

# CSE 351 Lecture 1 – Binary

## Base Representation

A **numeral** represents a number using a series of **digits** in a particular **base**. Each digit is one of a fixed set of **symbols** that represent values between 0 and base – 1. We are taught to think in base 10 (*decimal*, which uses the ten symbols 0 through 9). We number the digit **positions** from the right, starting with 0. In decimal, the value of the digit  $d$  in position  $i$  in base  $b$  is  $d \times b^i$ .

$$\begin{aligned} \text{digit position} &\rightarrow 2 \quad 1 \quad 0 \\ \text{digit symbol} &\rightarrow 3 \quad 5 \quad 1_{10} = 3 \times 10^2 + 5 \times 10^1 + 1 \times 10^0 \end{aligned}$$

In these examples, the subscript on the numeral indicates the base (no subscript means base 10):

$$\begin{aligned} 1011_2 &= 2^3 + 2^1 + 2^0 = 11 = 1 \times 3^2 + 2 \times 3^0 = 12_3 \\ -1234_5 &= -(1 \times 5^3 + 2 \times 5^2 + 3 \times 5^1 + 4 \times 5^0) = -194 \end{aligned}$$

**Note:** We're still in the realm of pure math here, so the negative sign sits separately from the number and does its own thing. All numbers can be assumed to have an infinite number of *leading zeros*, which would appear to the left of all shown digits but add no value.

## Binary

Computers use base 2 (*binary*). Each **binary digit** is called a *bit*. Collections of bits can be easily grouped and converted into other bases that are powers of two. The most common grouping is four bits (base  $2^4 =$  base 16, which is called *hexadecimal* or *hex*). In order to get to 16 symbols, we incorporate the first six letters of the alphabet, which can be equivalently written in upper or lowercase.

These are the most common bases that we will use in this class: 2, 10, and 16. We differentiate between these bases by adding the prefix '0b' for binary and '0x' for hexadecimal; there is no prefix for decimal. The table on the right shows the equivalent numerals for the numbers 0 through 15 in these three major number bases.

Binary	Decimal	Hex
0b0000	0	0x0
0b0001	1	0x1
0b0010	2	0x2
0b0011	3	0x3
0b0100	4	0x4
0b0101	5	0x5
0b0110	6	0x6
0b0111	7	0x7
0b1000	8	0x8
0b1001	9	0x9
0b1010	10	0xA
0b1011	11	0xB
0b1100	12	0xC
0b1101	13	0xD
0b1110	14	0xE
0b1111	15	0xF

**Note:** In C, literals (*i.e.* constants) are expressed in decimal by default. You can use the appropriate prefix to express literals in binary or hexadecimal instead.

However, you should NOT use binary constants in Lab 1a or Lab 1b.