Binary Conversions

In this activity, you will practice converting 8-bit RGB (red, green, and blue) color codes from decimal (RRR, GGG, BBB) to binary notation (RRRRRRRRGGGGGGGGBBBBBBBB) using long division. Divide each decimal number in the triplet by 2, the base number for binary notation. The remainder will be an integer less than 2 and will represent a single binary digit or bit (0 or 1). The quotient will become the dividend in the next step, and long division is repeated to find the next binary digit. Repeat until the quotient is zero. Follow this process for each of the three colors (RGB). For more information about decimal and binary notation and their place values, read the "Decimal, Hexadecimal, and Binary Number Systems Overview" sheet.

Color	8-Bit RGB Triplet	
	Decimal Color Code	**Binary Color Code**
White	255, 255, 255	111111111111111111111111
Bright yellow	247, 252, 32	111101111111110000100000
Orange		
Indigo		
Violet		

- In the 8-Bit RGB Triplet chart above, you must find the decimal color code for the given colors using a computer or a table provided by the teacher, then convert the decimal RGB color code to binary notation using long division.
- Review the example below, then practice with the given colors. For the blank rows in the Color column, choose your own colors and fill in the chart accordingly. Be sure to show your work on a separate sheet of paper.

Example for converting an 8-bit RGB triplet from decimal to binary

To determine the binary color code for bright yellow, divide each decimal number (247, 252, 32) by 2. Use a table like the one below to organize the dividend, divisor, quotient, and remainder for each step. Start with the red integer value 247.

1. Divide 247 by 2. Determine the quotient, or how many times 2 divides evenly into 247, and then find the remainder. The remainder is your first binary digit or the digit in the ones (2^0) place. The quotient will be used in the next step.

   247 ÷ 2 = 123 remainder <u>1</u>

2. Divide 123 by 2. Determine how many times 2 divides evenly into 123, and then find the remainder. The remainder is your next binary digit or the digit in the twos (2^1) place. The quotient will be used in the next step.

   123 ÷ 2 = 61 remainder <u>1</u>

3. Divide 61 by 2. Determine how many times 2 divides evenly into 61, and then find the remainder. The remainder is your next binary digit or the digit in the fours (2^2) place. The quotient will be used in the next step.

   61 ÷ 2 = 30 remainder <u>1</u>

4. Divide 30 by 2. Determine how many times 2 divides evenly into 30, and then find the remainder. The remainder is your next binary digit or the digit in the eights (2^3) place. The quotient will be used in the next step.

   30 ÷ 2 = 15 remainder <u>0</u>

5. Divide 15 by 2. Determine how many times 2 divides evenly into 15, and then find the remainder. The remainder is your next binary digit or the digit in the sixteens (2^4) place. The quotient will be used in the next step.

   15 ÷ 2 = 7 remainder 1

6. Divide 7 by 2. Determine how many times 2 divides evenly into 7, and then find the remainder. The remainder is your next binary digit or the digit in the thirty-twos (2^5) place. The quotient will be used in the next step.

   7 ÷ 2 = 3 remainder 1

7. Divide 3 by 2. Determine how many times 2 divides evenly into 3, and then find the remainder. The remainder is your next binary digit or the digit in the sixty-fours (2^6) place. The quotient will be used in the next step.

   3 ÷ 2 = 1 remainder 1

8. Divide 1 by 2. Determine how many times 2 divides evenly into 1, and then find the remainder. The remainder is your next binary digit or the digit in the one hundred twenty-eights (2^7) place.

   1 ÷ 2 = 0 remainder 1

9. Since the quotient is zero, stop dividing. There are eight place values in this binary number. Put the eight binary digits or eight bits together, starting with the first remainder in the ones place (2^0) and working to the left to the last remainder for the one hundred twenty-eights place (2^7), to form the 8-bit binary number or byte.

   11110111

10. Check your work by doing the steps above in reverse. In base 2, the number 11110111 has eight digits. Reading from right to left, the first three digits (111) are in the ones (2^0), twos (2^1), and fours (2^2) places, respectively. The zero is in the eights (2^3) place, and the last four digits (1111) are in the sixteens (2^4), thirty-twos (2^5), sixty-fours (2^6), and one hundred twenty-eights (2^7) places. Multiply each digit by its corresponding place value and find the sum.

    11110111 = (1 × 128) + (1 × 64) + (1 × 32) + (1 × 16) + (0 × 8) + (1 × 4) + (1 × 2) + (1 × 1) = 128 + 64 + 32 + 16 + 0 + 4 + 2 + 1 = 247

Dividend	Divisor	Quotient	Remainder
247	2	123	**1**
123	2	61	**1**
61	2	30	**1**
30	2	15	**0**
15	2	7	**1**
7	2	3	**1**
3	2	1	**1**
1	2	0 STOP	**1**

Now repeat the ten steps above for the green integer value 252.

1. 252 ÷ 2 = 126 remainder 0
2. 126 ÷ 2 = 63 remainder 0
3. 63 ÷ 2 = 31 remainder 1
4. 31 ÷ 2 = 15 remainder 1
5. 15 ÷ 2 = 7 remainder 1
6. 7 ÷ 2 = 3 remainder 1
7. 3 ÷ 2 = 1 remainder 1
8. 1 ÷ 2 = 0 remainder 1
9. **1111100** is the binary number for 252.
10. 11111100 = (1 × 128) + (1 × 64) + (1 × 32) + (1 × 16) + (1 × 8) + (1 × 4) + (0 × 2) + (0 × 1) = 128 + 64 + 32 + 16 + 8 + 4 = 252

Dividend	Divisor	Quotient	Remainder
252	2	126	**0**
126	2	63	**0**
63	2	31	**1**
31	2	15	**1**
15	2	7	**1**
7	2	3	**1**
3	2	1	**1**
1	2	0 STOP	**1**

Now repeat the steps above for the blue integer value 32. There will be fewer than 10 steps since this integer is less than 64 and will need place-holding zeros for the sixty-four (2^6) and one hundred twenty-eight (2^7) places.

1. 32 ÷ 2 = 16 remainder 0
2. 16 ÷ 2 = 8 remainder 0
3. 8 ÷ 2 = 4 remainder 0
4. 4 ÷ 2 = 2 remainder 0
5. 2 ÷ 2 = 1 remainder 0
6. 1 ÷ 2 = 0 remainder 1
7. To generate an 8-bit binary code, the place values for sixty-four (2^6) and one hundred twenty-eight (2^7) would both be zero. Place-holding zeros at the beginning of a binary string do not impact the value of the number. The binary number 100000 is equivalent to the binary number 00100000. **00100000** is the binary number for 32.
8. 00100000 = (0 × 128) + (0 × 64) + (1 × 32) + (0 × 16) + (0 × 8) + (0 × 4) + (0 × 2) + (0 × 1) = 0 + 0 + 32 + 0 + 0 + 0 + 0 + 0 = 32

Dividend	Divisor	Quotient	Remainder
32	2	16	**0**
16	2	8	**0**
8	2	4	**0**
4	2	2	**0**
2	2	1	**0**
1	2	0 STOP	**1**
0	2	0	**0**
0	2	0	**0**

Put all three 8-bit binary numbers together in the same order as the RGB color (247, 252, 32); the binary color code is **111101111111110000100000.**