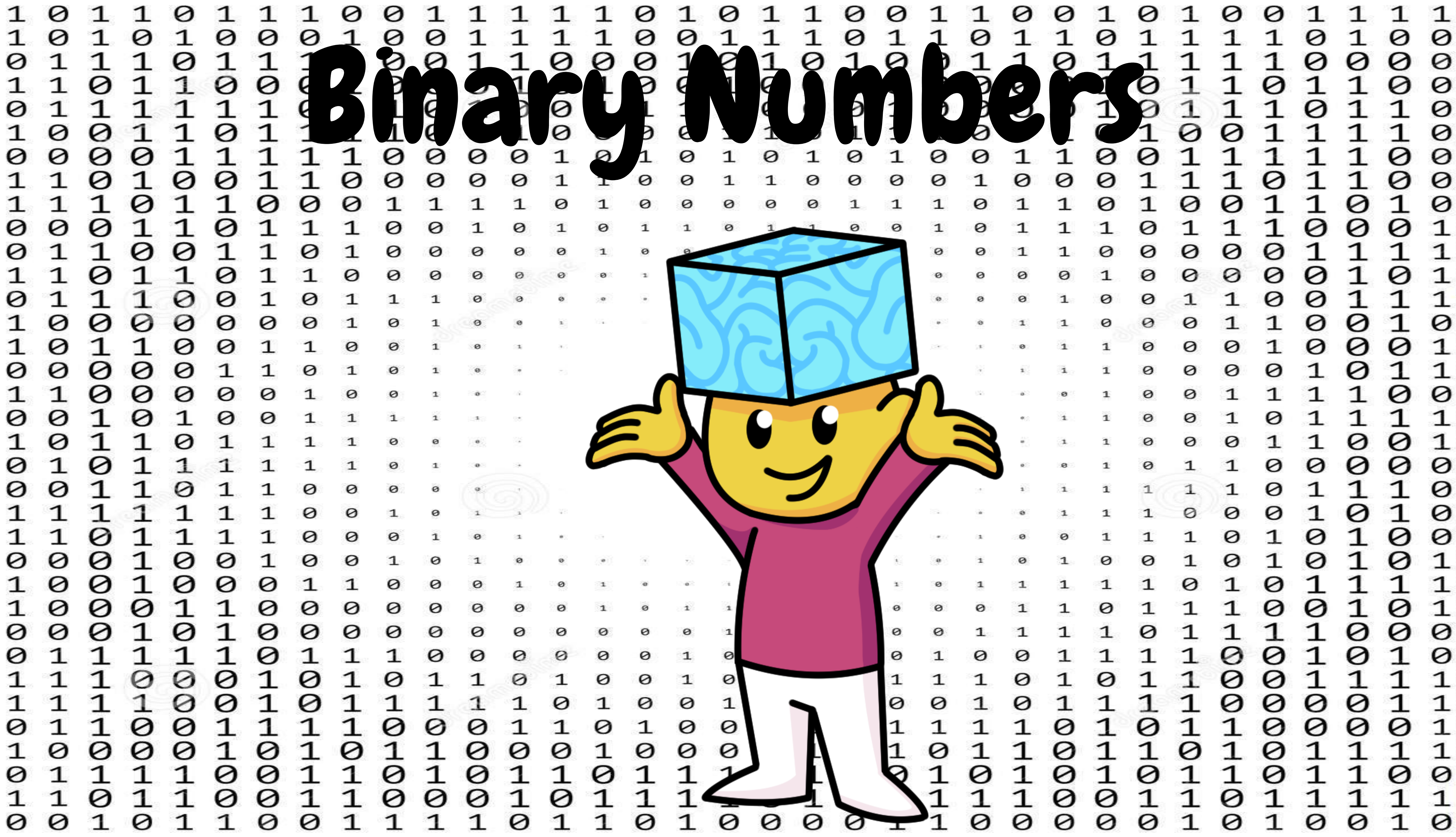
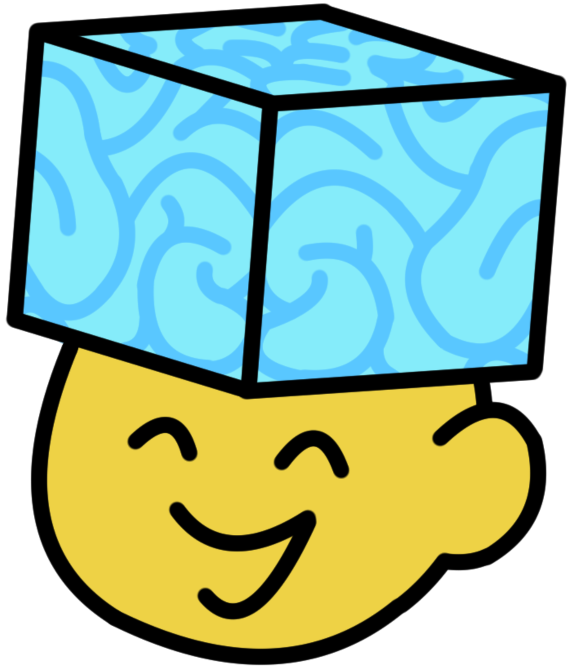
Binary Numbers



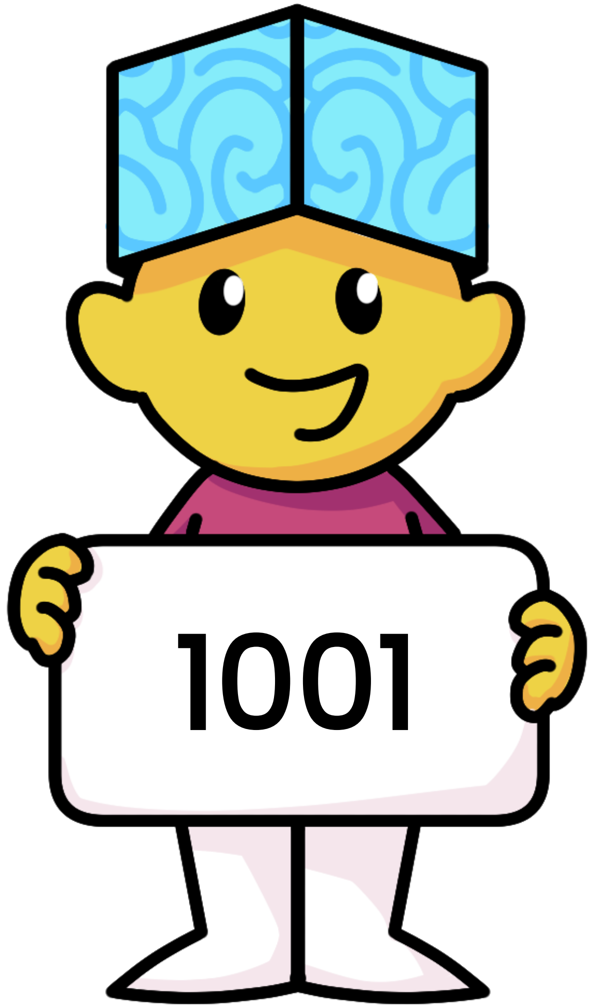
Why is this Joke Funny?

“There are 10 types of people. Those you know binary and those who don’t!”



The joke is funny because it uses the difference between binary and decimal number systems to “confuse us”

What is Binary?



Computers use the binary number system.

A **base-2** number system that uses two digits (0 and 1) in place values to get any numeric value.

How Base-2 Works

2^3	2^2	2^1	2^0
Eights	Fours	Twos	Ones
1	0	0	1

$$= (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$
$$= (1 \times 8) + (0 \times 4) + (0 \times 2) + (1 \times 1)$$
$$= 8 + 0 + 0 + 1$$
$$= \boxed{9}$$

This is how a base-2 system is used to write the value 9.

How Base-2 Works

2^3	2^2	2^1	2^0	$\longleftarrow$ Notation Value
Eights	Fours	Twos	Ones	$\longleftarrow$ Place Value
1	0	0	1	$\longleftarrow$ Numerical Value in Base 2 = 1001
8 + 0 + 0 + 1				= 9 $\longleftarrow$ Expanded Value

What is Binary?

In our daily lives we use the decimal number system also known as base-10.

In a **base-10** number system ten digits are used (0,1,2,3,4,5,6,7,8,9) in place values to get any numerical value.

How Base-10 Works

10^3	10^2	10^1	10^0
Thousands	Hundreds	Tens	Ones
1	4	0	7

$$= (1 \times 10^3) + (4 \times 10^2) + (0 \times 10^1) + (7 \times 10^0)$$

$$= (1 \times 1000) + (4 \times 100) + (0 \times 10) + (7 \times 1)$$

$$= 1000 + 400 + 0 + 7$$

$$= \boxed{1407}$$

This is how a base-10 system is used to write the value 1407.

Decimal to Binary

Decimal

$$87 \longrightarrow \underbrace{64 + 16 + 4 + 2 + 1}$$

The decimal number can be broken down into the sum of its parts. It was broken down this way so that it can be converted into binary.

Decimal to Binary

2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1
1	0	1	0	1	1	1

= **Decimal** 87

Each 1 represents the value above it. Each 0 means the value above it is not in use.

Binary 1010111

Binary to Decimal

Binary

110011

2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1
	1	1	0	0	1	1

The sum of the parts are 32+16+0+0+2+1

Binary to Decimal

Binary

110011

32 + 16 + 0 + 0 + 2 + 1

Equals →

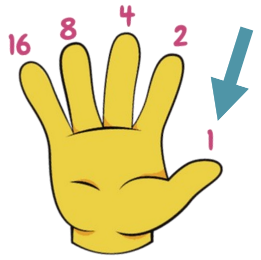
Decimal

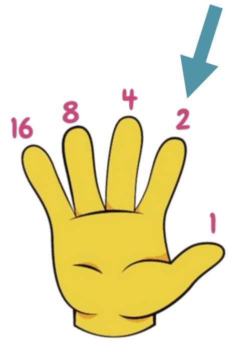
51

Counting in Binary

Did you know that you could use your five fingers on one hand to count up to **31**?! Assign a number value to each of your fingers and count in binary!

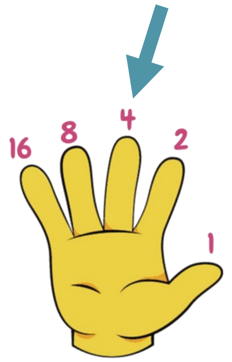
Hold your right hand, palm-side up and assign the number 1 to your thumb.

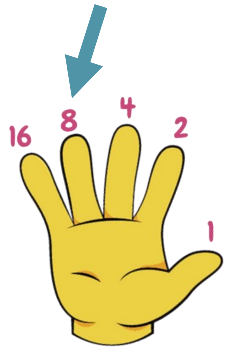




Double that number and assign the number 2 to your index finger.

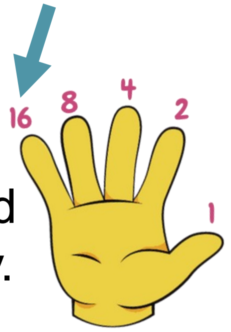
Double that and assign the number 4 to your middle finger.



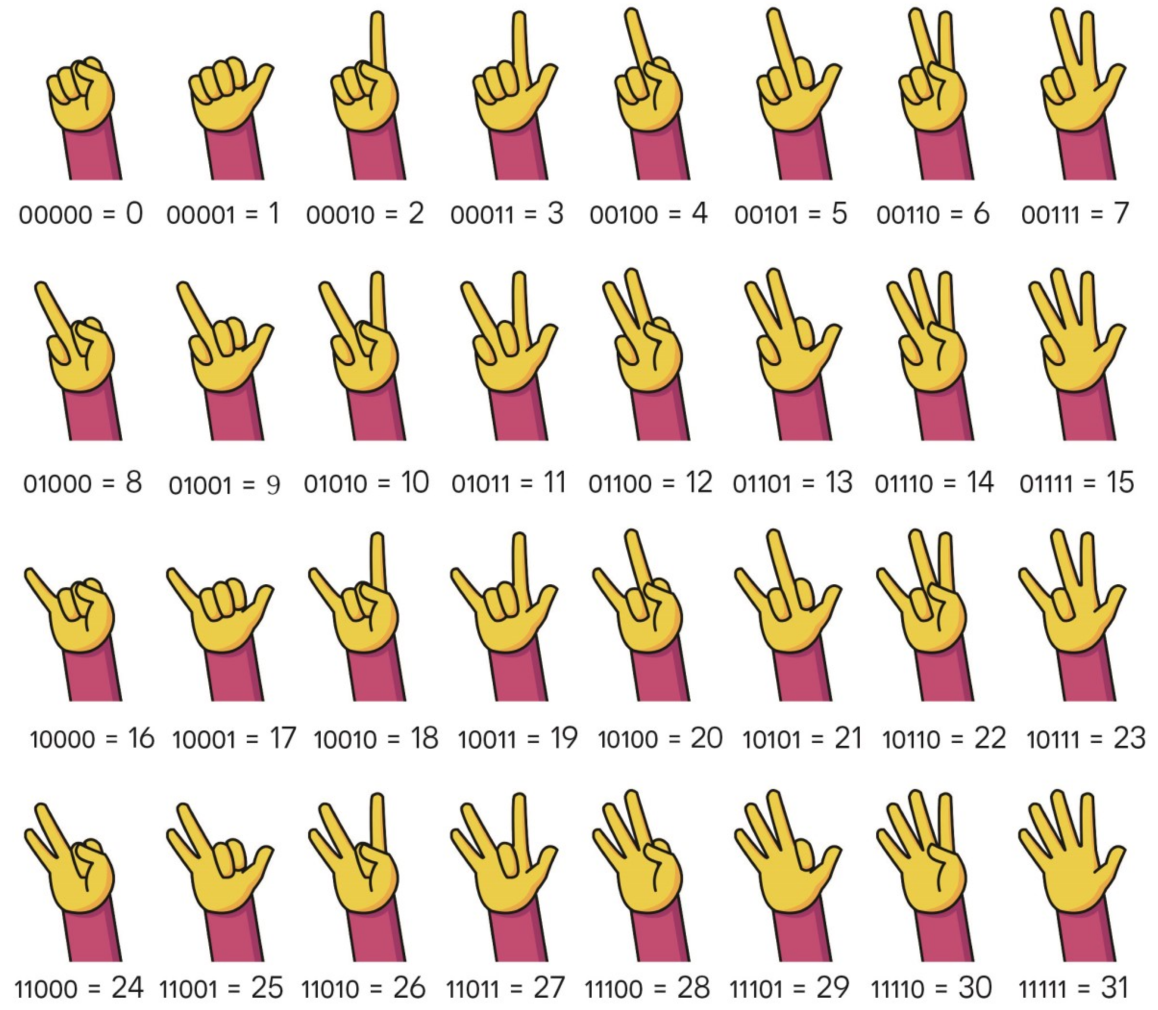


Double that and assign 8 to your ring finger.

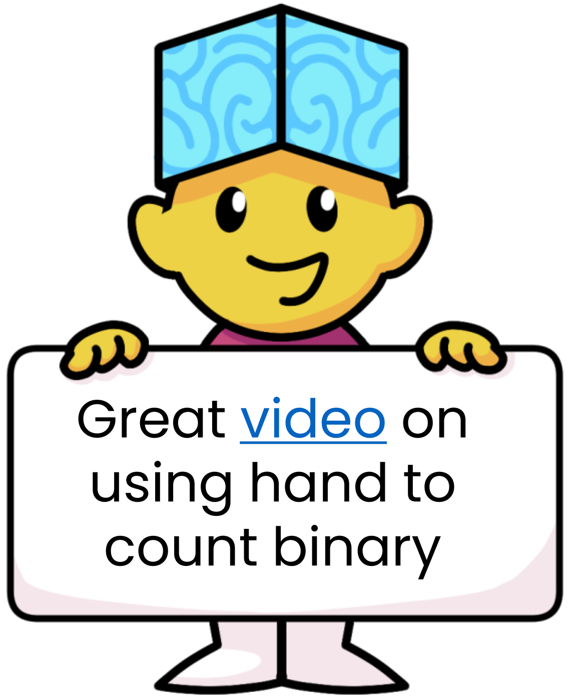
Finally, double that, and assign 16 to your pinky.



Counting in Binary

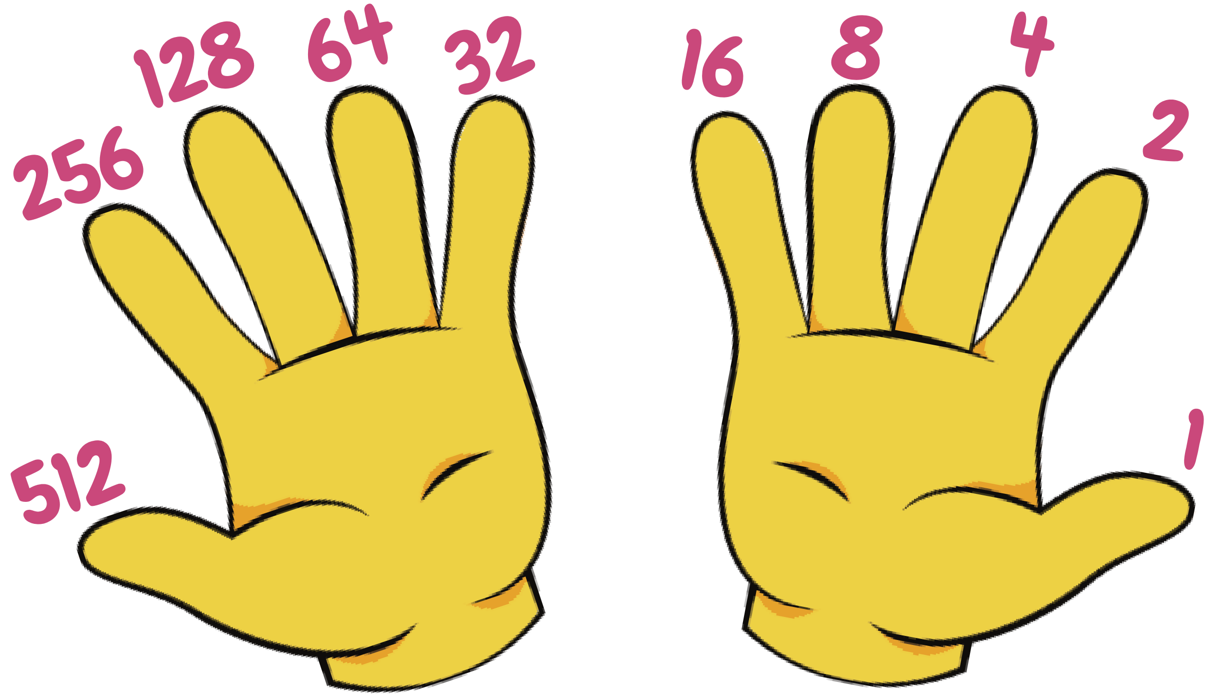


Let's look at the hand signals for each all the numbers 1-31!

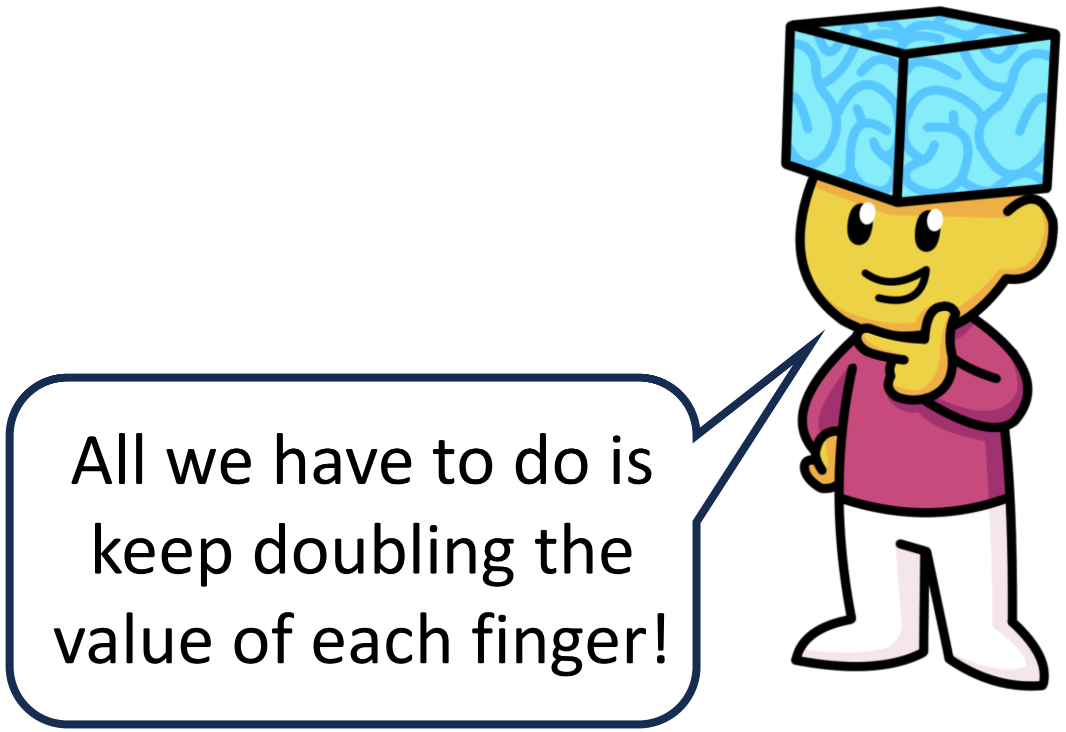


Great video on using hand to count binary

Counting in Binary



Did you know that you could use your 10 fingers on both hands to count up to 1023?



All we have to do is keep doubling the value of each finger!

Base Notation

When looking at numbers of different bases, we use base notation to clarify what the base is, reducing confusion when converting between bases.

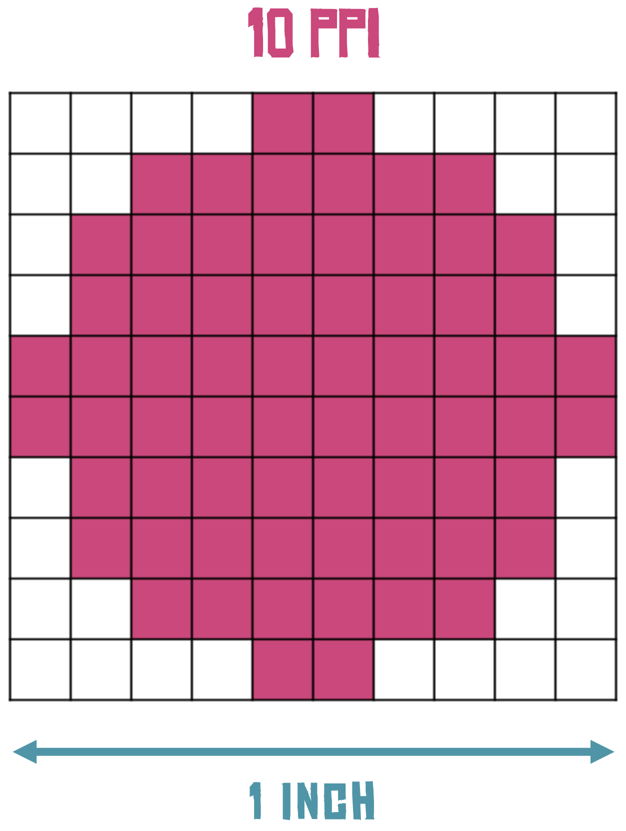
The regular number is the number to be evaluated in the stated base.

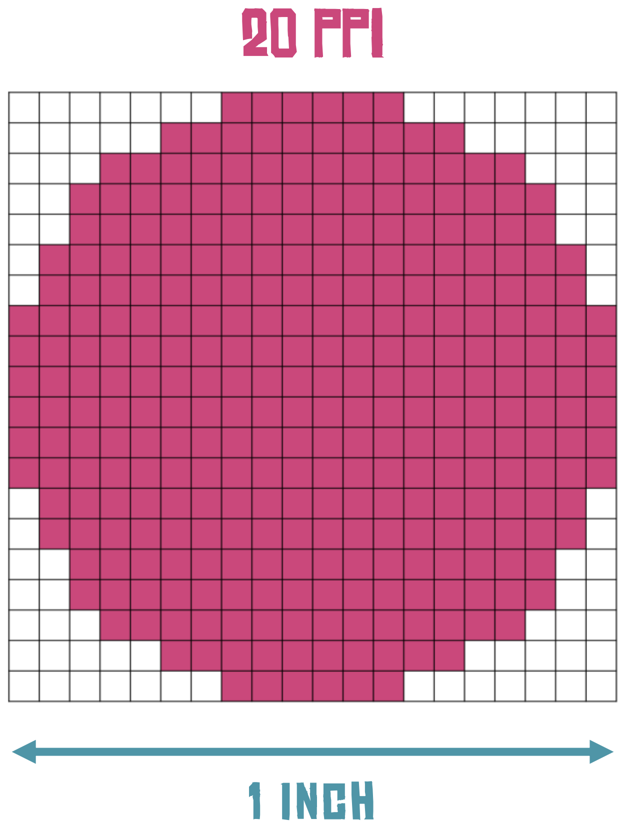
$$10_2$$

We use subscript to notate the base.

Common Bases:
Base-2 → Binary
Base-10 → Decimal
Base-16→ Hexadecimal

Bitmaps

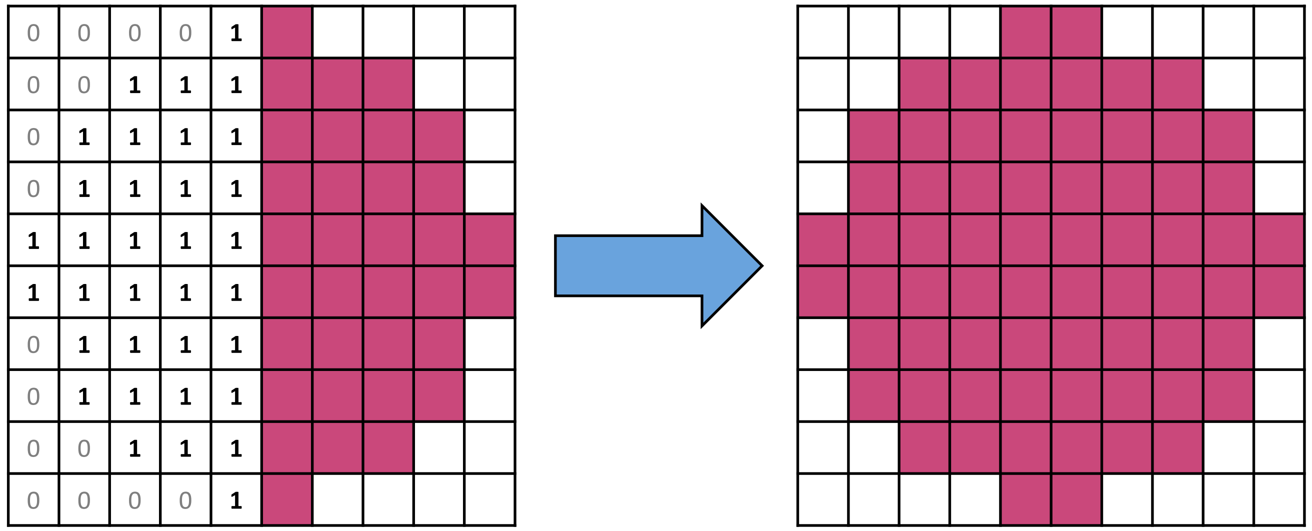




A bitmap is a way to represent and store digital images in a computer. It is a type of format that uses a grid divided into tiny squares called pixels to describe an image. The pixels per inch (PPI) can be adjusted depending on the application.

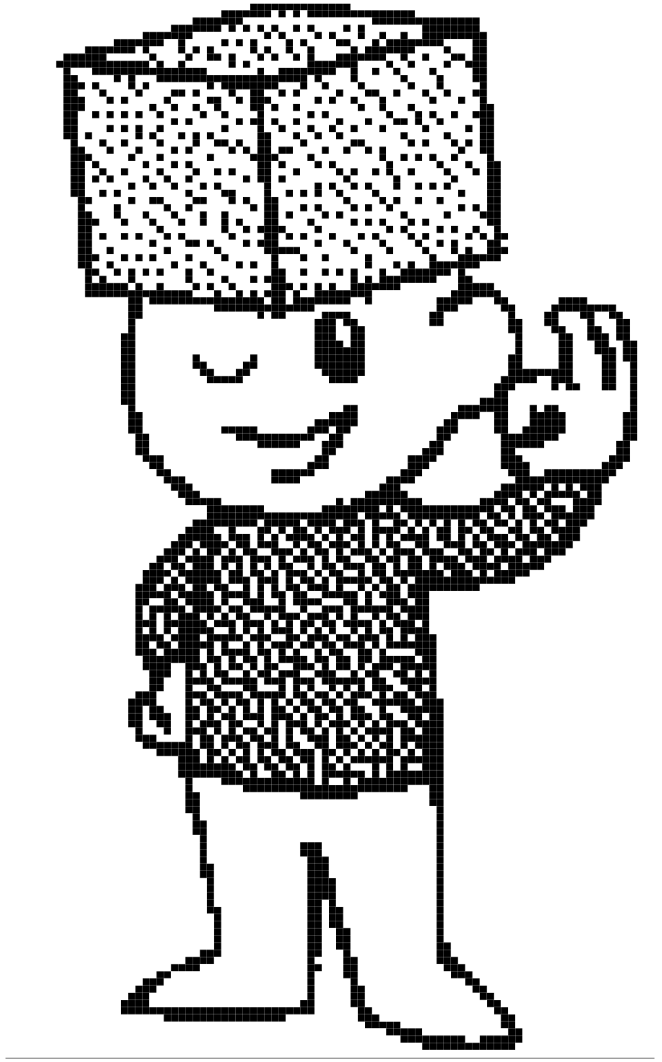
Bitmaps

The simplest digital images are monochrome (black and white) images that use one bit per pixel, "1" for black and "0" for white. The computer stores information about each pixel, such as its position and color, in a specific order.

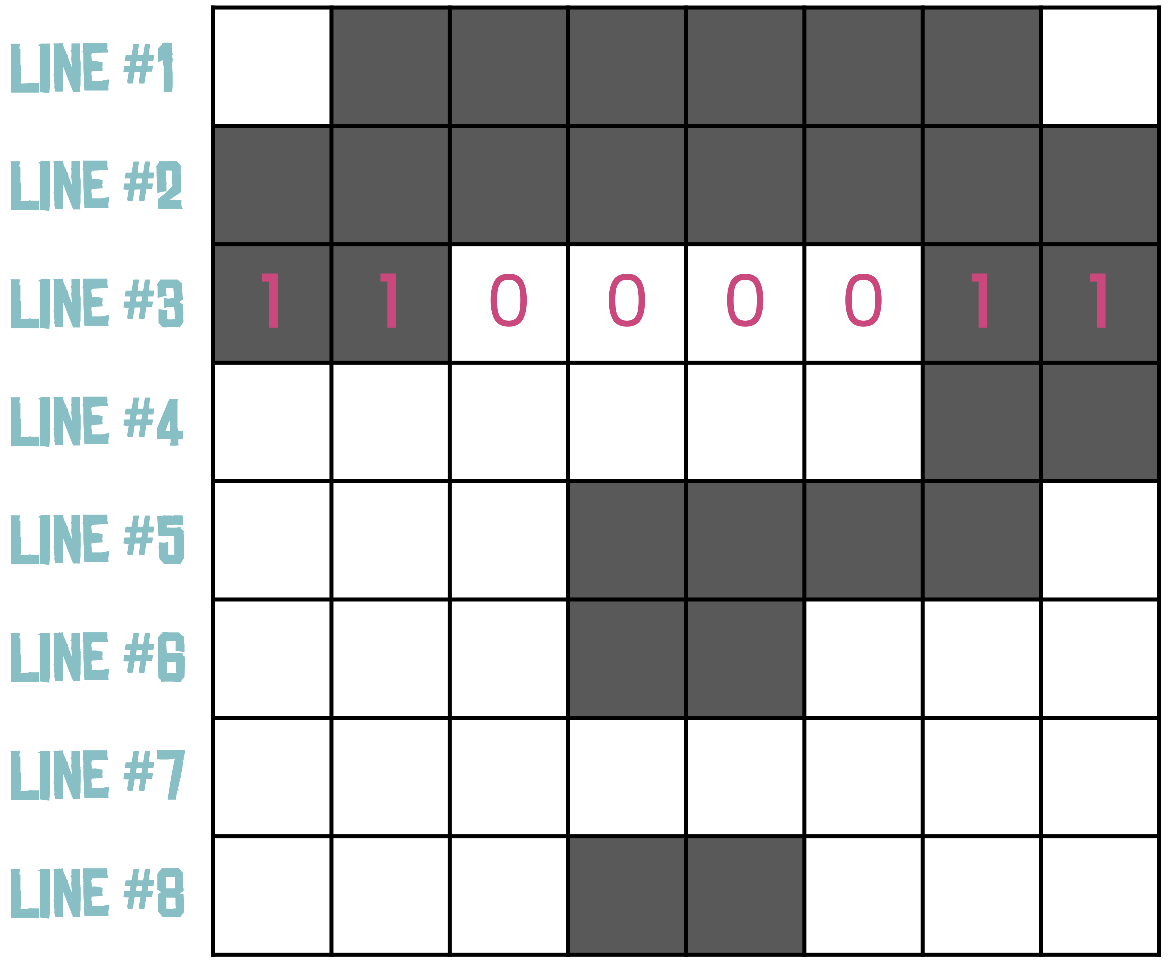


0	0	0	0	1	🟥				
0	0	1	1	1	🟥	🟥	🟥		
0	1	1	1	1	🟥	🟥	🟥	🟥	
0	1	1	1	1	🟥	🟥	🟥	🟥	
1	1	1	1	1	🟥	🟥	🟥	🟥	🟥
1	1	1	1	1	🟥	🟥	🟥	🟥	🟥
0	1	1	1	1	🟥	🟥	🟥	🟥	
0	1	1	1	1	🟥	🟥	🟥	🟥	
0	0	1	1	1	🟥	🟥	🟥		
0	0	0	0	1	🟥				

				🟥	🟥				
		🟥	🟥	🟥	🟥	🟥	🟥		
	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	
	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	
🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	
	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	
		🟥	🟥	🟥	🟥	🟥	🟥		
				🟥	🟥				



Bitmaps



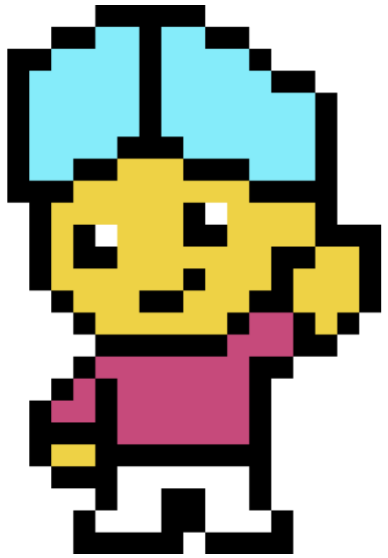
Each row and column of a bitmap can be represented by lines of binary code.

Bitmaps

However, we can make more complex and detailed bitmaps by storing more than one bit inside of each pixel.

We can use 8 bits (1 byte) to achieve 256 different states that range from black to white. This is how we achieve grayscale.

For color, we can use 24 bits and assign 8 bits to red, green and blue to achieve a wide range of colors. There are over 16 million different combinations!



SQUARE BRAIN

Binary Numbers