Decimal to Binary Conversion

15 min

Converting from binary to decimal can get tedious, especially as our binary data grows in length. Lucky for us, converting from decimal to binary is very straightforward.

The method we will use only requires us to be able to divide by 2, hence its name, *Division-by-2 Method*.

Whenever we divide a number we always have two answers:

- the *result* (also known as *quotient*)
- the *remainder*

If our remainder is 0, we normally don't include it when we give the answer but it is still there.

- 35 / 7 = 5 remainder 0
- 36 / 7 = 5 remainder 1

When we divide by 2 we will always end up with a remainder that is either a 1 or a 0, in other words, binary digits.

The Divide-By-2 method will continue dividing a number by two until the result, the first part of the answer, is 0. The first time we divide, the remainder goes in the LSB column. Each time we divide after that, the remainder is written as the next digit to the left.

We can use our even/odd trick we learned to check if we've got the right digit in the LSB. Odds will always be 1's and evens will be 0's.

Follow along with the conversion of 27_{10} to binary:

Dividend$_{10}$	**Divisor$_{10}$**	**Result$_{10}$**	**Remainder$_{10}$**	**Cumulative Binary$_{2}$**
27 (odd)	2	13	1 -> LSB	1
13 (odd)	2	6	1	11
6 (even)	2	3	0	011
3 (odd)	2	1	1	1011
1 (odd)	2	**0**	1 -> MSB	11011

**Answer: $27_{10} = 11011_{2}$

Instructions

1. Checkpoint 1 Passed

1.

We are going to be converting the number 206_{10} from decimal to binary.

Create two variables, first_result and first_remainder_lsb, and set them equal to the first stage of the Division-by-2 Method conversion.

Hint

Build a table to help with the organization:

Dividend$_{10}$	**Divisor$_{10}$**	**Result$_{10}$**	**Remainder$_{10}$**	**Cumulative Binary$_{2}$**
27	2	13	1 -> LSB	1
13	2	6	1	11
6	2	3	0	011
3	2	1	1	1011
1	2	0	1 -> MSB	11011

2. Checkpoint 2 Passed

2.

Create the variables third_result, fourth_remainder, sixth_result, and final_remainder_msb. Continue the conversion and set each variable equal to the appropriate answer.

3. Checkpoint 3 Passed

3.

Now that we've completed our Division-by-2, create the variable final_binary_number and set it equal to the complete result of your Division-By-2.

Hint

Remember to order your results from right to left, starting the with LSB and ending with the MSB.

4. Checkpoint 4 Passed

4.

Finally, uncomment the last few lines of code and view the results in the console. As you can see, most programming languages have built-in methods to do these conversions for us, here is Python's documentation on bin() and int().

script.py

```
number_to_convert = 206


# Part 1:

first_result = 103

first_remainder_lsb = 0


# Part 2:

third_result = 25

fourth_remainder = 1
```

```
sixth_result = 3

final_remainder_msb = 1

# Part 3:

final_binary_number = 11001110


# Part 4:


print(f"I convert Decimal to Binary using the 'bin()' function, see: \n52345 = {bin(52345)} in binary.\n\n")


print(f"I convert Binary to Decimal using the 'int() function, see: \n1100110001111001 =
{int(0b1100110001111001)} in decimal.")
```

```
>>
I convert Decimal to Binary using the 'bin()' function, see:

52345 = 0b1100110001111001 in binary.



I convert Binary to Decimal using the 'int() function, see:

1100110001111001 = 52345 in decimal.
```