

Binary to Decimal Conversion

9 min

The ability to know the max size of a binary number allows us to check our work when we go to find the actual value of a binary number.

A five-digit binary number can never be more than 31, or 2^5-1 because 11111 is equal to 31.

To help keep our workspaces clear and concise, it is common practice to add subscripts to numbers when working multiple numbering systems in the same space.

11111 and 31 from above should be represented as 11111_2 and 31_{10} representing their bases for clarity. If no subscript is used, it is assumed to be a decimal number.

To convert from a binary to a decimal number, make a table like the one below. For every bit that contains a 1, add that decimal number to the total. Let's look at the 8-bit number 11001110_2 .

Adding the decimal values of all the 1s highlighted in yellow gives us: $(128) + (64) + (8) + (4) + (2) = 206_{10}$.

Instructions

1. Checkpoint 1 Passed

1.

Create a new variable, `decimal_conversion1`, with the converted decimal value of 100110_2

Hint

Just like in the previous exercise, make sure your value is an int and not a String.

2. Checkpoint 2 Passed

2.

Create a new variable, `decimal_conversion2`, with the converted decimal value of 1111011110011_2

Hint

Draw out a table like the example from the exercise on some paper to keep your numbers straight.

script.py

```
decimal_conversion1 = 38
```

```
decimal_conversion2 = 7923
```