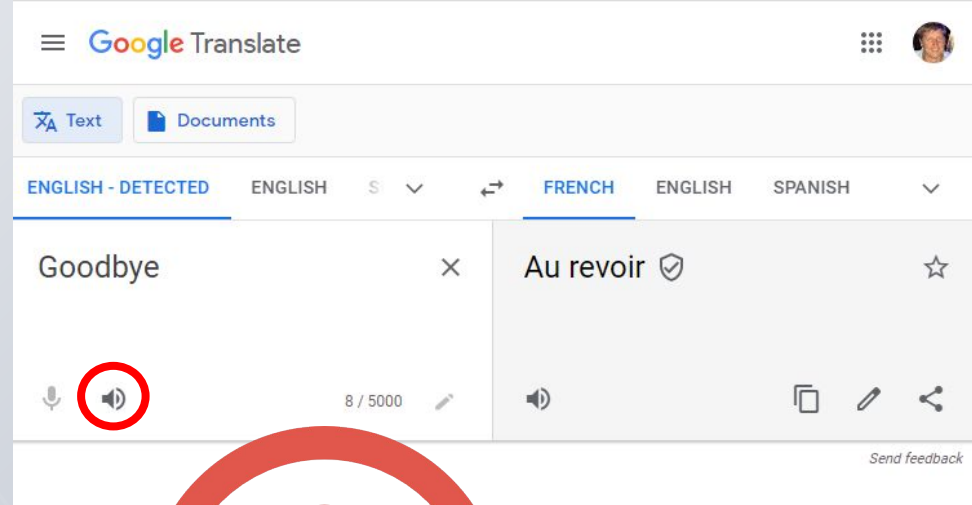
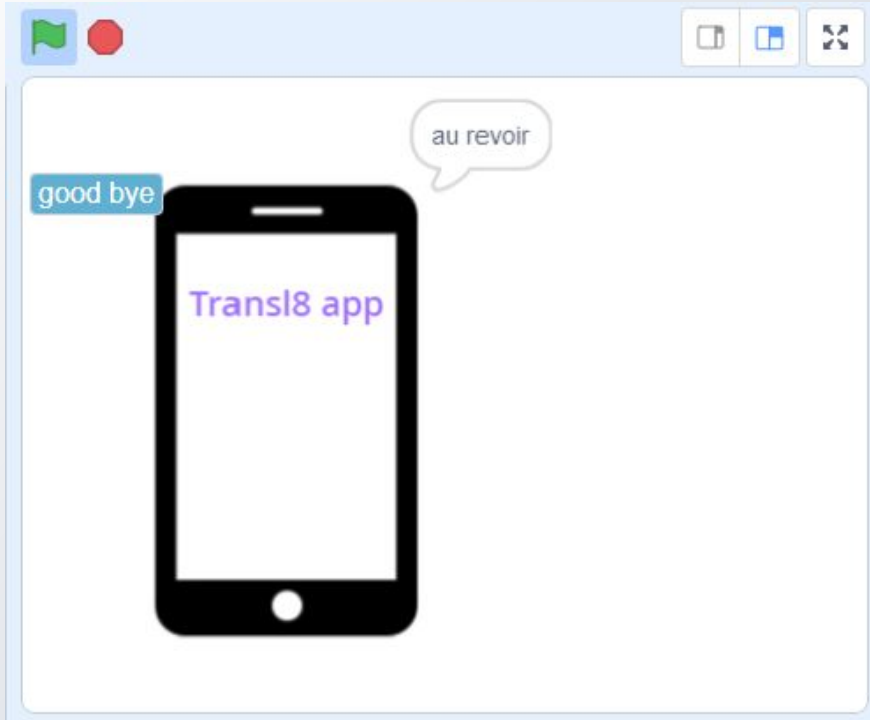
# Programs: low entry/high ceiling

## Translator program

Investigate examples

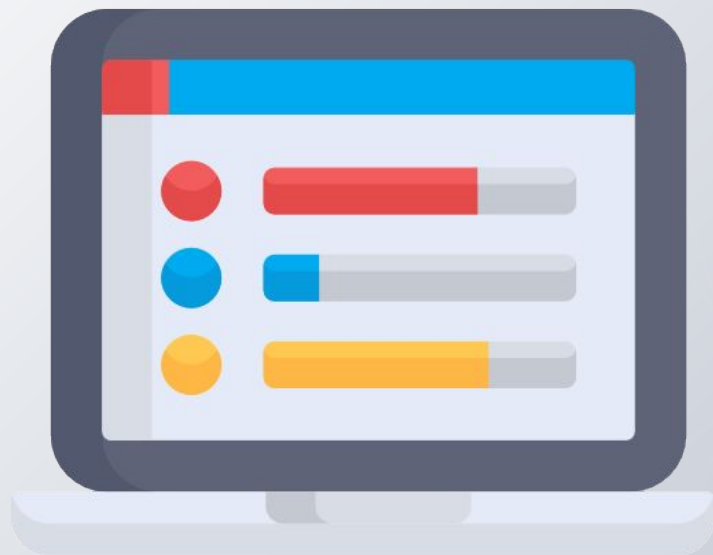


# Creator vs consumer



Google translate

# Poll



Martin Richards created a poll  
Poll: Not anonymous | Results shared

## How long would it take students to create a program that mirrors Google Translate?

- ☐ Too long: An impossible task
- ☐ As long as they have left in primary school
- ☐ At least 4 weeks working non stop
- ☐ About 10-15 min

Submit Vote

Too long: An impossible task 0% (0)

As long as they have left in primary school 0% (0)

At least 4 weeks working non stop 0% (0)

About 10-15 min 0% (0)

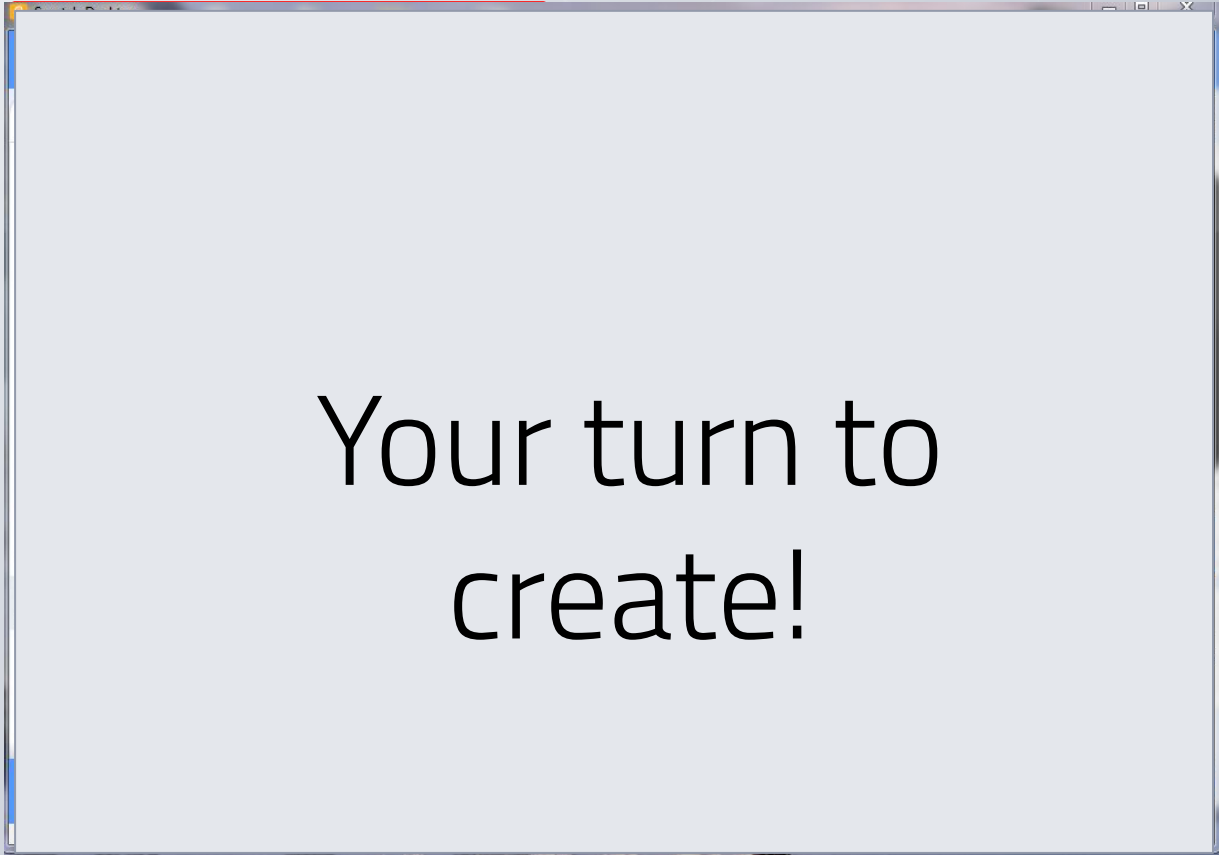
0 responses



# Text to speech recognition & Translation



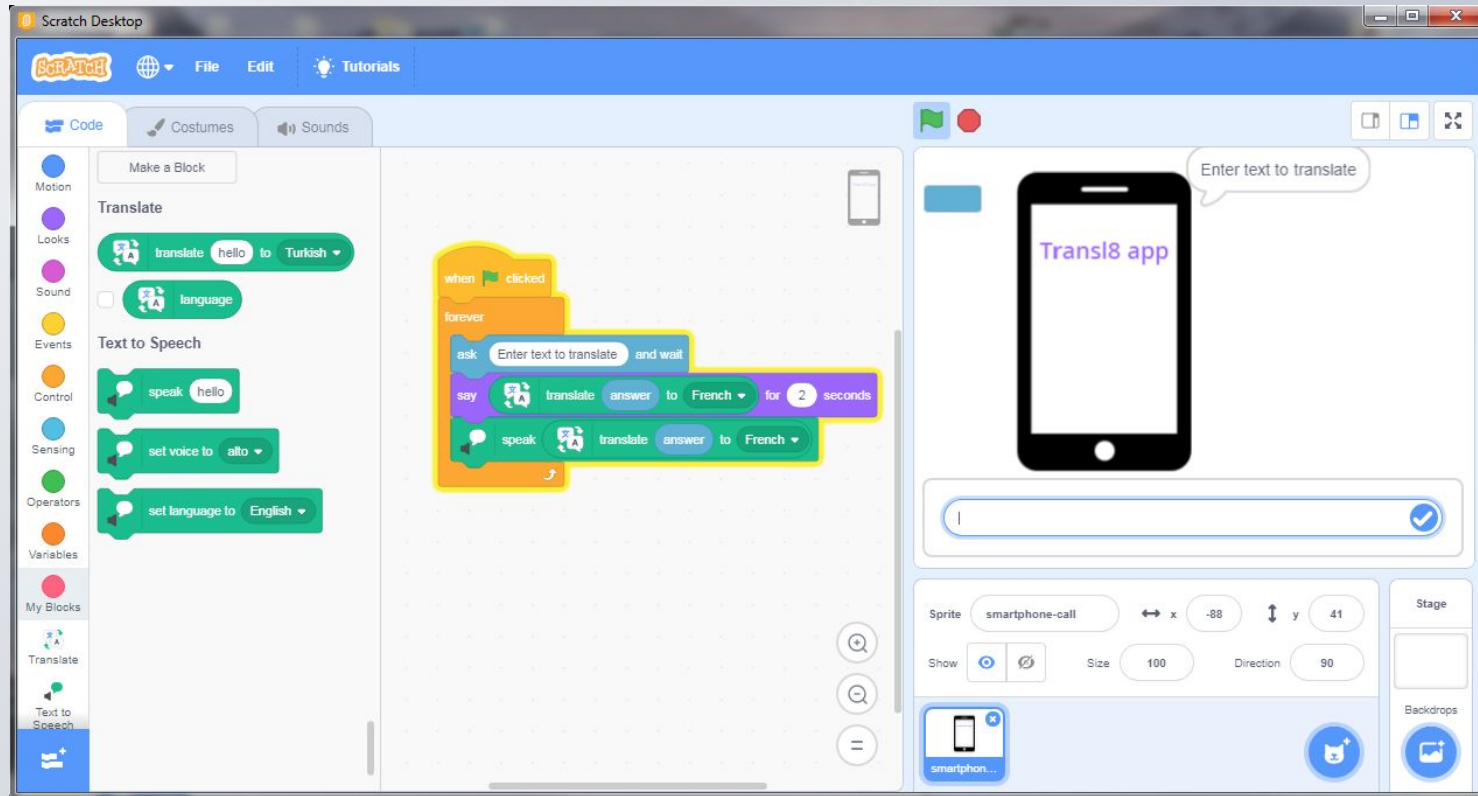
For our test program  
you can use  
browser-based  
version of [Scratch 3.0](#)

A screenshot of the Scratch 3.0 web editor interface. The main stage area is a large, empty white rectangle. In the center of this stage, the text "Your turn to create!" is displayed in a large, black, sans-serif font. The browser window's title bar and standard OS window controls (minimize, maximize, close) are visible at the top of the editor frame.

Your turn to  
create!

Speech to Text blocks and Translate blocks accessed in the additional blocks available in [Scratch 3.0](#)

# Sample code



# Remixing

In each programming lesson, sample code is provided.

Isn't that just giving students the answer?

This is an evidenced-based approach.

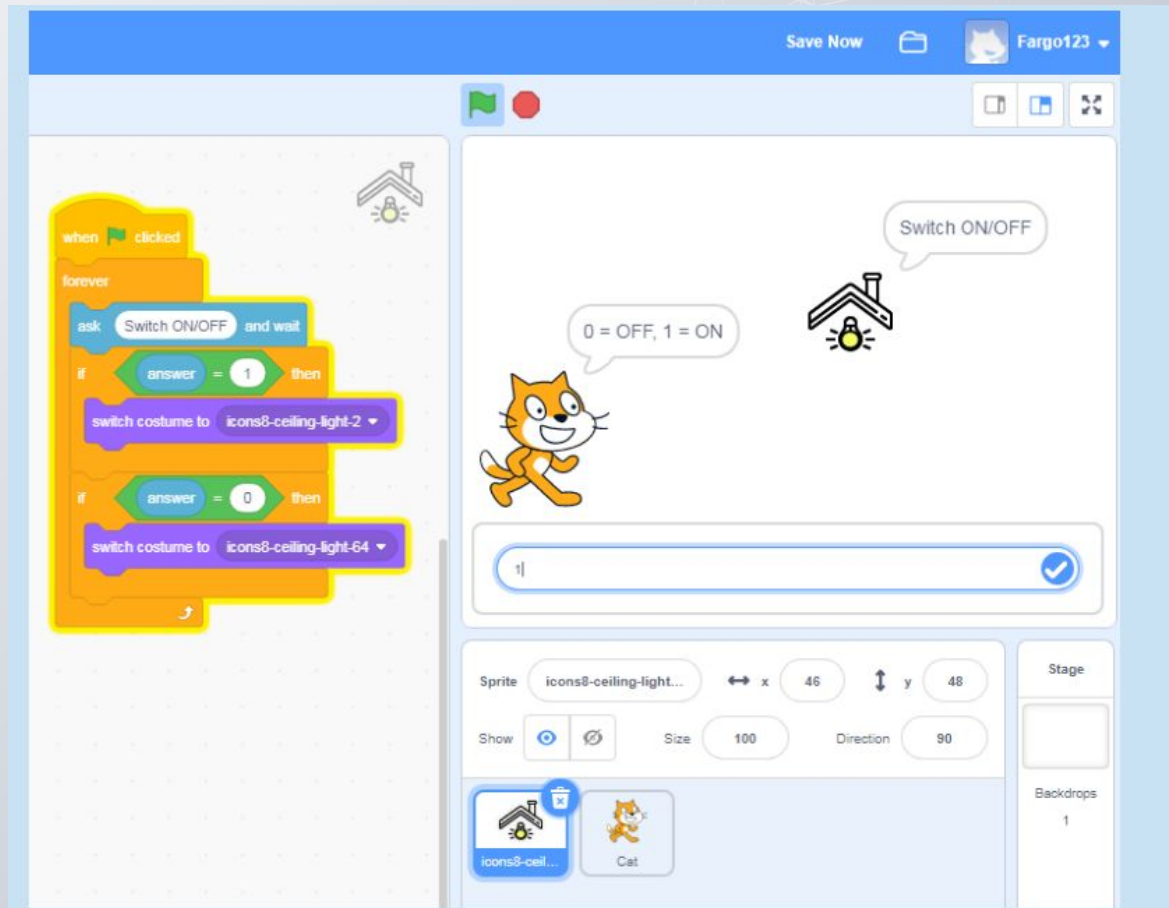


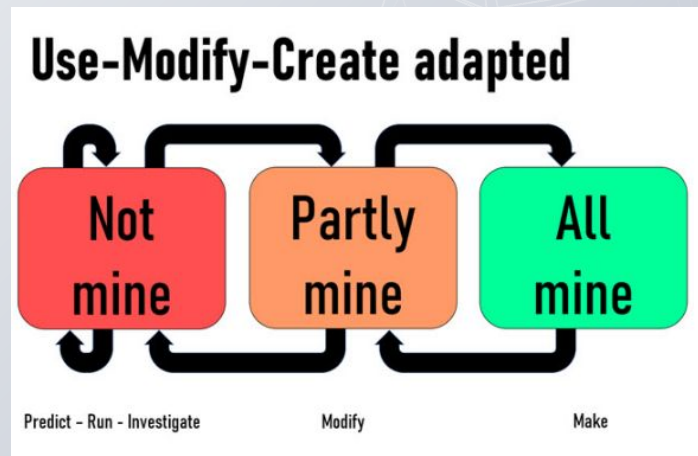
Image: Screen capture of Scratch 3.0 program showing if/then blocks based on user input answer (1), to turn light on

For sample code, see [Binary switch light on/off](#).



Image credit: [United Kingdom Department of Education](#)

PRIMM is one approach that can help teachers structure lessons in programming. PRIMM stands for Predict, Run, Investigate, Modify, Make.



Exploring pedagogies for teaching programming in school

This approach expands on the approach Use-Modify-Create. This is when a learner runs (uses) an existing program to see what it does, then modifies it, and then when able, creates a new project of their own.



# Remixing :Teaching tips

Using the sample code

1. Select link from the lesson plan which opens a Scratch File.
2. Select 'See inside'.
3. Save to a folder on your network.
4. Students open Scratch and 'Load from computer'
5. Save their own version.

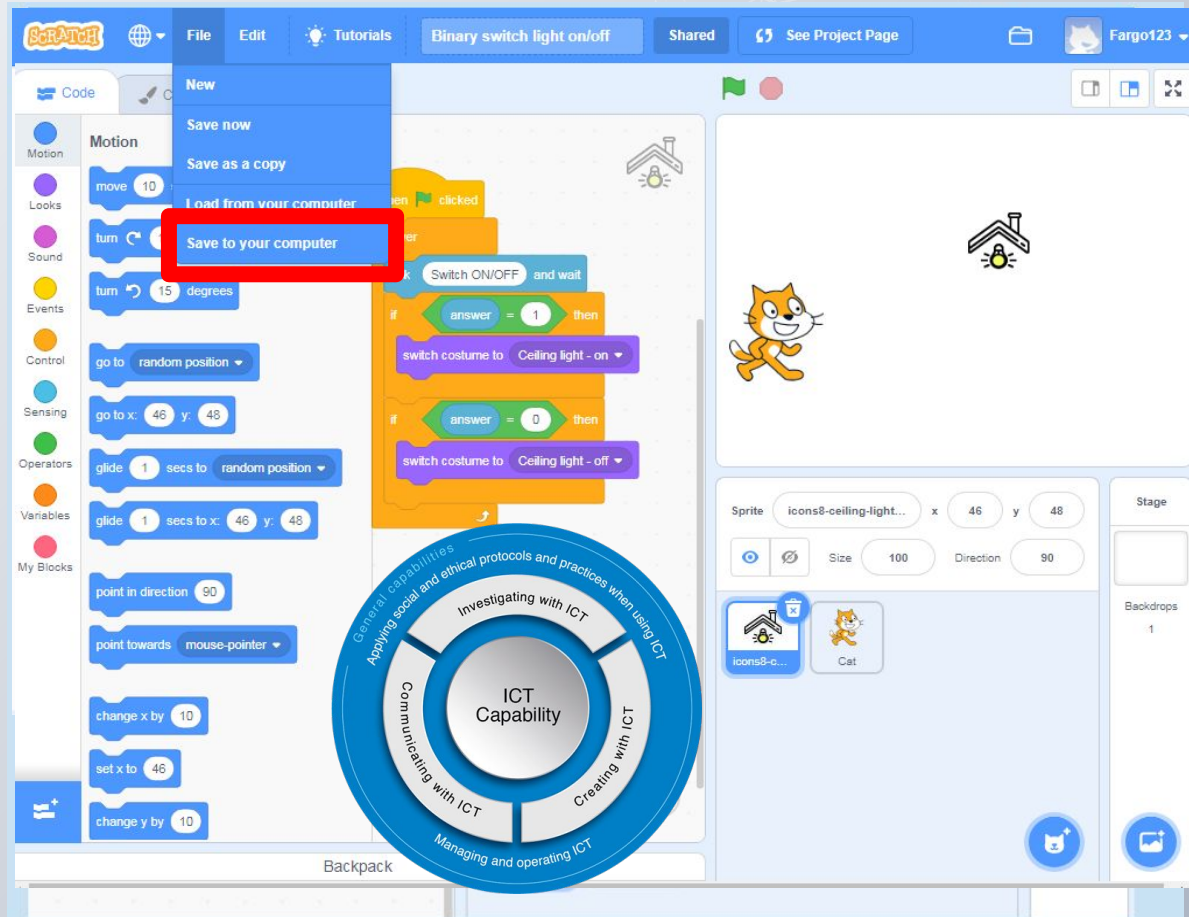


Image: Screen capture of Scratch 3.0 program showing if/then blocks based on user input answer (1), to turn light on

For sample code, see [Binary switch light on/off](#).

# Scratch 3.0

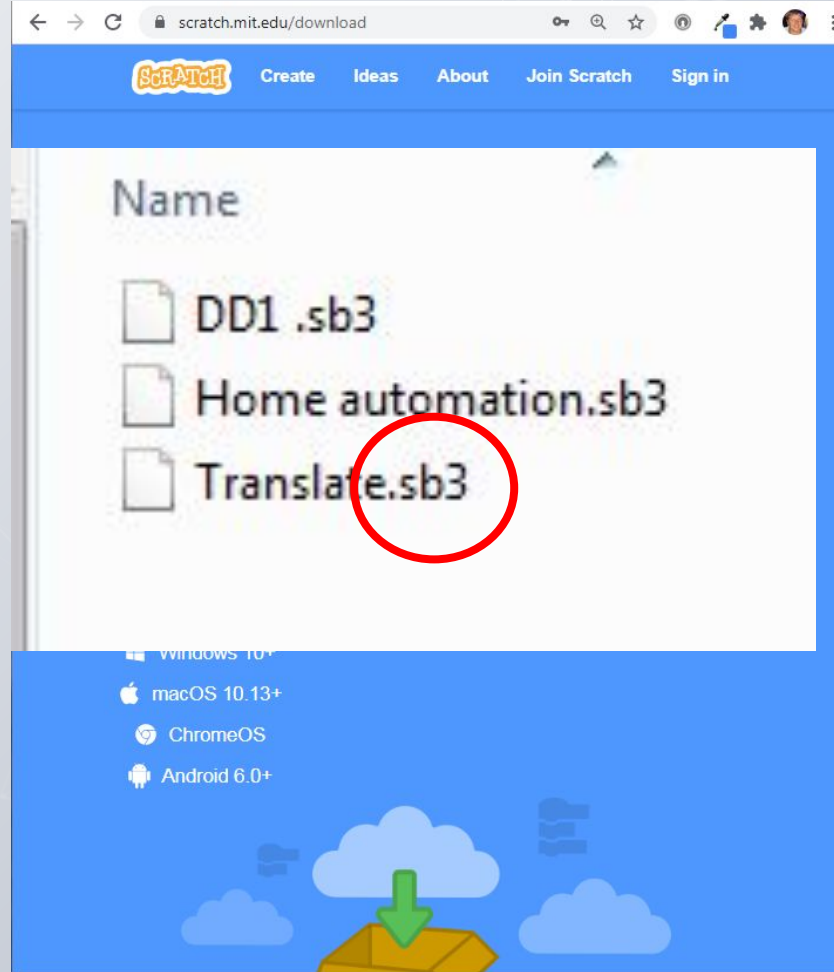
## Desktop version

1. Text to speech blocks
2. Translate blocks

To operate and test blocks requires internet.

Risk assessment (Privacy)

- No sharing/No comments section
- No log-in



# A chance to ask questions ...

## Use the chat...

How can you incorporate these teaching ideas?

How would your students handle the programming task?

What would you need to consider?

# Privacy considerations when using AI tools in the classroom





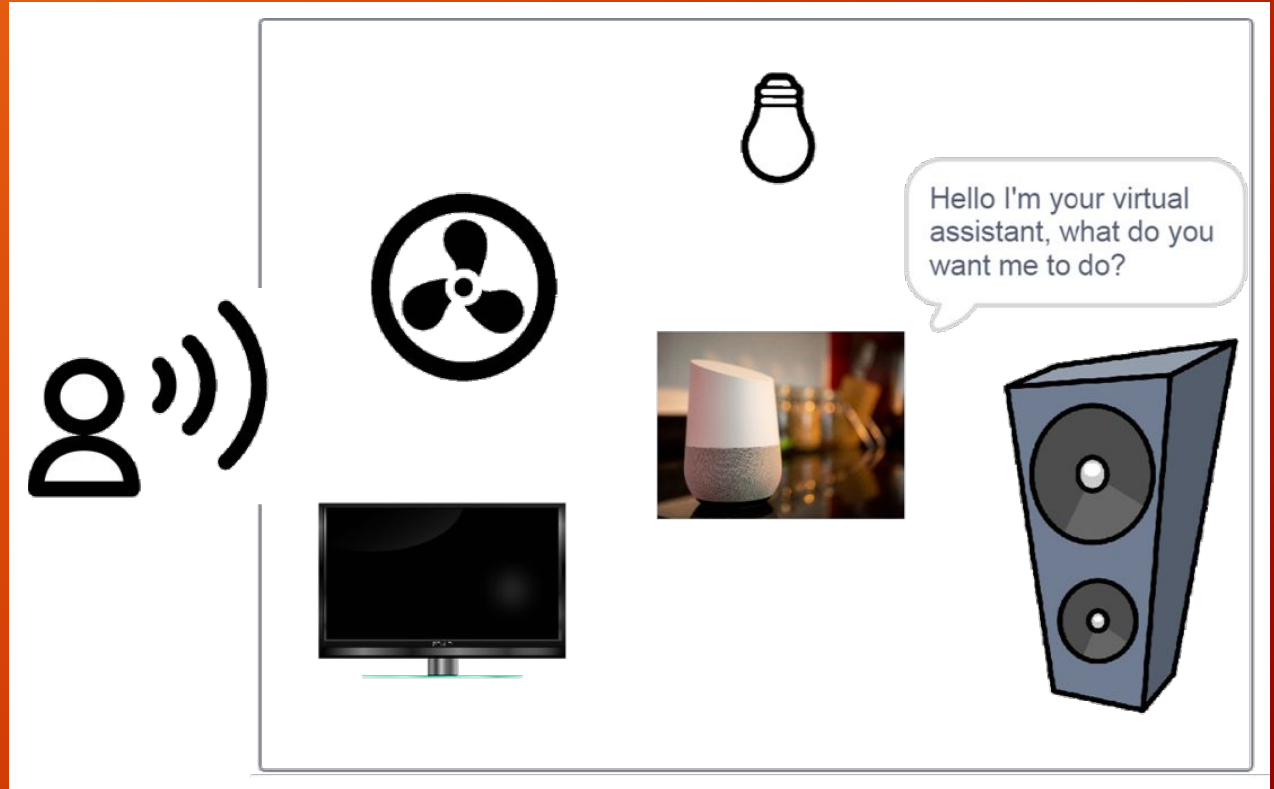
# E-safety: risk assessment

Consider	Yes	No	Suggestions to mitigate risks	
Have minimum age requirements for the technology or platform been adhered to?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>Check age appropriateness prior to use.</li> <li>Teach students about age recommendations and the reasons behind them.</li> </ul>	
Does the platform promote privacy and security for students and their accounts?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>Empower students to protect their privacy and explain how to adjust security settings.</li> </ul>	Information
Have parents/carers consented to their child using this technology or platform?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>Ensure appropriate consent has been provided by parents/carers. Some schools request consent to use a broad range of platforms at the start of the school year to avoid having to ask for consent each time a new platform is used. It's important to be as clear as possible about what this consent includes, as well as providing information on any possible risks to users and how the school mitigates them.</li> </ul>	ams. ock
Are staff comfortable and confident using the platform?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>Provide access to professional learning so staff are skilled in the platforms/technologies they use.</li> </ul>	and passwords. re a safer
Is there a staff member moderator for chat or comment functions?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>A staff member (or team) would ideally be appointed to moderate chat or comment functions, to encourage safe and positive interactions and to take down and investigate inappropriate posts.</li> </ul>	
Does the platform have capacity to report problems or misuse?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>All platforms should have terms of use that clearly identify inappropriate content or behaviour, and how to report problems or misuse.</li> <li>Visit <a href="#">The eSafety Guide</a> for more information.</li> </ul>	that have
Do all users know how to set the platforms' privacy settings?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>Share <a href="#">The eSafety Guide</a> with staff. This has links to the latest games, apps and social media, with tips on how to set privacy settings.</li> </ul>	orm it is
Have you identified how data is stored and used by the platform?	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> <li>Privacy issues arise when data is collected and not stored securely or shared inappropriately. Good practice is to find out how data will be stored and who has access.</li> <li>Check education department or sector policies to see if there are any standard protocols schools should follow, as well as advice about privacy legislation and data storage.</li> </ul>	ams. me across

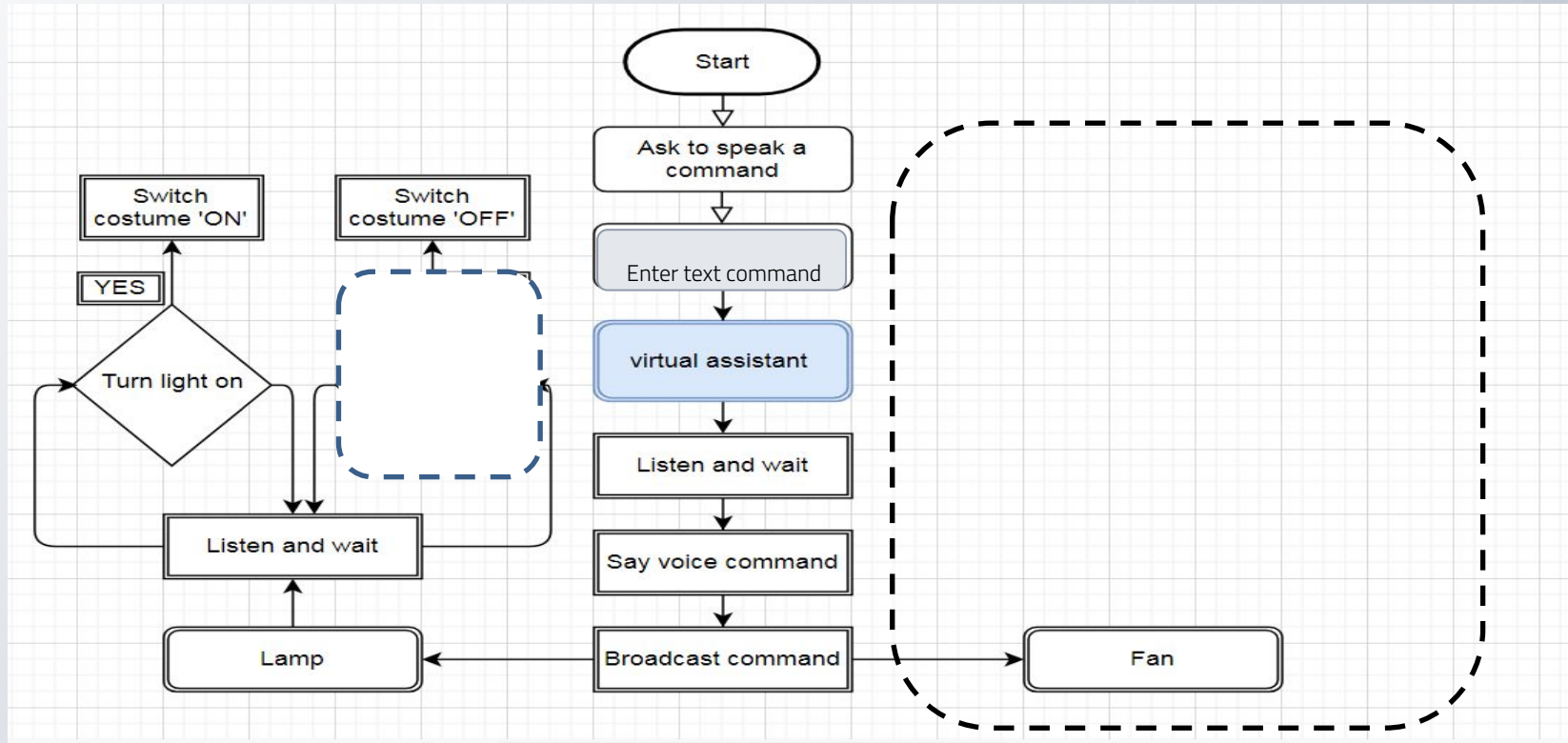


# Programming a virtual assistant

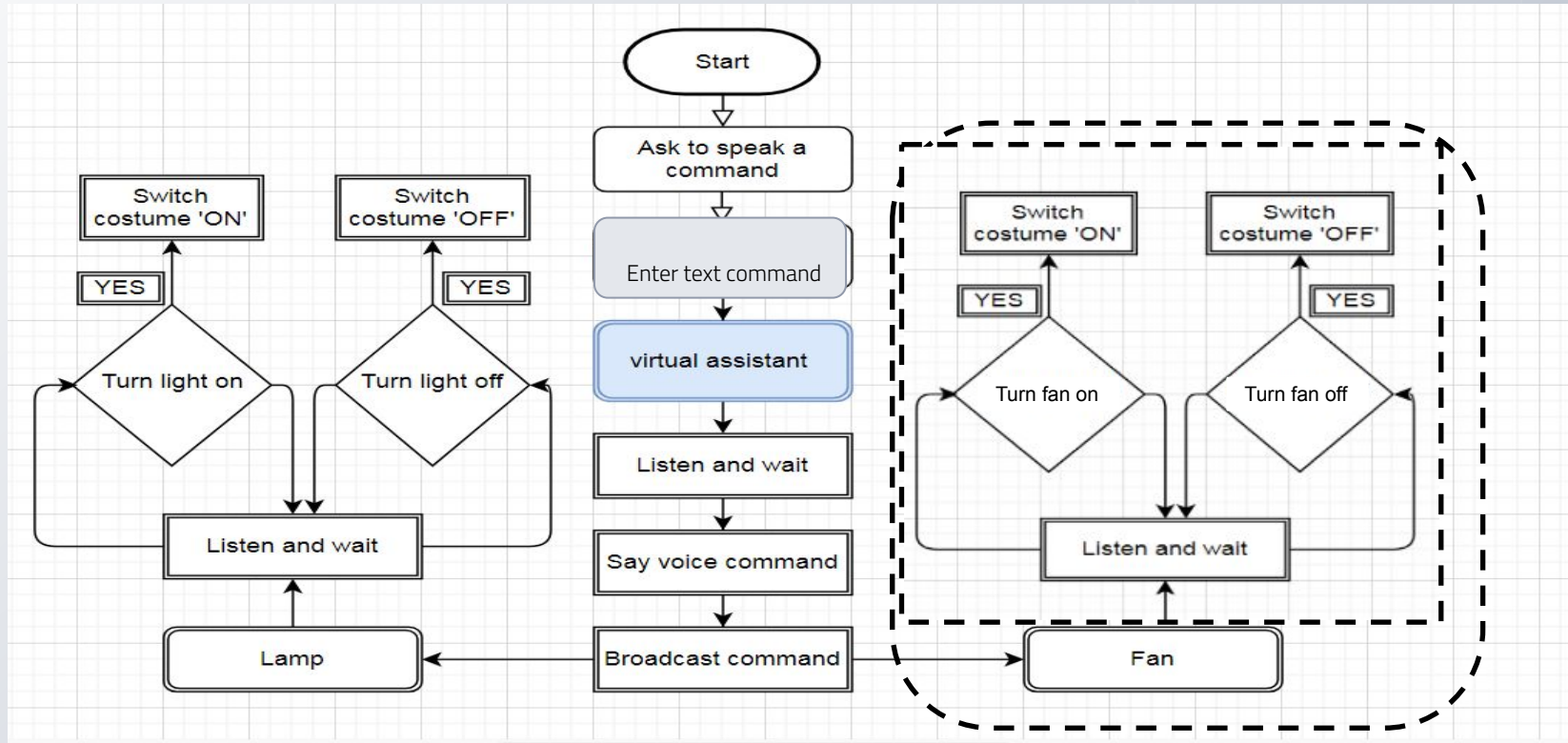
Apply  
Computational  
thinking



# An algorithm represented as a flowchart

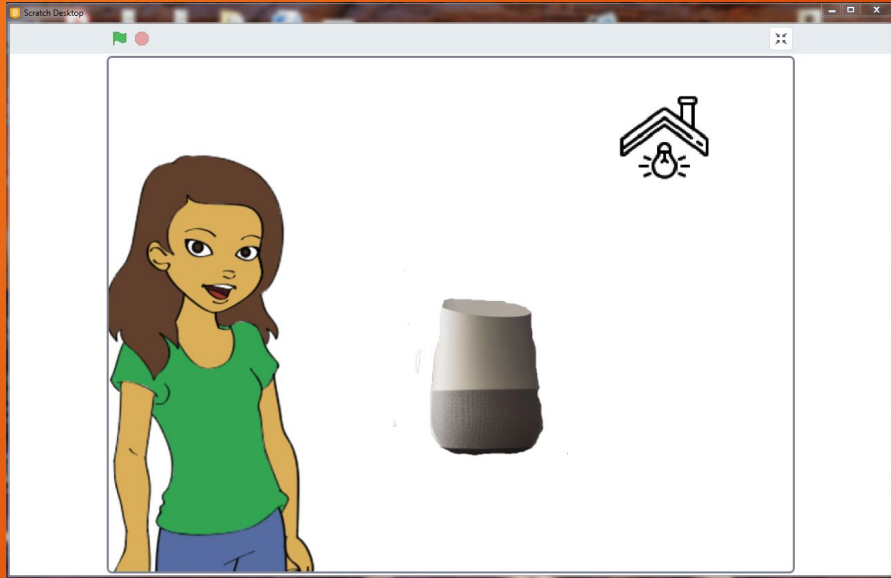


# An algorithm represented as a flowchart



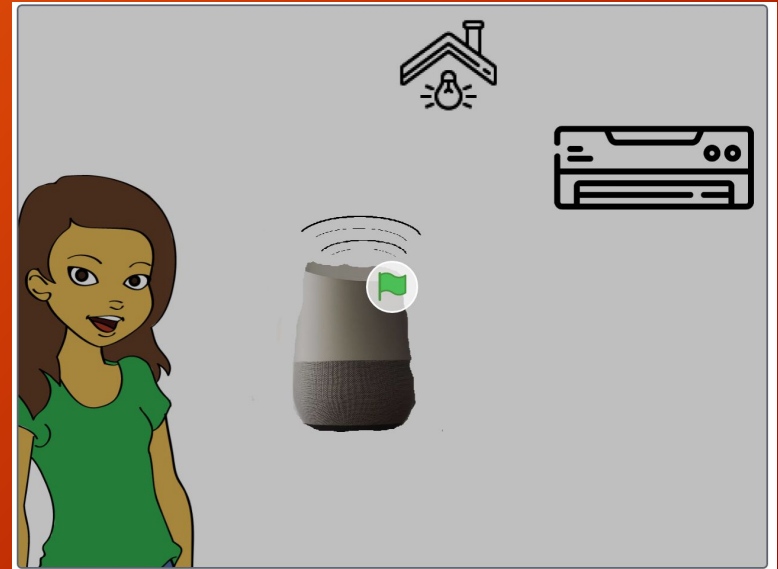
# Spot the difference

Open each program and find as many differences as you can.



Sample code:

<https://scratch.mit.edu/projects/559150293>



Sample code:

<https://scratch.mit.edu/projects/559169857/editor/>

# Programming a virtual assistant

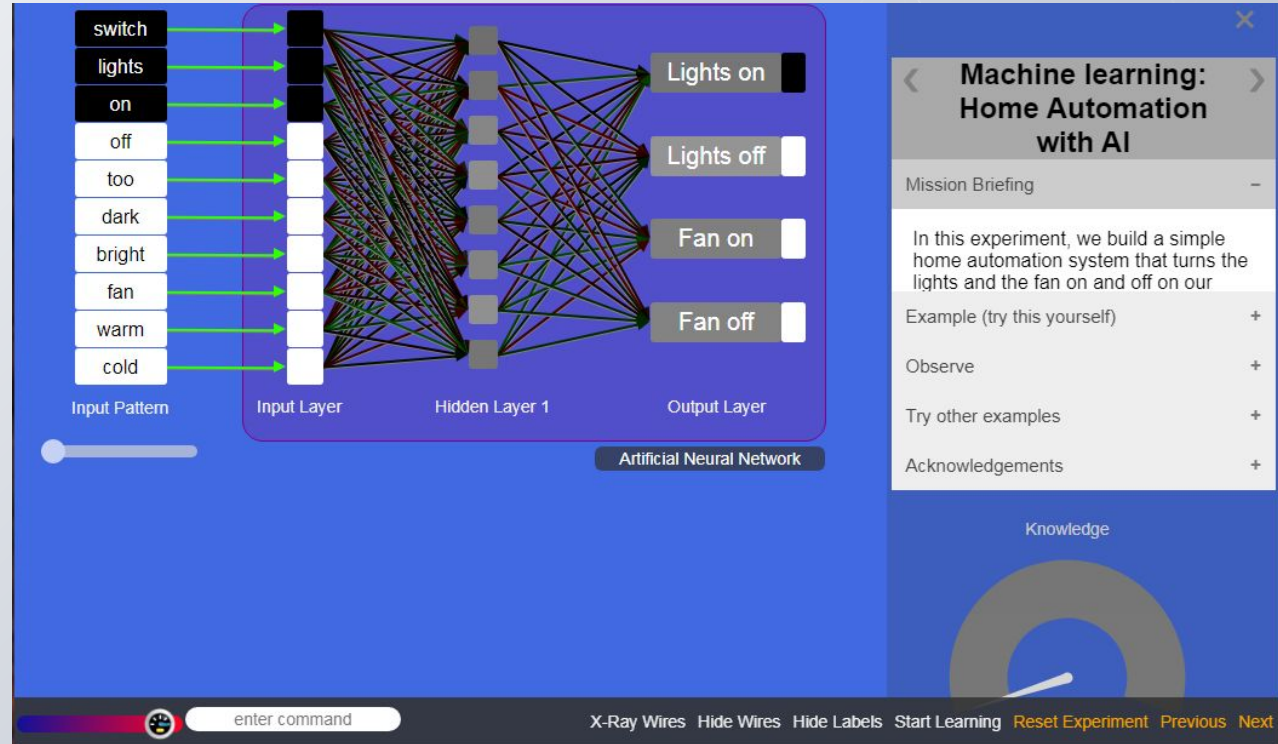
The image displays the Scratch Desktop interface, which is used for creating and programming virtual characters and environments. The interface is divided into several panels:

- Left Panel (Tools):** Contains various tool categories such as Operators, Motion, Looks, Sound, Events, Control, Sensing, Variables, My Blocks, and Translate. The Operators category is currently selected, showing blocks like "pick random", "if/then", "and", "or", "not", "join", "letter", "length of", "contains", "mod", "round", and "abs of".
- Center Stage:** The main workspace where the virtual assistant is programmed. It features a character (a woman) and a smart speaker (a virtual assistant). The stage is titled "Scratch Desktop".
- Right Panel (Stage):** Displays the current scene, including the character and the smart speaker. It also shows the "Stage" properties, such as the "Sprite" (icons8-ceiling-light...), "Size" (100), and "Direction" (90).
- Code Editor:** The central area where the program is written. It shows a sequence of blocks: a "when clicked" event, a "wait 2 seconds" block, and a "forever" loop. Inside the loop, there are "if" blocks that check the "answer" variable for "on" and "off" states, and "switch costume to" blocks that change the smart speaker's appearance accordingly.

The program is designed to respond to user input (clicks) by waiting 2 seconds and then checking the "answer" variable. If the answer is "on", it switches the smart speaker's costume to "Ceiling light - on". If the answer is "off", it switches the costume to "Ceiling light - off".



# An AI in action: virtual assistant



**LESSON:** Home automation with AI (Years 5-6)

# A chance to ask questions ...

## Use the chat...

How can you incorporate these teaching ideas?

How would your students handle the programming task?

What would you need to consider?

# Assessment

Artificial Intelligence is a rich field for assessment opportunities. Here are a few examples in the core concept areas of data, algorithms and implementation

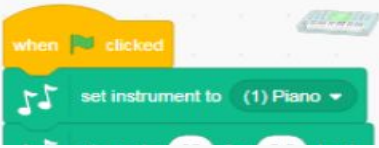



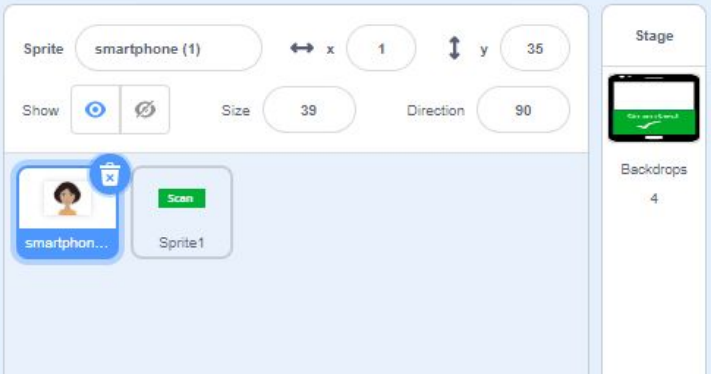
# How do I know if a student's program contains input, branching and repetition?

## Years 3-4

Implement simple digital solutions as visual programs with algorithms involving **branching (decisions)** and **user input**

## Years 5-6

Implement digital solutions as simple visual programs involving **branching, iteration** (repetition), and **user input**.

	Function:	 
Score (max. 40)	0-10	
Designing a digital solution that requires user input using visual programming language.	A sequence that <b>does not</b> include an option for user input.	  




# Project logs

**Project log**

Name \_\_\_\_\_

**My project**

I made a translation program.  
You type a word in and it translates it into German.



Date: 21 / 07 / 2021	Date: __ / __ / __	Date: __ / __ / __
<p><b>What I did:</b> I used some one else's program and changed it. I changed the language. Instead of typing I changed it to speech.</p> <p><b>I had trouble with:</b> Testing in a noisy room.</p> <p><b>I feel</b> 😞 😞 😞</p>	<p><b>What I did:</b></p> <p><b>I had trouble with:</b></p> <p><b>I feel</b> 😞 😞 😞</p>	<p><b>What I did:</b></p> <p><b>I had trouble with:</b></p> <p><b>I feel</b> 😞 😞 😞</p>

**I can**

- Use ask blocks
- Use speech blocks
- Use repeat until
- Use translate blocks
- Use if/then
- Create a variable
- Create a sprite
- Use an AI input
- Use a loop

My files are stored in the folder:

\_\_\_\_\_

Digital Technologies Hub is brought to you by Education Australia  
Creative Commons BY 4.0 license unless otherwise indicated.  
Australian Government Department of Education

# Selected Lesson Plans

An assorted selection of the  
many lesson plans available  
on the DT Hub

Specifically made for you



# Next steps

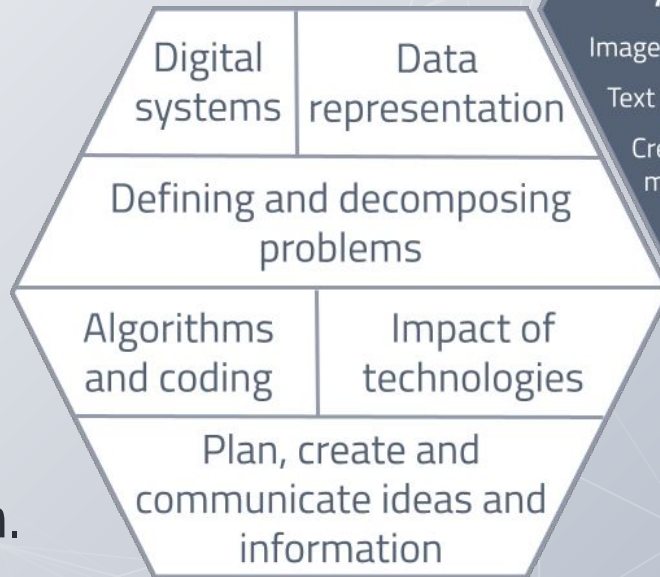
Making a commitment to implementing AI in your classroom

Use the chat to **write your idea** of where you will include AI as part of your teaching and learning program.

Connecting and sharing with the group.

email:

**[digitaltechnologieshub@esa.edu.au](mailto:digitaltechnologieshub@esa.edu.au)**



## AI topics

Image recognition

Text & speech recognition

Creating & using AI models (machine learning)

Bias and ethical issues



**DIGITAL  
TECHNOLOGIES  
HUB**