

Set 0. Bits and Bytes

<p>Skill 0.1: Describe how computers represent data</p> <p>Skill 0.2: Determine the amount of data that can be stored in a series of bits</p> <p>Skill 0.3: Describe how computers interpret bits</p> <p>Skill 0.4: Convert between bit and bytes</p> <p>Skill 0.5: Explain the importance of bytes</p>
--

<p>Skill 0.1: Describe how computers represent data</p> <p>Skill 0.2: Determine the amount of data that can be stored in a series of bits</p> <p>Skill 0.3: Describe how computers interpret bits</p> <p>Skill 0.4: Convert between bit and bytes</p> <p>Skill 0.5: Explain the importance of bytes</p>
--

<p>Skill 0.1: Describe how computers represent data</p> <p>Skill 0.2: Determine the amount of data that can be stored in a series of bits</p> <p>Skill 0.3: Describe how computers interpret bits</p> <p>Skill 0.4: Convert between bit and bytes</p> <p>Skill 0.5: Explain the importance of bytes</p>
--

<p>Skill 0.1: Describe how computers represent data</p> <p>Skill 0.2: Determine the amount of data that can be stored in a series of bits</p> <p>Skill 0.3: Describe how computers interpret bits</p> <p>Skill 0.4: Convert between bit and bytes</p> <p>Skill 0.5: Explain the importance of bytes</p>
--

<p>Skill 0.1: Describe how computers represent data</p> <p>Skill 0.2: Determine the amount of data that can be stored in a series of bits</p> <p>Skill 0.3: Describe how computers interpret bits</p> <p>Skill 0.4: Convert between bit and bytes</p> <p>Skill 0.5: Explain the importance of bytes</p>
--

Skill 0.1: Describe how computers represent data

Skill 0.1 Concepts

When we look at a computer, we see text and images and shapes. To a computer, all of that is just binary data, or 1s and 0s. The following 1s and 0s for example represent a tiny GIF.

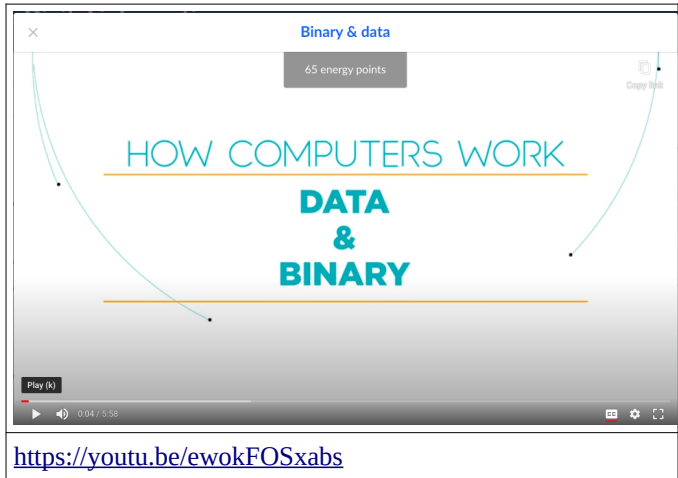
```
01000111010010010100011000111000001110010110000100000001
00000000000000001000000001000000000000000000000000000000
000000000000000011111111111111111110010000111111001
0000010000000001000000000000000000000000000000000000101100
00000000000000000000000000000000000000000000000000000001
00000000000000000000000000000000000000000000000000000001
000000000000000000000000000000000000000000000000000000011011
```

This next string of 1s and 0s represents a command to add a number:

0001 0001 0000 0101

You might be scratching your head at this point. Why do computers represent information in such a hard to read way? And how can 1s and 0s represent so many different things? That's what we'll explore in this lesson.

To start off, check out the next video from Code.org where engineers from Microsoft and Adafruit introduce the basics of bits and binary data.




Computers store information using bits. A **bit** (short for "binary digit") stores either the value 0 or 1.

A single bit can only represent two different values. That's not very much, but that's still enough to represent any two-valued state. The following represent information that could be stored in a single bit,


Is a lightbulb on or off?

State

Bit



0



1

Is a button enabled or disabled?

State

Bit

Submit

0

Submit

1

Is the current time AM or PM?

State

Bit

9:35 AM

0

9:35 PM

1

Skill 0.1 Exercise 1

Skill 0.2: Determine the amount of data that can be stored in a series of bits

Skill 0.2 Concepts

Computers use multiple bits to represent data that is more complex than a simple on/off value.

A sequence of two bits for example can represent four distinct values,

00, 01, 10, 11

A sequence of three bits can represent eight different values,

000, 001, 010, 011, 100, 101, 110, 111

A sequence can represent many things: a number, a character, a pixel. Plus, the same sequence can represent different types of data in different contexts. The sequence 1000011 for example could represent 67 in a calculator application while also representing the letter “C” in a text file.

Skill 0.2 Exercise 1

Skill 0.3: Describe how computers interpret bits

Skill 0.3 Concepts

In a computer, information travels over wires. The easiest way to convey information in a wire is to consider it “on” or “off”, based on how much electricity is going through it.



An “on” wire represents 1, and an “off” wire represents 0.



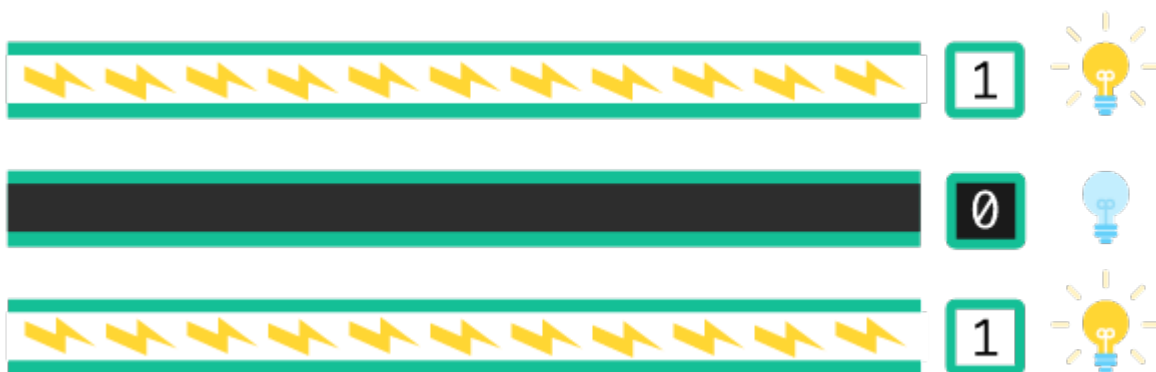
This small piece of information is called a “bit”, and it’s the smallest piece of information that computers process.

A single wire can only represent one bit, one piece of information. We can represent the result of a coin flip with a single bit – by saying that 0 represents tails and 1 represents heads, but we usually need to represent much more information than that in a computer.

The solution??? More wires!

Each wire adds an additional bit of information, an extra bit that can be considered on or off, 1 or 0.

For example, let’s say we want to represent which of three light bulbs to turn on. We can use three wires, with each wire representing the on/off state of a light bulb.



Skill 0.3 Exercise 1

Skill 0.4 Convert between bits and bytes

A **bit** is the smallest piece of information in a computer, a single value storing either 0 or 1.

A **byte** is a unit of digital information that consists of 8 of those bits.

Here's a single byte of information,

11110110

Here are three more bytes of information,

000010100101010011011011

Conversion between bits and bytes is a simple calculation: divide by 8 to convert from bits to bytes or multiply by 8 to convert from bytes to bits.

Skill 0.4 Exercise 1

Skill 0.5 Explain the importance of bytes

What is so special about 8 bits that it deserves its own name?

Computers do process all data as bits, but they prefer to process bits in byte-sized groupings. Or to put it another way: a byte is how much a computer likes to "bite" at once.



The byte is also the smallest addressable unit of memory in most modern computers. A computer with byte-addressable memory can *not* store an individual piece of data that is smaller than a byte.

Memory

Address	Data
1	11110000
2	00001010
3	00100110
4	11000001

What's in a byte?

A byte represents different types of information depending on the context. It might represent a number, a letter, or a program instruction. It might even represent part of an audio recording or a pixel in an image.

We'll explore how computers can use bits and bytes to represent all types of information in upcoming lessons.

Skill 0.5 Exercise 1