

Bit, Byte, and Binary

Number of bits	Number of values	2 raised to the power	Number of bytes	Unit	
1	2	1		Bit	0 / 1
2	4	2			
3	8	3			
4	16	4		Nibble	Hexadecimal unit
5	32	5			
6	64	6			
7	128	7			
8	256	8	1	Byte	One character
9	512	9			
10	1024	10			
16	65,536	16	2		

Number of bytes	2 raised to the power	Unit	
1		Byte	One character
1024	10	KiloByte (Kb)	Small text
1,048,576	20	MegaByte (Mb)	A book
1,073,741,824	30	GigaByte (Gb)	An large encyclopedia
1,099,511,627,776	40	TeraByte	

bit: Short for binary digit, the smallest unit of information on a machine. John Tukey, a leading statistician and adviser to five presidents first used the term in 1946. A single bit can hold only one of two values: 0 or 1. More meaningful information is obtained by combining consecutive bits into larger units. For example, a byte is composed of 8 consecutive bits. Computers are sometimes classified by the number of bits they can process at one time or by the number of bits they use to represent addresses. These two values are not always the same, which leads to confusion. For example, classifying a computer as a 32-bit machine might mean that its data registers are 32 bits wide or that it uses 32 bits to identify each address in memory. Whereas larger registers make a computer faster, using more bits for addresses enables a machine to support larger programs. Graphics are also often described by the number of bits used to represent each dot. A 1-bit image is monochrome; an 8-bit image supports 256 colors or grayscales; and a 24- or 32-bit graphic supports true color.

nibble: Half a byte – four bits. Nibbles are important in hexadecimal and BCD representations. The term is sometimes spelled nybble.

byte: Abbreviation for binary term, a unit of storage capable of holding a single character. On almost all modern computers, a byte is equal to 8 bits. Large amounts of memory are indicated in terms of kilobytes, megabytes, and gigabytes. A disk that can hold 1.44 megabytes, for example, is capable of storing approximately 1.4 million characters, or about 3,000 pages of information.

hexadecimal: Refers to the base-16 number system, which consists of 16 unique symbols: the numbers 0 to 9 and the letters A to F. For example, the decimal number 15 is represented as F in the hexadecimal numbering system. The hexadecimal system is useful because it can represent every byte (8 bits) as two consecutive hexadecimal digits representing the two parts (nibble) of the byte. It is easier for humans to read hexadecimal numbers than binary numbers. To convert a value from hexadecimal to binary, you merely translate each hexadecimal digit into its 4-bit binary equivalent. Hexadecimal numbers have either an 0x prefix or an h suffix. For example, the hexadecimal number 3F7A translates to the binary number 0011 1111 0111 1010

big-endian: Refers to which bytes are most significant in multi-byte data types. In big-endian architectures, the leftmost bytes (those with a lower address) are most significant. In little-endian architectures, the rightmost bytes are most significant. For example, consider the number 1025 (2 to the tenth power plus one) stored in a 4-byte integer

	Big-Endian representation of 1025	Little-Endian representation of 1025
first byte	00000000	00000001
second byte	00000000	00000100
third byte	00000100	00000000
fourth byte	00000001	00000000

Note that the example above shows only big- and little-endian byte orders. The bit ordering within each byte can also be big- or little-endian, and some architectures actually use big-endian ordering for bits and little-endian ordering for bytes, or vice versa. The terms big-endian and little-endian are derived from the Lilliputians of Gulliver's Travels, whose major political issue was whether soft-boiled eggs should be opened on the big side or the little side. Likewise, the big-/little-endian computer debate has much more to do with political issues than technological merits.