

# Binary Numbers

Computers today use **digits** to represent information - that's why they're called **digital systems**. The simplest and most common way to represent digits is the binary number system, with just two digits (usually written as 0 and 1). It is called binary because there are only two different digits used, or two states.

## What is it?

The binary number system plays a central role in how information of all kinds is stored on computers. Understanding binary representation can lift a lot of the mystery from computers, because at a fundamental level they're really just machines for flipping binary digits on and off. Computers are simple machines, and they need very exact instructions to make them do complex tasks.

## Why?

Teaching binary numbers as an introduction to computational thinking introduces students to algorithms and decomposition, as they learn to break down the problems of calculating binary numbers and converting between binary and decimal numbers, into step by step processes that they can follow to solve these problems; it also introduces abstraction, as students learn that two different things can be used to represent any and all information. It shows them that a computer isn't actually that complex, and we use simple concepts in a clever way to make computers do extraordinary things!

## Link to Digital Technologies Curriculum

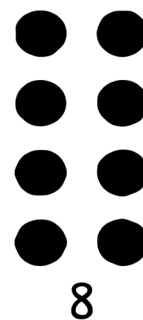
This activity is a good first-step for the first Progress Outcome in Computational Thinking. We used an algorithm in this lesson to convert a decimal number to a binary one. This is an algorithm because it is a step-by-step process that will always give the right solution for any input you give it as long as the process is followed exactly.

*This activity utilises the CS Unplugged material and is released under Creative Commons CC BY-SA 4.0.*

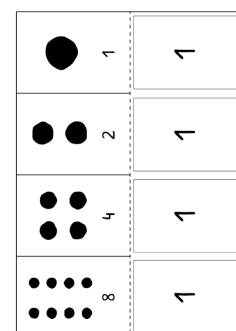
**Mathematics:** Numeracy  
**Literacy:** Speaking  
**Performing Arts:** Drama

## Downloadable Resources

(One Per Class):



## Binary Cards



## Binary Windows

**Note:** Be sure to print double-sided!

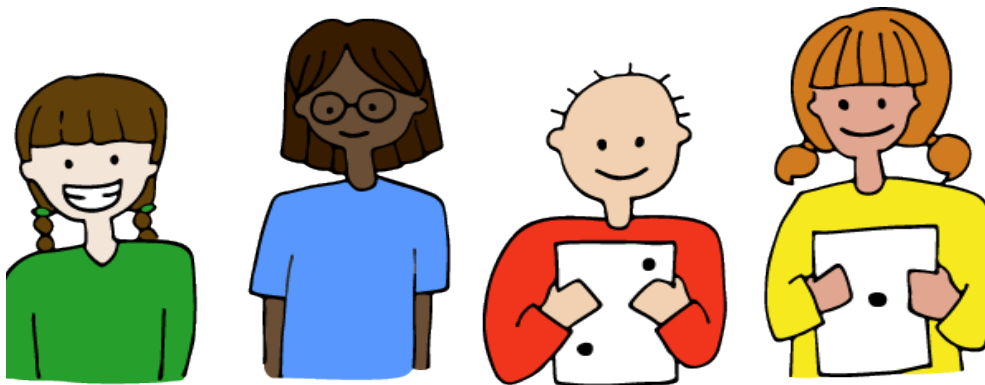
## Activity Background

Did you know that right in the inside of any computer there are billions of little tiny things that can be turned on or off, like a light switch, and that when you have lots of these things together they can display a number or a letter or a movie or make up your favourite game on your device? So let's look at how they work. Now we need to pretend that we are so incredibly small that we are now inside a computer and we are making the computer show a number. Are you ready?

First of all here's a card that is the tiny thing that can be turned on or off. The technical word for that is a "bit".

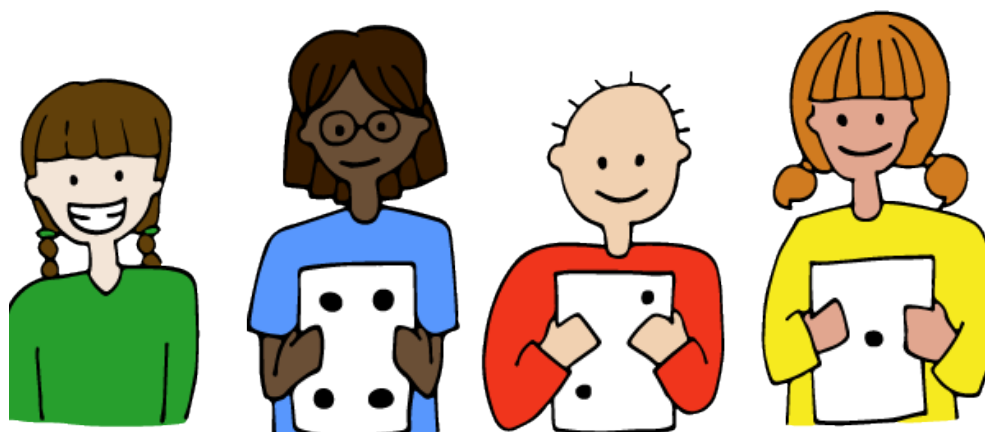
Hold the 4 cards (1, 2, 4, and 8 dots), but don't let students see the dots. Ask for 4 students to volunteer to be "bits", and have them stand in a line in front of the class.

Hand out the 1-dot card to the person on the right. Explain that they are one "bit" (binary digit), and can be on or off, black or white, 0 or 1 dots. The only rule is that their card is either completely visible, or not visible (i.e. flipped over). Hand out the second card so that the first card is on the far right. Point out that this card has either 2 dots, or none.



Ask the class what the number of dots on the next card will be. Get them to explain why they think that.

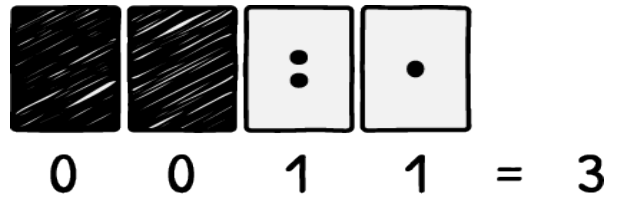
Silently give out the 4-dot card, and let them try to see the pattern. This will be dependent on their level of number knowledge. Mention that each number is doubling (or that if you had two of a card, that would give you the card beside it), and move onto the next step.



# The Activity

Tell the students that the rule is that a card either has the dots showing, or hiding. If we can turn cards on (showing) and off (hiding) by showing the front and back of the card, how would we show 3 dots? Begin by asking: how many dots are on the left-most card? Count together that there are 8 dots. Let's look at the number line. Is 8 bigger than 3? Let's hide it because it's too big. Now let's look at the next card, how many dots can we see? Let's count them. There are 4, is 4 bigger than 3? Yes, so we need to hide the card. If we didn't hide the 4 card what would happen? (There would be too many dots). How many dots are on the next card? Let's count them. There are 2 dots. Let's look at the number 3, (show it with materials and show that 2 fits into 3, with how many left over?) We need one more dot to make the number 3. So we will leave that card visible.

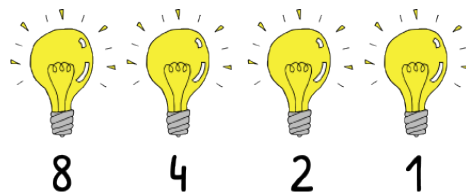
Without being given any rules other than each card being visible or not, students will usually come up with the following representation.



Now ask "How would you make the number 6?" (Again, start by asking if they want the 8 card, and so on from left to right).

The process we have been following to make these numbers is an algorithm, which converts our normal counting numbers to a binary representation. Together let's think through the steps we followed to do this. We'll do this by using the earlier example of representing the number 3 again.

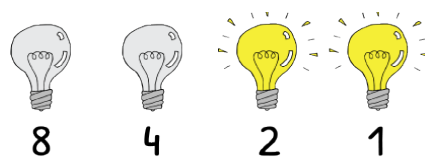
a. Start with all the numbers switched to on (all the dots visible). You could start to introduce abstract representations, like varying the description of the cards (visible/hidden; dots/no-dots; white/black if the backing is black; yes/no; or on/off)



b. Does 8 fit into 3? No - so turn it off



c. Does 4 fit into 3? No - so turn it off



d. Does 2 fit into 3? Yes - so keep it on. How many are left over? (1)

e. Does 1 fit into 1? Yes - so keep it on. How many are left over? (none)

## Extending The Lesson

- Group students into pairs
- Give each pair a set of the smaller binary cards
- Have them practise the algorithm for numbers below 10.

1. Explain to students that we're working with just two digits, so they are called binary digits. (You could explore the meaning of the "bi" prefix with words like bicycle, biennial, bilingual and bicultural.) Binary digits are so common that we have a short name for them: write "binary digit" on a piece of paper, then rip off the "bi" at the start, and the "t" at the end, put it together and ask what the combined word ("bit") spells. This is the short name for a binary digit, which is why we've been referring to the cards as bits; the 4 cards that they have are actually 4 bits.

2. Now let's count from the smallest number that we can make, up to the highest number. What is the smallest number? (they may suggest 1, then realise that it's actually 0).

3. Get the number 0 displayed on the cards (i.e. no dots showing)

4. Now count up 1, 2, 3, 4 ... (each pair should work out these numbers between them)

5. Once they start to get into a routine, ask: how often are we seeing the 1-dot card? (every second time, every odd number)

a) What other patterns are we seeing? (the 2-dot card flips on every second count, the 4-dot on every 4th and so on; the 8 dot card doesn't do much; this may be challenging for some students to recognise, and the main goal is that they are aware that the 1-dot card flips every time, and that every second number is an odd number).

6. Continue until all the cards are switched to "on" and have counted to 15. What happens next? (We have to add a new card.) How many dots on it? (16) What do we have to do to the other 4 cards when we get to 16? (we have to turn them all off)



### Extension ideas

- A. So when I have two bits I can make a maximum of? (3)
- B. I add another bit and that has how many dots on it? (4)
- C. I turn off the first two bits to make 4 right?
- D. Now let's turn on all three bits, so now we have how many? (7)

- E. I add another bit and that has how many dots on it? (8)
- F. Repeat until a pattern is recognised that the number on the next card is one more than the total number of dots on all the cards to the right (e.g. there are 7 dots on the 4, 2 and 1 cards, so the next card to the left is 8). This makes it easy to work out the number if all the bits are switched on - double the left-hand card, and subtract 1.
- G. How many different numbers can I make with two bits? (4; often students will say 3 because they haven't counted 0).
- H. Let's add the next bit; how many different numbers can we make now? (8, again 7 will often be given as an answer first).
- I. Repeat until a pattern is recognised that each time we add another bit we can now represent twice as many numbers.

