Activity: Binary Floating-Point Numbers

A. Why?

We need to represent numbers that fall between adjacent integers. Also, extremely large numbers are impractical to represent using integers. Floating-point numbers solve both problems.

B. Outcomes

After this activity you should

- Know how impractical integers are for writing extremely large values.
- Know how to use a binary point to represent fractional values
- Be vaguely familiar with the IEEE standard for floating-point numbers.

C. Questions

(As usual, in your groups)

1. We know $1024 = 2^{10}$ is roughly equal to 10^3. Take the Avogadro constant 6.02×10^{23} and say it's roughly 10^{24}. For what n is $10^{24} = (10^3)^n$? Roughly how many bits do we need to represent 10^{24} ?
2. What decimal value does 1.0101_2 represent?
3. What are the three parts of an IEEE standard floating-point number?

Question 3′: What is a better question than number 3 to test your knowledge of IEEE floating-point numbers?
In IEEE, what exponent would 10101101 represent? (10101101_2 - 127_{10})
If both the exponent and fraction of an IEEE f.p. number are zero, what number(s) does this represent? (positive 0 or negative 0; it depends on the sign bit)