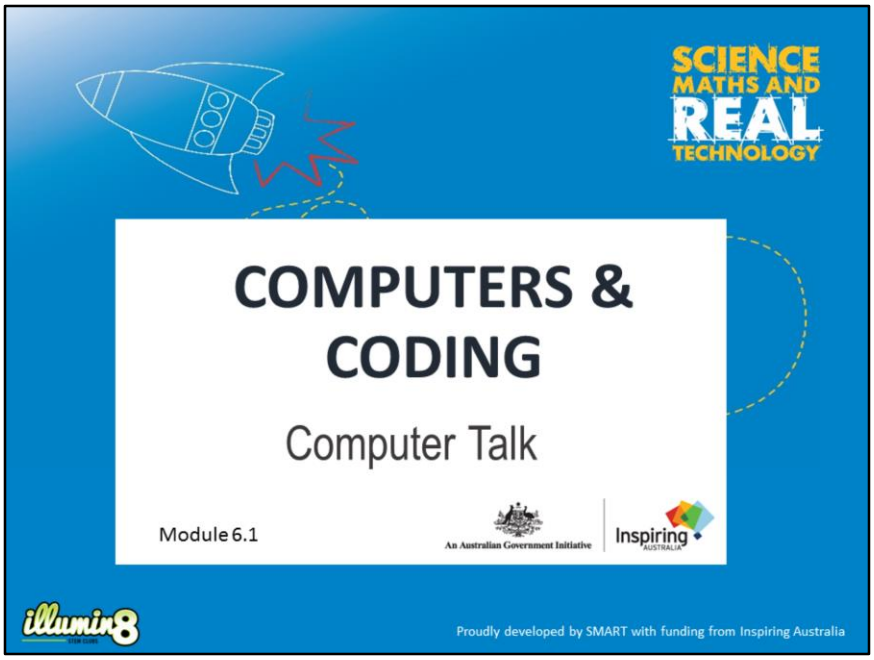
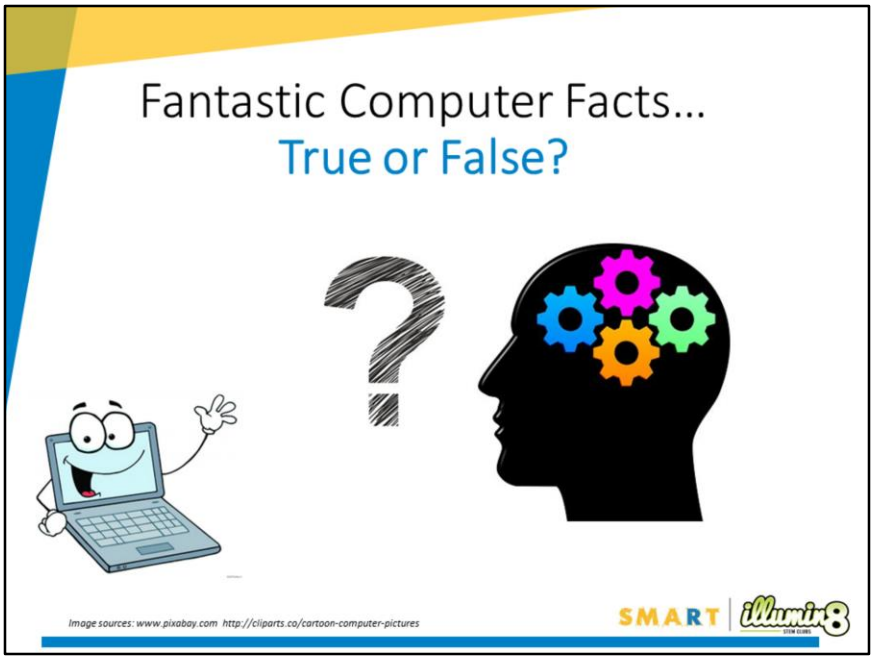

SCIENCE
MATHS AND
REAL
TECHNOLOGY
COMPUTERS &
CODING
Computer Talk
Module 6.1
An Australian Government Initiative
Inspiring
AUSTRALIA
illumin8
Proudly developed by SMART with funding from Inspiring Australia

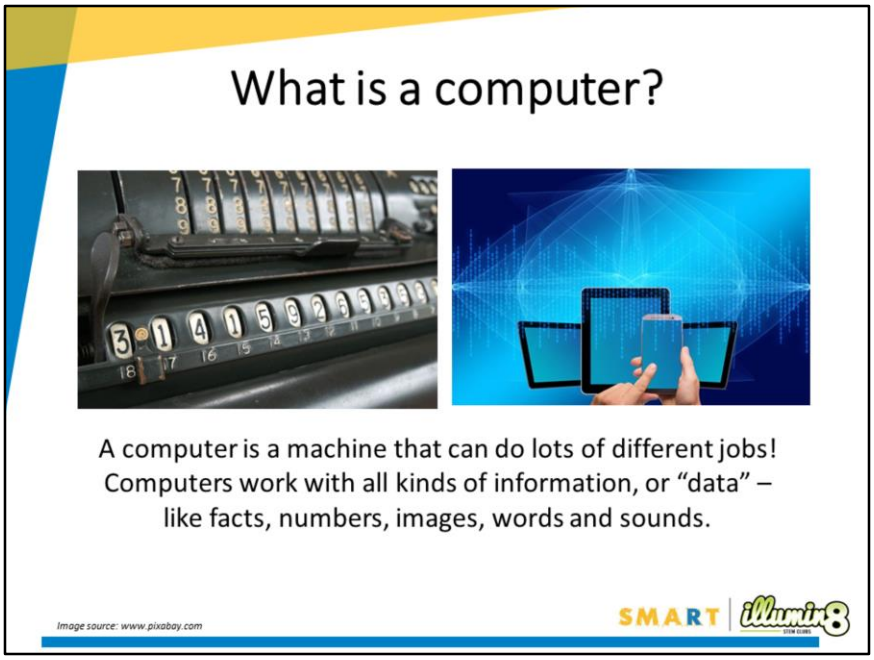


Welcome! In this session we will explore computers and coding – and how to "talk" to computers!
First, lets play some Fact or Fiction Games.

Game 1: Of the following three statements…. Which one is false?
1. Smartphones and tablets are computers
2. The oldest computers date back to at least 100BC
3. All computers require electricity
Number 3 is false – computers may be mechanical!
See: http://www.wisegeek.com/what-is-a-mechanical-computer.htm
http://www.dailymail.co.uk/sciencetech/article-2853082/World-s-oldest-computer-ancient-thought-Antikythera-Mechanism-created-205-BC-study-claims.html

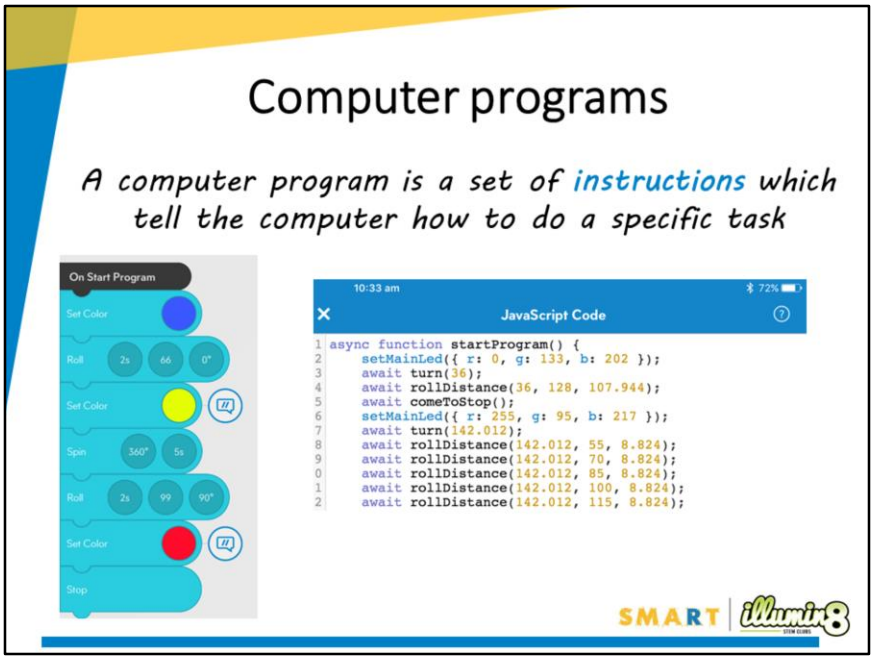
Game 2: Activity 6.1.1
Ask all students to stand up.
Read out the facts listed in the coordinator notes, and ask students to decide if they are true or false.
If students believe it is true, they put their hands on their heads. If they believe it is false they put their hands on their hips. Announce if the fact is true or false. Those students who guessed correctly stay standing, everyone else sits down. Repeat till there is only one student

standing or you run out of facts.



Discuss and share ideas about what computers are. What do we use computers for today?
What might we want to use computers for in the future?
What kinds of computers do we use?
Personal computers (PC's) and also known as desktop computers. We use laptops, tablets, even phones! In today's society, we use computers in almost all aspects of life. Computers have developed and improved dramatically over time. For example, in the 1800's computers were mechanical and had to be powered by a hand crank. They could often only perform one task, such as a simple mathematical calculation.
Just over 40 years ago, when your grandparents and parents were children, almost no one had a computer at home!
(See: http://io9.gizmodo.com/the-history-of-early-computing-machines-from-ancient-t-549202742 for more information).

Modern computers need both hardware and software to work efficiently.
Hardware is the physical parts of a computer that make it work.
Software is the digital programs which allow the computer to do many different jobs.
To use computers, we have to be able to put information, or data, into them, and get results out. We call what goes in "input" and what comes out "output".
Input could be via a keyboard and mouse, a camera or camera phone, or even a microphone.
Output could be a display on a screen, a sound or music through speakers, or something printed out using a printer.
Computer programmers and coders are the people who write the software or 'instructions' which tell a computer how to do different tasks.
<u>Interesting facts to share:</u>
Most early computers could do useful jobs, but could not be programmed.
Charles Babbage was the first person to design a truly programmable computer. From the 1820's, Babbage worked on designs that could do calculations. Using lots of cogs, wheels and rods, they could store numbers, do sums, and give results!
Babbage worked with a mathematician called Ada Lovelace. She helped work out how to give Babbage's machine instructions, using cards with patterns of holes punched in them. Ada is known as the first ever programmer!



To do a job, computers follow a set of instructions, called a computer program. The instructions in a program are known as “code”, and writing computer programs is called “coding”.

“Processing” is what the computer does to the information that goes in, making decisions and organising things, depending on the program, running.

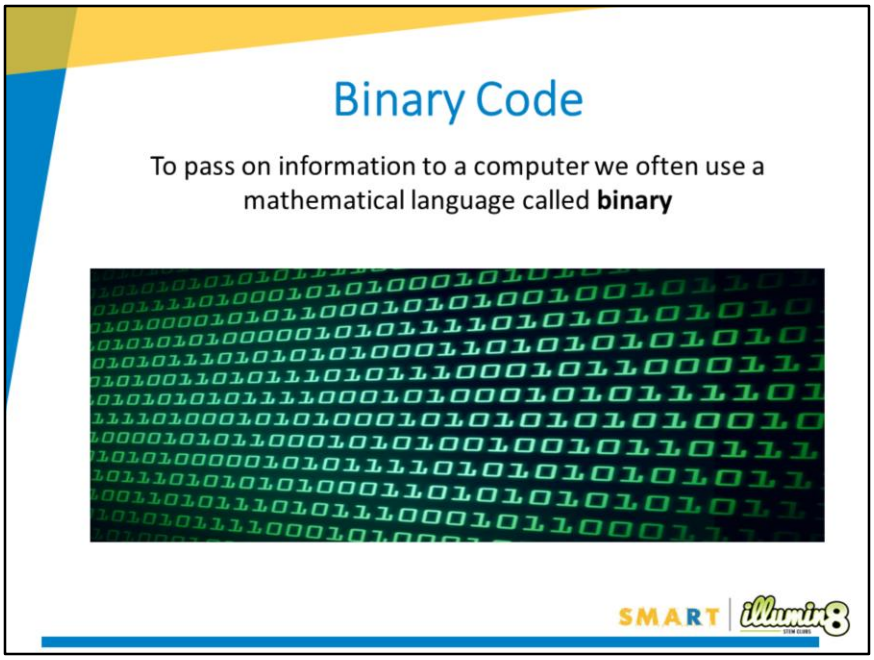
A computer program has to tell the computer exactly how to do every single step of a job!

When modern computers were first invented, users had to write their own programs for each task they wanted their computer to do. E.g. If someone wanted to write a story on the computer, they would first have had to write a program to store and display each letter they pressed on the keyboard!

Now, most computers come with prewritten programs like Microsoft Word, a document editing program which tells the computer how to process our input (typing on the keyboard for example).

This saves us lots of time!

Computer programs can appear and be written in many ways. Sometimes in text, in numbers, or in pictures. Different ways to program are called **coding languages.**



Computers don't think in words and sentences like we do. Instead, they think in binary numbers as patterns of ones and zero's.

Binary is a way of encoding numbers and letters.

Binary uses only zeros and ones.

Binary is used because computer hardware circuits have two possible states: on, or off.

1 indicates an electric current is on. 0 indicates an electric current is off.

When we communicate with each other in numbers, we use all 10 numbers of the decimal system. 1,2,3,4,5,6,7,8,9,0

So why can't we just code using the decimal system? The decimal system wouldn't work in computers, because we would need a switch with 10 different possibilities. Whereas in reality we only have two; on or off.

That's why binary, only using two digits, is so useful!

Binary 'Computer Talk': Code!

Each **b**inary dig**it**, a zero or one, is called a **bit**.

A combination of 8 bits together, are called a **byte**.
For example: 0101 1100 is a byte!

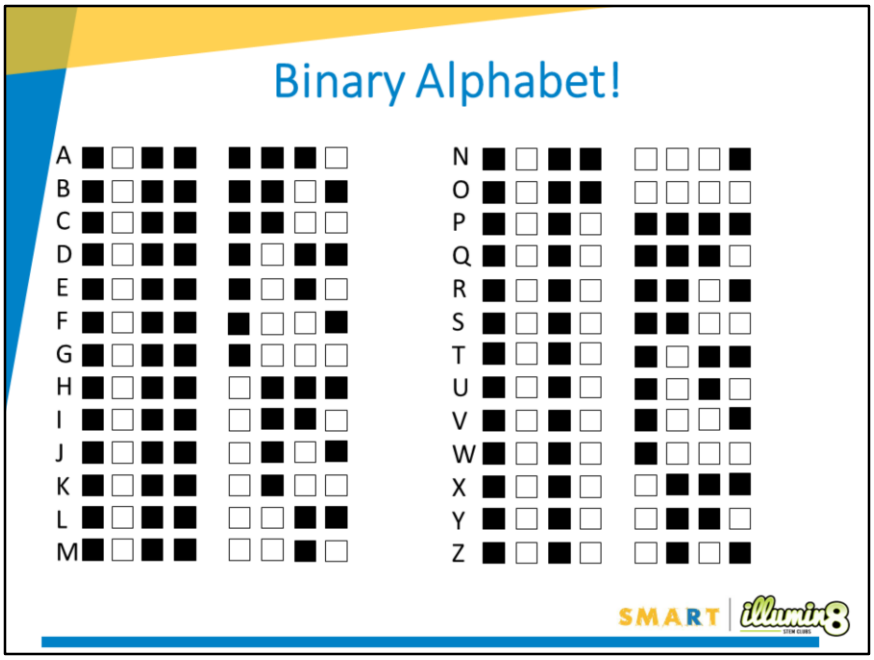
Let's look at how these zeros and ones can be used to make letters!

All letters can be represented by a combination of 8 bits!

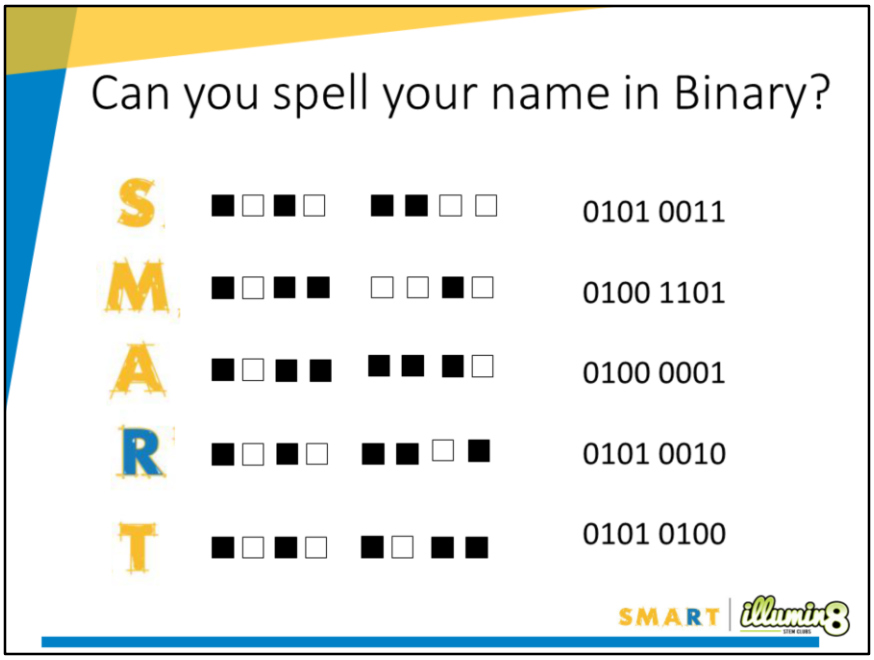
Image Source: https://www.thinglink.com/scene/482541017224970242

SMART | illuminate

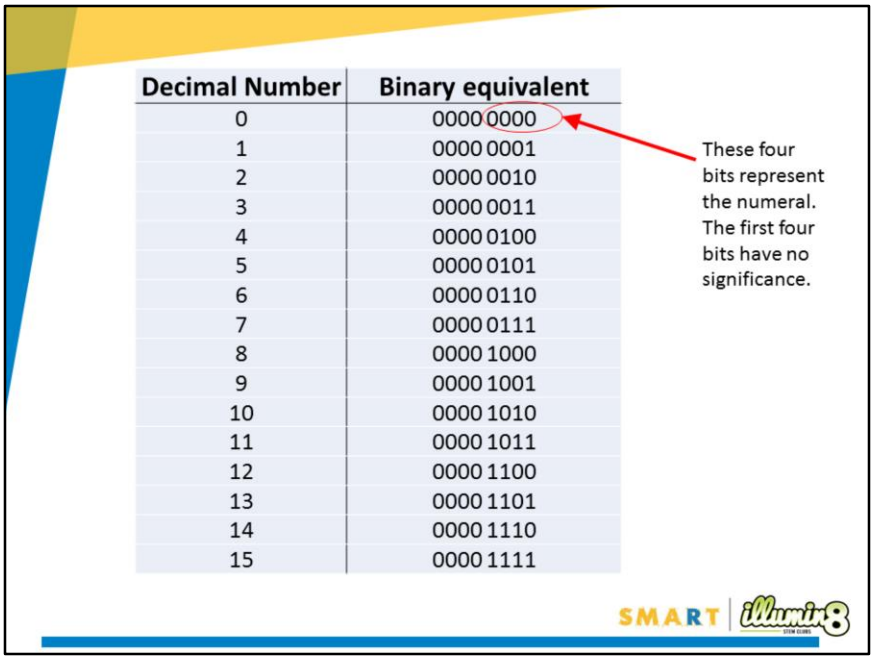
Binary Code is just that – a special code for expressing letters in a different form. We can use the binary code system to write words, instructions – even our names! Let's have a look a the binary alphabet and see what we can write!



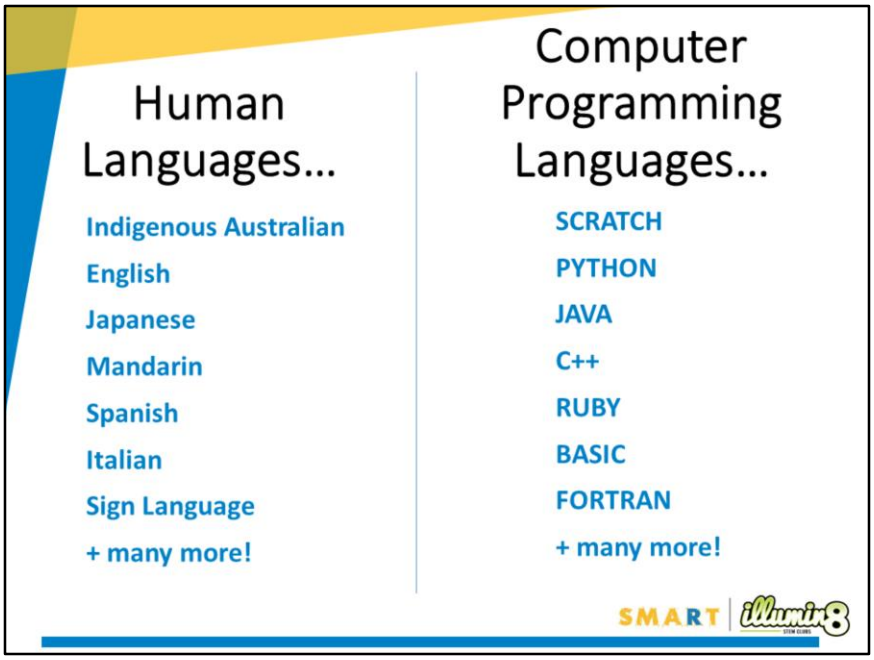
Each letter is coded by one byte.
In this image the blacks squares represent zero's and the white squares represent one's.
This is how computers represent and code letters!
We can use this system to code words – and even write our names!



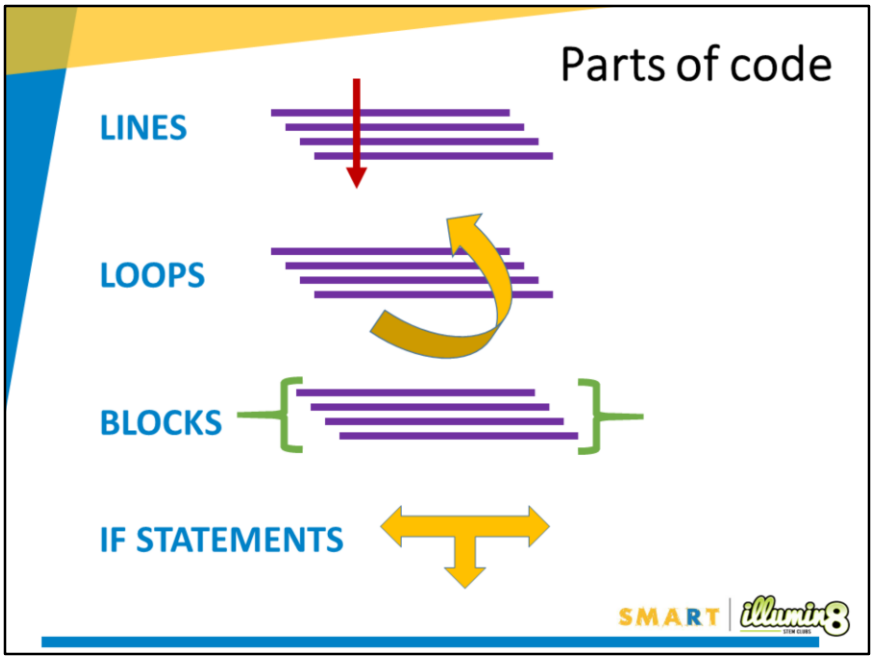
In this image the blacks squares represent zero's and the white squares represent one's.
Let's do an activity together to practice binary. Try writing your first name out in binary code.
What other words can you write in binary?
Could you write a simple instruction for a computer, perhaps: "draw"
……..
Refer to Coordinator notes for Activity 6.1.2 and 6.1.3. Use the provided worksheet in the coordinator notes for students to write out their names in binary. Extend the activity to making binary necklaces if student interest level supports this extension.
Additional binary code activities can be found online at:
http://csunplugged.org/wp-content/uploads/2014/12/unplugged-01-binary_numbers.pdf



Decimal numbers may also be written in binary form. Most computers deal with data in 8-bit bytes.
Writing decimal numbers like this in an 8-bit byte, uses a method called ‘unpacked’. Each numeral is encoded into one byte, with four bits representing the numeral and the remaining bits having no significance.



It would be really hard to write computer programs in binary code – the programs would be very long!
Instead, we use computer programming languages, which are easier to work with. There are lots of them! They make it easier for humans to write programs. However, the computer's processing system can't understand them – so, there is always a translator program in the middle, which converts our programming language into binary code, so the computer can understand.
Humans speak and write in lots of different languages across the world.
Can you think of any other languages, other than English?
Have you heard of any computer programming languages?



LINES

Programs are writing in lines. The computer starts at the top and reads the first line, then the next line, then the next line, and so on – just like reading a book.

LOOPS

Sometimes, you need to do something several times. Instead of writing out the same line of code over and over again, you tell the computer to repeat the instructions. This is called a loop.

BLOCKS

A block is a section of code that does a particular job. Blocks are able to be reused in different programs, and different parts of a program. For example, you may write a block to give an instruction that you want to happen at the end of all of your programs, like “when program ends, make a beeping sound for 1 second”. If you create this as a block, you can copy it each time, rather than recoding it each time.

IF STATEMENTS

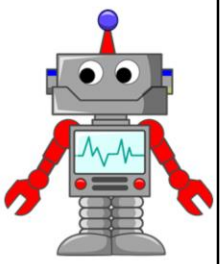
An “if statement” tells the computer to do something only if something else is true.

Writing computer programs is called 'coding' and the written down programs are called 'code'.
Whatever job your computer is doing, and whatever software or app you are using, computer code is making it work! When you tell your computer to use your code, it's called "running" the code.
Programmers write programs as lists of instructions. They need to be very specific!
Let's have a look at the code we would need to write to have a group member pick up a pencil!
Would the code on this slide work? Lets test it out!
Note: Don't ask the students to move, or find a pencil. Just ask them to follow the instructions. Read the instructions out, one line / step at a time.
How did everyone go – who was able to pick up a pencil using this code?
What might we need to change, add or ensure to make it better? For example, we are assuming the pencil is nearby, and within reach. We're also assuming the group member knows what a pencil is! What if there were also a pen on the table? Maybe we could add an 'if statement' to our code, such as, "IF there is a pencil within reach, run the program".
We need to write lots of simple steps in our codes, to ensure the actions are undertaken as we intend.
We can run our code to test it out.
We can then check for errors, and make changes, until the program runs as we have

intended.

Video:
Josh Gaines and his two children, working on exact instructions to make a peanut butter and jelly sandwich.
https://www.youtube.com/watch?v=cDA3_5982h8

'Program Your Robot' Challenge!



In this week's challenge you will need to write a program to get your "ROBOT" through a maze!

Your "ROBOT" will be one of your group members – take it in turns!

Make sure you give clear instructions!

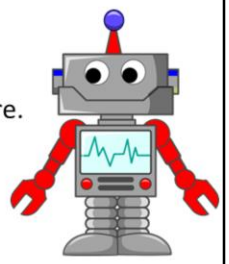
Remember computers inside robots don't 'know' anything – they can only do what they have been told by your program!

Image Source: https://pixabay.com

SMART | illumin8

The Challenge

Computer code can be written in lots of different languages like HTML, Python, Java Script and more.

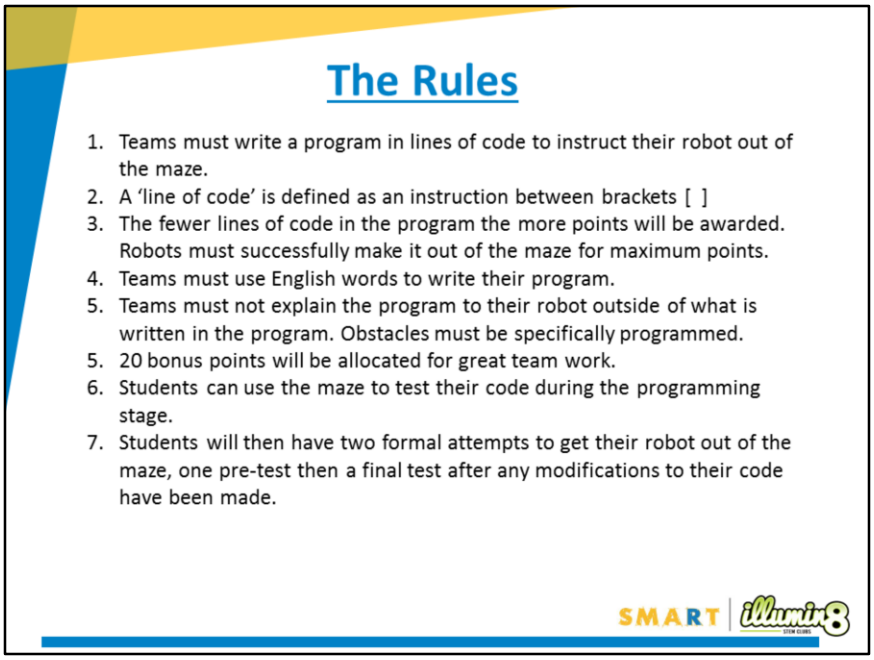


Today we are going to use the English language to write a program to get our ROBOT out of the maze.

The aim of the game if to get the ROBOT out with as few instructions as possible. It's harder than it looks!

SMART | illumin8

We don't want to tell our robots to walk forever, so we can tell them how many steps to take, by writing how many times we'd like them to repeat the program. For example, repeat 10 times, will result in the robot taking 10 steps with their left and right feet in total (20 steps).
We can add in some IF STATEMENTS to tell our robots when to stop, and when to turn.
For example, IF the path is clear, repeat program 10 times. IF the path is blocked, turn to the right.
Have a go!



Use slides 13 and 16 as guides for writing code.

Scoring

Line of code*	-5 per line
Obstacles hit	-10 per hit
Robot makes it to halfway point	+150
Robot escapes complete maze	+250
Teamwork Bonus	+20

*Note: One line of code is an instruction/ set of instructions contained within [ ]
e.g. REPEAT x10:
[Lift left leg >Move left foot forwards>Place left foot down>Lift right leg >Move right foot forwards >Place right foot down]

SMART | illumin8

References

CSUnplugged.org

CS Education Research Group at the University of Canterbury, NZ, Creative Commons BY-NC-SA 4.0 licence.

Code.org

Other References:

- http://csunplugged.org/wp-content/uploads/2014/12/unplugged-01-binary_numbers.pdf
- http://www.explainthatstuff.com/howcomputerswork.html
- https://www.youtube.com/watch?v=1sWCBgGALXE
- http://nookkin.com/articles/computer-science/why-computers-use-binary.ndoc
- https://studio.code.org/unplugged/unplug1.pdf
- https://www.mathsisfun.com/binary-number-system.html
- https://www.youtube.com/channel/UCI8AQqkapE9K6pd8aK71B7w

SMART | illumin8