Lesson 4.2: Talking Like a Computer

Objectives

In this lesson, students will:

- ❖ Learn and practice translating decimal numbers into binary numbers
- ❖ Learn and practice converting letters into its binary equivalent and vice versa.

Agenda

1. Review	10 mins
2. Converting Decimal to Binary	10 mins
3. How do Computers Understand Words?	20 mins
4. Wrap Up and Reflections	10 mins

Preparation

- ❏ Print the student activity worksheet, one per student.
- ❏ Print the Binary Character Table, one per student pair.

Resources & Links

- ❏ Video about converting decimal to binary: https://tinyurl.com/y364gqfv

1. Review

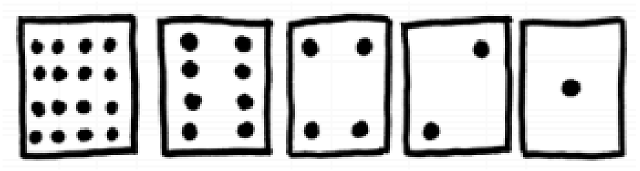
In the last lesson we learned about binary numbers and how to count in binary. We also learned how to convert a binary number into a decimal number.

It would be helpful to do a quick review and call out some binary numbers and prompt students to name the decimal number.

2. Converting Decimal to Binary

We converted binary numbers into decimal numbers. How about converting decimal to binary. How might we do that?

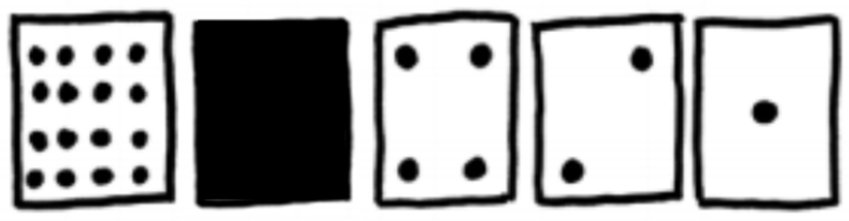
To convert a decimal number, let's say 23, into decimal, we line up our cards again from the highest number of dots to the lowest (from left to right).



We start looking at the cards from the left to right and ask:

1. Does 16 fit into 23? Yes, so we keep the card ON
2. How many dots are left ? 23 - 16 = 7
3. The next card has 8 dots on it. Does 8 fit into 7? No, so the card is OFF.
4. The next card has 4 dots on it. Does 4 fit in 7? Yes, so the card is ON.
5. How many dots are left ? 7-4 = 3 .
6. The next card has 2 dots on it, we keep it ON.
7. We need 3-2 = 1 more dot so we keep the last card ON.

Looking at the cards facing up and facing down, we deduct that 23 (Decimal) = 1 0 1 1 1 (Binary).



Students Turn:

Write a list of decimal numbers on the board and tell students to convert them into binary numbers. Here is a sample list with answers:

13	=	01101
9	=	01001
16	=	10000
3	=	00011

3. How do Computers Understand Words?

Like Scratch, many programming languages look a lot like English, but your computer does not understand English. So how does the computer understand words since we now know everything in the computer is stored in binary?

All instructions given to a computer, no matter the programming language used, are converted to binary.

What about letters? How do computers understand letters? Every letter in the alphabet is represented by a binary number. There is a convention that states that a particular number represents a particular letter. For example, the letter **A** (uppercase) is 01000001 in binary. The computer uses 8 bits to represent each letter. The lowercase letter **a** has a different binary number, namely 01100001. This code for representing english letters as numbers is called ASCII, pronounced '*askey'.*

Student Activity:

Students work in pairs for this activity. Distribute the Binary Character Table, one per student pair. One student goes first and picks a letter of the alphabet and writes it down on a piece of paper and gives it to their partner.

The partner has to translate the letter into its binary equivalent. Then the first student picks another letter, but this time writes down the binary code for the letter. The partner has to find the letter equivalent for the binary code.
Students then switch roles and repeat the same exercise.

Once all students have completed the activity, distribute the student activity worksheet, ***Talking Like a Computer***. In this activity, students follow the directions translating to and from binary to letters.

4. Wrap Up and Reflections

Reflection Points:

- How do computers understand a programming language that looks like english?
- Why do you think there is a different binary code for an uppercase and lowercase letter?
- What strategy did you use to translate the long string of 0s and 1s to letters?
- What is ASCII?

Student Activity Worksheet: Talking Like a Computer

1. Write your first name in binary code

2. What is the computer saying?

01010000010100100100111101000111010100100100000101001
1101010011010100100101001110010001110010000001001001
0101001100100000010001100101010101001110

Binary Character Table

	Binary	Letter	Binary
a	01100001	A	01000001
b	01100010	B	01000010
c	01100011	C	01000011
d	01100100	D	01000100
e	01100101	E	01000101
f	01100110	F	01000110
g	01100111	G	01000111
h	01101000	H	01001000
i	01101001	I	01001001
j	01101010	J	01001010
k	01101011	K	01001011
l	01101100	L	01001100
m	01101101	M	01001101
n	01101110	N	01001110
o	01101111	O	01001111
p	01110000	P	01010000
q	01110001	Q	01010001
r	01110010	R	01010010
s	01110011	S	01010011
t	01110100	T	01010100
u	01110101	U	01010101
v	01110110	V	01010110
w	01110111	W	01010111
x	01111000	X	01011000
y	01111001	Y	01011001
z	01111010	Z	01011010
_	00100000		