Binary

- Computers work in binary for two main reasons:
  - The binary system has been around since ancient times (math-idea)
    - It was the great German philosopher/mathematician Gottfried von Leibnitz (1646-1716) who truly defined and formalized the system
  - The easiest way to design a circuit involves elementary on/off switches (engineering-technology)
    - The switch could only be in one of two possible states:
      - "on" (usually represented by a 1) or
      - "off" (usually represented by a 0)

Decimal Numbers

- Positional notation

  $275 = (2 * 100) + (7 * 10) + (5 * 1)$ or

  $275 = (2 * 10^2) + (7 * 10^1) + (5 * 10^0)$

  $5061 = (5 * 10^3) + (0 * 10^2) + (6 * 10^1) + (1 * 10^0)$

10^3	10^2	10^1	10^0
	2	7	5
5	0	6	1

Binary Integers

- Binary numbers work in the same fashion, except:
  - Each position represents the power of 2
  - We only have two digits available 0 and 1

$$1110 = (1 * 2^3) + (1 * 2^2) + (1 * 2^1) + (0 * 2^0) = ?$$

- Let's parse the following two numbers together
  1010, 0101

Binary Conversion

2^3	2^2	2^1	2^0
0	1	0	1

Integers

- A solution to negative numbers is to reserve one bit to represent the sign, e.g. 0 positive, 1 negative (excess bit)

-	2^2	2^1	2^0
1	1	0	0

- N bits represents 2^N units of information
  - With 1 bit we can represent 2 numbers: 0 or 1
  - With 2 bits we can represent 4 numbers:
    - 00, 01, 10, 11
  - With 3 bits we can represent 8 numbers:
    - 000, 001, 010, 011, 100, 101, 110, 111

Who Cares?

- The range of integer numbers we can represent depends on the number of bits being used to represent them
  - Python uses RAM
- Overflow error – result of a calculation that is a higher value than a computer can represent
- Underflow error – result of a calculation that is a lower value than a computer can represent

Real Numbers: Decimal

- How do we use positional notation to represent fractions – let's only consider the digits to the right of the radix point?

  0.2 = ?

  0.034 = ?

10^{-1}	10^{-2}	10^{-3}	10^{-4}
2			
0	3	4	0

Real Numbers: Decimal

- Let's put the two elements together and consider

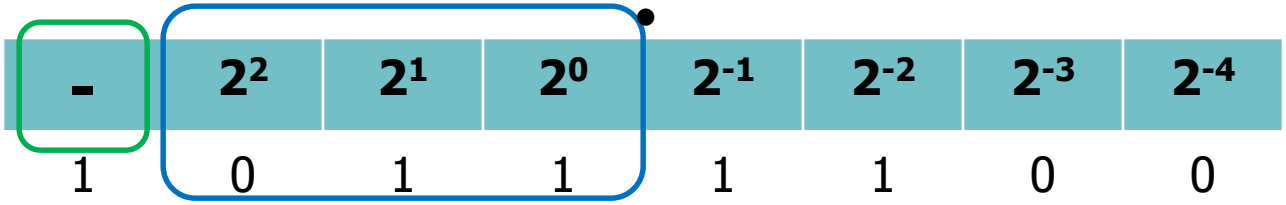
  275.2 = ?

  5061.034 = ?

10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}
	2	7	5	2			
5	0	6	1	0	3	4	0

Real Numbers: Binary

- Example uses 1 byte (8 bits)
  - First bit denotes the number sign
  - Next three denote the whole part of a number
  - Last four denote the fraction and use negative powers of 2



- (2 + 1 . ½ + ¼)

-3.75

Who Cares?

- The precision of a number depends on the number of bits used to represent fractions.

- Not all numbers can be represented – representational error (π)

- A number of arithmetic operations may lead to a
  - Round-off error

    1.0 / 3 = 0.333…

    0.333 * 3 = 0.999 (not 1!)