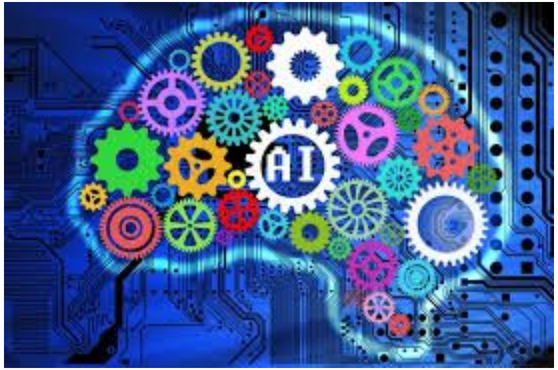


Artificial Intelligence

By Jen Graham
Australian Curriculum Lessons
https://www.australiancurriculumlessons.com.au/

Australian
CURRICULUM LESSONS

Australian Curriculum Outcomes

Technologies - Digital Technologies

Year 3 and 4 - Digital Technologies Knowledge and Understanding

Identify and explore a range of digital systems with peripheral devices for different purposes, and transmit different types of data (ACTDIK007

Recognise different types of data and explore how the same data can be represented in different ways (ACTDIK008

Year 5 and 6 - Digital Technologies Knowledge and Understanding

Examine the main components of common digital systems and how they may connect together to form networks to transmit data (ACTDIK014

Examine how whole numbers are used to represent all data in digital systems (ACTDIK015

Year 3 and 4 - Digital Technologies Processes and Production Skills

Collect, access and present different types of data using simple software to create information and solve problems (ACTDIP009

Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010

Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011

Explain how student solutions and existing information systems meet common personal, school or community needs (ACTDIP012

Plan, create and communicate ideas and information independently and with others, applying agreed ethical and social protocols (ACTDIP013

Year 5 and 6 - Digital Technologies Processes and Production Skills

Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information (ACTDIP016

Define problems in terms of data and functional requirements drawing on previously solved problems (ACTDIP017

Design a user interface for a digital system (ACTDIP018

Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019

Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020

Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021

Plan, create and communicate ideas and information, including collaboratively online, applying agreed ethical, social and technical protocols (ACTDIP022

Depends on students' prior knowledge and time spent on this unit will depend on the specific outcomes covered and achieved.

Learning Intentions

Content

To understand artificial intelligence by extending students knowledge on binary codes, data, coding and helping them see in to the future of artificial intelligence.

Language

To define words artificial intelligence, machine learning, coding, data and algorithm.

Social

To discuss their knowledge in this area with students in a whole class setting and small groups.

Success Criteria

I can understand the terms artificial intelligence, machine learning, coding and data.
I can use an artificial intelligence program such as Autodraw
I can use a simple coding program
I can explain artificial intelligence and machine learning to another person
I know where I can go if I want extra information on artificial intelligence

What is Artificial Intelligence?

Brainstorm either as a whole class, with a partner or independently a definition of artificial intelligence from your current knowledge.

There is no wrong idea. Have-a-go even if you don’t think you know the exact right answer.

Artificial Intelligence - Dictionary Definitions.

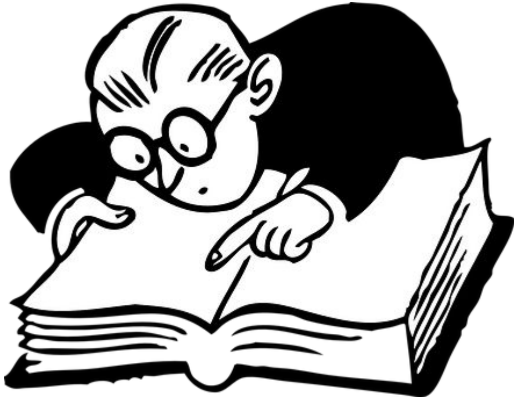
Oxford Dictionary “The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.”

Merriam-Webster defines artificial intelligence this way:

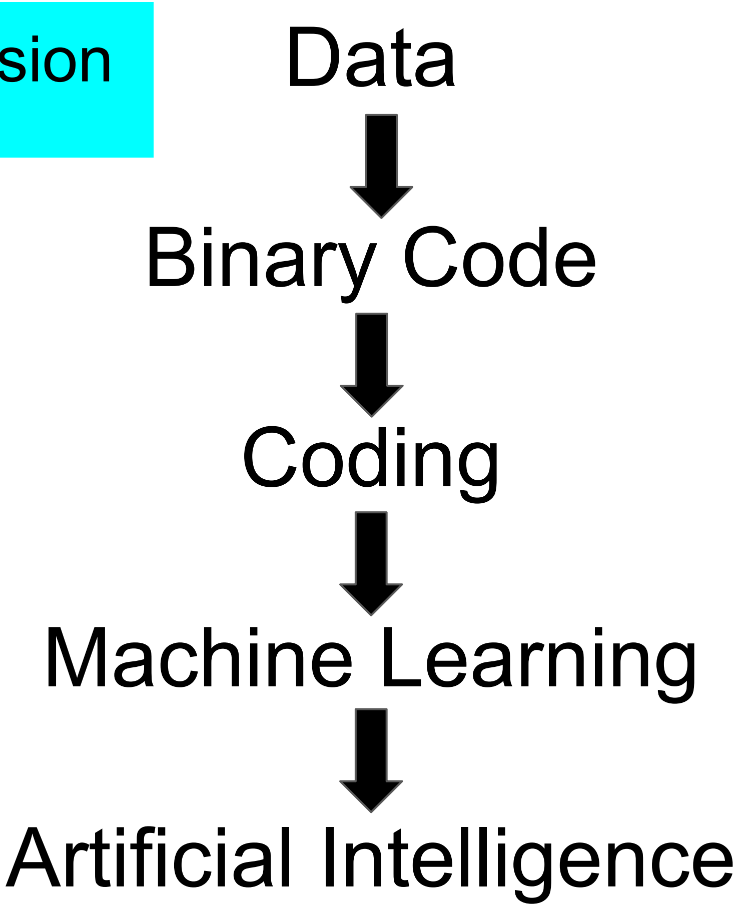
1. A branch of computer science dealing with the simulation of intelligent behavior in computers.
2. The capability of a machine to imitate intelligent human behavior.

The Encyclopedia Britannica states, “artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings.”

How close were you to the dictionary definition? Discuss.



Learning Progression



Data

Binary Code

Coding

Machine Learning

Artificial Intelligence

Data

Computers need DATA.

Do you have a good understanding of the word ‘data’?

Questions:
What does data mean?
What are some examples of data?
What do you remember from Chance and Data lessons?

Good websites with definitions of data
Level 3 and 4
http://www.amathsdictionaryforkids.com/qr/d/data.html
Level 5 and 6
https://www.mathsisfun.com/data/data.html
https://www.computerhope.com/jargon/d/data.htm

Remember to ask students to explain / define the word ‘data’ in their own words and have an area to display the language of Digital Technologies in the room for students to use as a reference.

Data: Binary System

Binary is a **base 2** number system invented by Gottfried Leibniz that is made up of only two numbers: 0 and 1. This number system is the basis for all **binary code**, which is used to write data such as the computer processor instructions used every day. (ComputerHope website - link below)

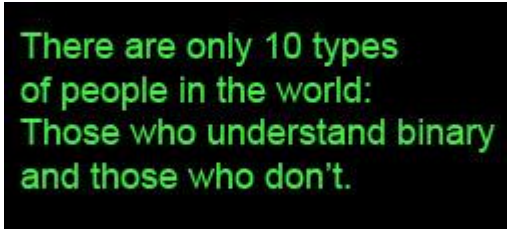
To write hello in binary code you would write the binary code for each letter - h e l l o

01001000 01100101 01101100 01101100 01101111

Practise writing your own name in binary code or a secret word to share with your friend
https://www.sciencefriday.com/educational-resources/write-your-name-in-binary-code/
**Check your answers here https://www.convertbinary.com/to-text/

Further information about the binary system
https://www.computerhope.com/jargon/b/binary.htm

http://www.amathsdictionaryforkids.com/qr/b/binarySystem.html



There are only 10 types
of people in the world:
Those who understand binary
and those who don't.

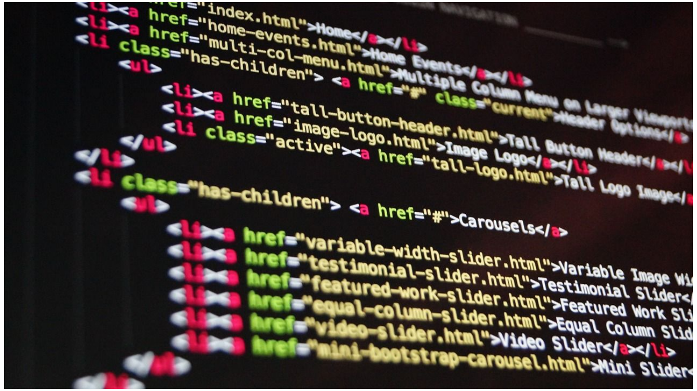
From Computer Hope

Transition from Binary Codes to Programming

Watch this short video to find out when the first computer program was created.

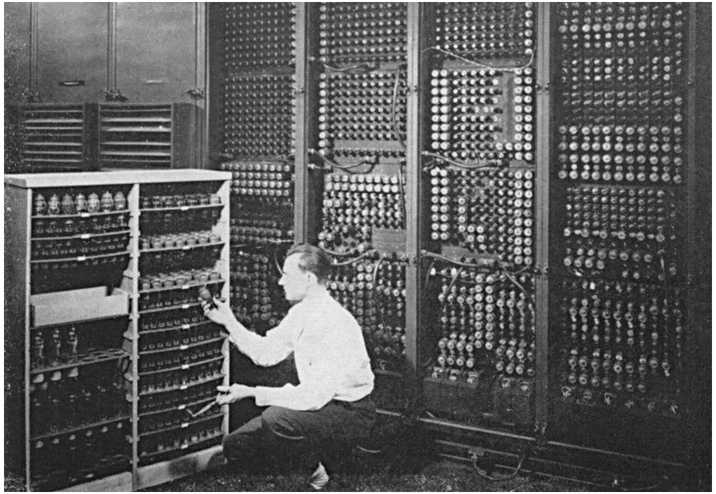
https://www.youtube.com/watch?v=Wchru8alhaE

If students are not familiar with writing a program practice computer programming by using a coding website / app such as www.code.org A great one to begin is https://studio.code.org/hoc/1 (no sign in required)



```
<li><a href="index.html">Home</a></li>
<li><a href="home-events.html">Home Events</a></li>
<li class="has-children"> <a href="multi-col-menu.html">Multiple Column Menu on Larger Viewport
    <ul>
        <li><a href="#" class="current">Header Options</a
        <li><a href="tall-button-header.html">Tall Button Header</a></l
        <li><a href="image-logo.html">Image Logo</a></li>
        <li class="active"><a href="tall-logo.html">Tall Logo Image</a
    </ul>
</li>
<li class="has-children"> <a href="#">Carousels</a>
    <ul>
        <li><a href="variable-width-slider.html">Variable Image Wi
        <li><a href="testimonial-slider.html">Testimonial Slider</
        <li><a href="featured-work-slider.html">Featured Work Sli
        <li><a href="equal-column-slider.html">Equal Column Slid
        <li><a href="video-slider.html">Video Slider</a></li>
        <li><a href="mini-bootstrap-carousel.html">Mini Slider<
    </ul>
```

Discuss students use of technology in their day-to-day life.



First computer
image from reddit

First Personal Computer
image from zdnet

Recent Computer
image from wealthlife

Digital Learning Glossary

Ensure their glossary or your Digital Learning Word Wall is up to date. Remember to use student's own language.

Artificial Intelligence

Programming

Coding

Binary Numbers

Data

Binary System

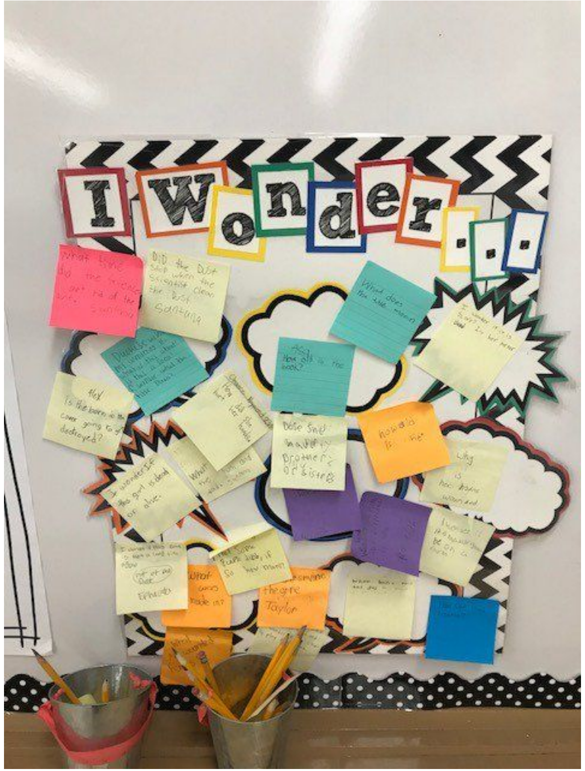
```
extract_num-
ber_and_incr (destination, source) int
*destination; unsigned char **source; {  extract_num-
ber (destination, *source); *source += 2; } #ifndef EXTRACT_MAC-
ROS #undef EXTRACT_NUMBER_AND_INCR #define EXTRACT_NUM-
BER_AND_INCR(dest, src) \ extract_number_and_incr (&dest, &src) #endif /*
not EXTRACT_MACROS */ #endif /* DEBUG */ ⊠ /* If DEBUG is defined, Regex prints
many voluminous messages about what it is doing (if the variable `debug' is nonzero).  If
linked with the main program in `iregex.c', you can enter patterns and strings interactively.
And if linked with the main program in `main.c' and the other test files, you can run the al-
ready-written tests.  */ #ifdef DEBUG /* We use standard I/O for debugging.  */ #include <stdio.h>
/* It is useful to test things that ``must'' be true when debugging.  */ #include <assert.h> static int
debug = 0; #define DEBUG_STATEMENT(e) e #define DEBUG_PRINT1(x) if (debug) printf (x) #define
DEBUG_PRINT2(x1, x2) if (debug) printf (x1, x2) #define DEBUG_PRINT3(x1, x2, x3) if (debug) printf
(x1, x2, x3) #define DEBUG_PRINT4(x1, x2, x3, x4) if (debug) printf (x1, x2, x3, x4) #define DE-
BUG_PRINT_COMPILED_PATTERN(p, s, e)\ if (debug) print_partial_compiled_pattern (s, e) #define DE-
BUG_PRINT_DOUBLE_STRING(w, s1, sz1, s2, sz2)     \ if (debug) print_double_string (w, s1, sz1, s2, sz2)
extern void printchar(); /* Print the fastmap in human-readable form.  */ void print_fastmap (fastmap)
char *fastmap; { unsigned was_a_range = 0; unsigned i = 0;   while (i < (1 << BYTEWIDTH)) { if (fastmap[i++])
{ was_a_range = 0; printchar (i - 1); while (i < (1 << BYTEWIDTH)  &&  fastmap[i]) { was_a_range = 1; i++; } if
(was_a_range) { printf ("-"); printchar (i - 1); } } } putchar ('\n');  } /* Print a compiled pattern string in hu-
man-readable form, starting at the START pointer into it and ending just before the pointer END.  */ void
print_partial_compiled_pattern (start, end) unsigned char *start; unsigned char *end; { int mcnt, mcnt2; un-
signed char *p = start; unsigned char *pend = end; if (start == NULL) { printf ("(null)\n"); return; } /* Loop over
pattern commands.  */ while (p < pend) { switch ((re_opcode_t) *p++) { case no_op: printf ("/no_op");
break; case exactn: mcnt = *p++; printf ("/exactn/%d", mcnt); do { putchar ('/'); printchar (*p++); }
while (--mcnt); break; case start_memory: mcnt = *p++; printf ("/start_memory/%d/%d", mcnt,
*p++); break; case stop_memory: mcnt = *p++; printf ("/stop_memory/%d/%d", mcnt, *p++);
break; case duplicate: printf ("/duplicate/%d", *p++); break; case anychar: printf ("/anychar");
break; case charset: case charset_not: { register int c; printf ("/charset%s", (re_opcode_t) *(p -
1) == charset_not ? "_not" : ""); assert (p + *p < pend); for (c = 0; c < *p; c++) { unsigned bit;
unsigned char map_byte = p[1 + c]; putchar ('/'); for (bit = 0; bit < BYTEWIDTH; bit++) if
(map_byte & (1 << bit)) printchar (c * BYTEWIDTH + bit); } p += 1 + *p; break; } case beg-
line: printf ("/begline"); break; case endline: printf ("/endline"); break; case on_failure_-
jump: extract_number_and_incr (&mcnt, &p); printf ("/on_failure_jump/0/%d", mcnt);
break; case on_failure_keep_string_jump: extract_number_and_incr (&mcnt, &p); printf
("/on_failure_keep_string_jump/0/%d", mcnt); break; case dummy_failure_jump: ex-
tract_number_and_incr (&mcnt, &p); printf ("/dummy_failure_jump/0/%d", mcnt); break;
case push_dummy_failure: printf ("/push_dummy_failure"); break; case may-
be_pop_jump: extract_number_and_incr (&mcnt, &p); printf
("/maybe_pop_jump/0/%d", mcnt); break; case pop_failure_-
jump: extract_number_and_incr (&mcnt, &p); printf ("/pop_-
failure_jump/0/%d", mcnt); break;         case jump_past_alt:
extract_number_and_incr (&mcnt, &p); printf ("/-
```

I Wonder…….

Create an ‘I Wonder’ place for students to write their ‘wonderings’ - use either sticky notes on a class wall/poster or in student’s Inquiry books.

Encourage all thinking.

Image from Kramer1BSchool Twitter


I Wonder...
Did the Dust stop when the scientist clean the Dust
How old is the book?
Is the barn in the cover going to get destroyed?
I wonder if the girl is dead or alive.
Does she have any brothers or sisters
How old is she
Why
Taylor

Machine Learning

So now you have a glimpse of how we input data into a computer using binary codes and programming codes. Watch this video to help explain our next step….machine learning. https://www.youtube.com/watch?v=ukzFI9rgwfU

Use machine learning in Autodraw.

Autodraw

AutoDraw is a new kind of drawing tool that pairs the magic of machine learning with drawings from talented artists to help everyone create anything visual, fast.https://www.autodraw.com/

More examples and machine learning programs
https://machinelearningforkids.co.uk/#!/worksheets

http://www.mindmapsoft.com/mind-maps-artificial-intelligence-ai/ (AI Applications)

Discuss this image and all the ways AI is used in today’s society.

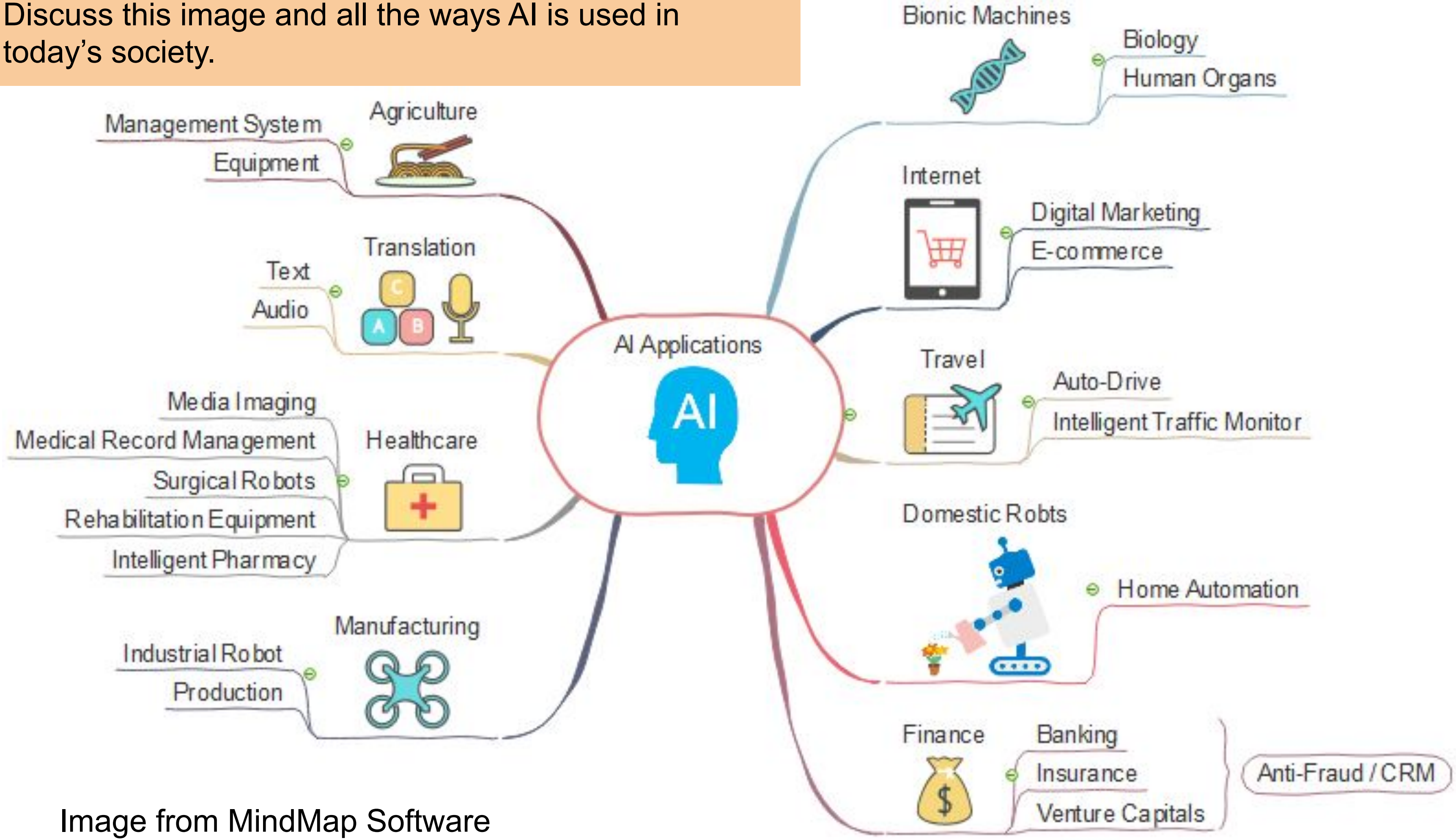


Image from MindMap Software

Where will AI lead us to next?
Choose one of the below which all look at AI in the future. Please view first to ensure it is suitable for your cohort of students. Please note when teaching Digital Technologies that it is rapidly changing so ensure you keep updated by searching online when needed.

https://www.youtube.com/watch?v=xoSsRMe6Uig

https://www.youtube.com/watch?v=wTbrk0suwbg

https://www.youtube.com/watch?v=rnBAdnNIIXk

What things will disappear in just 20 years?
https://www.youtube.com/watch?v=yulvmU0BLiI

15 jobs that will disappear in the next 20 years due to AI

https://www.youtube.com/watch?v=r211u89eUaY

Thank you for reading this unit, we hope you and your students found it useful.

Please head to www.australiancurriculumlessons.com.au for more free units of work and lessons.

I highly recommend for any educator to complete the free online course provided by The University of Adelaide to upskill in Digital Technologies.

https://csermoocs.adelaide.edu.au/

“We run a range of Digital Technologies programs for Australian teachers, including our free, online CSER MOOC courses, free professional learning events, and our National Lending Library”. (The University of Adelaide)