

# What Are Binary And Hexadecimal Numbers

## Binary number

*A binary number is a number expressed in the base-2 numeral system or binary numeral system, a method for representing numbers that uses only two symbols*

A binary number is a number expressed in the base-2 numeral system or binary numeral system, a method for representing numbers that uses only two symbols for the natural numbers: typically "0" (zero) and "1" (one). A binary number may also refer to a rational number that has a finite representation in the binary numeral system, that is, the quotient of an integer by a power of two.

The base-2 numeral system is a positional notation with a radix of 2. Each digit is referred to as a bit, or binary digit. Because of its straightforward implementation in digital electronic circuitry using logic gates, the binary system is used by almost all modern computers and computer-based devices, as a preferred system of use, over various other human techniques of communication, because of the simplicity of the language and the noise immunity in physical implementation.

## Binary file

*data as a sequence of hexadecimal (or decimal, binary or ASCII character) values for corresponding bytes of a binary file. If a binary file is opened in a*

A binary file is a computer file that is not a text file. The term "binary file" is often used as a term meaning "non-text file". Many binary file formats contain parts that can be interpreted as text; for example, some computer document files containing formatted text, such as older Microsoft Word document files, contain the text of the document but also contain formatting information in binary form.

## IEEE 754

*for Testing IEEE Decimal–Binary Conversion, Manuscript, CiteSeerX 10.1.1.144.5889 IEEE 754 2008, §5.12.3 &quot;6.9.3. Hexadecimal floating point literals —*

The IEEE Standard for Floating-Point Arithmetic (IEEE 754) is a technical standard for floating-point arithmetic originally established in 1985 by the Institute of Electrical and Electronics Engineers (IEEE). The standard addressed many problems found in the diverse floating-point implementations that made them difficult to use reliably and portably. Many hardware floating-point units use the IEEE 754 standard.

The standard defines:

arithmetic formats: sets of binary and decimal floating-point data, which consist of finite numbers (including signed zeros and subnormal numbers), infinities, and special "not a number" values (NaNs)

interchange formats: encodings (bit strings) that may be used to exchange floating-point data in an efficient and compact form

rounding rules: properties to be satisfied when rounding numbers during arithmetic and conversions

operations: arithmetic and other operations (such as trigonometric functions) on arithmetic formats

exception handling: indications of exceptional conditions (such as division by zero, overflow, etc.)

IEEE 754-2008, published in August 2008, includes nearly all of the original IEEE 754-1985 standard, plus the IEEE 854-1987 (Radix-Independent Floating-Point Arithmetic) standard. The current version, IEEE 754-2019, was published in July 2019. It is a minor revision of the previous version, incorporating mainly clarifications, defect fixes and new recommended operations.

0

*no tens, and five ones. The same principle applies in place-value notations that uses a base other than ten, such as binary and hexadecimal. The modern*

0 (zero) is a number representing an empty quantity. Adding (or subtracting) 0 to any number leaves that number unchanged; in mathematical terminology, 0 is the additive identity of the integers, rational numbers, real numbers, and complex numbers, as well as other algebraic structures. Multiplying any number by 0 results in 0, and consequently division by zero has no meaning in arithmetic.

As a numerical digit, 0 plays a crucial role in decimal notation: it indicates that the power of ten corresponding to the place containing a 0 does not contribute to the total. For example, "205" in decimal means two hundreds, no tens, and five ones. The same principle applies in place-value notations that uses a base other than ten, such as binary and hexadecimal. The modern use of 0 in this manner derives from Indian mathematics that was transmitted to Europe via medieval Islamic mathematicians and popularized by Fibonacci. It was independently used by the Maya.

Common names for the number 0 in English include zero, nought, naught (), and nil. In contexts where at least one adjacent digit distinguishes it from the letter O, the number is sometimes pronounced as oh or o (). Informal or slang terms for 0 include zilch and zip. Historically, ought, aught (), and cipher have also been used.

Double-precision floating-point format

*binary floating-point is a commonly used format on PCs, due to its wider range over single-precision floating point, in spite of its performance and bandwidth*

Double-precision floating-point format (sometimes called FP64 or float64) is a floating-point number format, usually occupying 64 bits in computer memory; it represents a wide range of numeric values by using a floating radix point.

Double precision may be chosen when the range or precision of single precision would be insufficient.

In the IEEE 754 standard, the 64-bit base-2 format is officially referred to as binary64; it was called double in IEEE 754-1985. IEEE 754 specifies additional floating-point formats, including 32-bit base-2 single precision and, more recently, base-10 representations (decimal floating point).

One of the first programming languages to provide floating-point data types was Fortran. Before the widespread adoption of IEEE 754-1985, the representation and properties of floating-point data types depended on the computer manufacturer and computer model, and upon decisions made by programming-language implementers. E.g., GW-BASIC's double-precision data type was the 64-bit MBF floating-point format.

Floating-point arithmetic

*mainframes support IBM's own hexadecimal floating point format and IEEE 754-2008 decimal floating point in addition to the IEEE 754 binary format. The Cray T90*

In computing, floating-point arithmetic (FP) is arithmetic on subsets of real numbers formed by a significand (a signed sequence of a fixed number of digits in some base) multiplied by an integer power of that base.

Numbers of this form are called floating-point numbers.

For example, the number 2469/200 is a floating-point number in base ten with five digits:

2469

/

200

=

12.345

=

12345

?

significand

×

10

?

base

?

3

?

exponent

$$\{ \displaystyle 2469/200 = 12.345 = \underbrace{\{ 12345 \}}_{\text{significand}} \times \underbrace{\{ 10 \}}_{\text{base}} \overbrace{\{ \}^{-3}}^{\text{exponent}} \}$$

However, 7716/625 = 12.3456 is not a floating-point number in base ten with five digits—it needs six digits.

The nearest floating-point number with only five digits is 12.346.

And 1/3 = 0.3333... is not a floating-point number in base ten with any finite number of digits.

In practice, most floating-point systems use base two, though base ten (decimal floating point) is also common.

Floating-point arithmetic operations, such as addition and division, approximate the corresponding real number arithmetic operations by rounding any result that is not a floating-point number itself to a nearby floating-point number.

For example, in a floating-point arithmetic with five base-ten digits, the sum  $12.345 + 1.0001 = 13.3451$  might be rounded to 13.345.

The term floating point refers to the fact that the number's radix point can "float" anywhere to the left, right, or between the significant digits of the number. This position is indicated by the exponent, so floating point can be considered a form of scientific notation.

A floating-point system can be used to represent, with a fixed number of digits, numbers of very different orders of magnitude — such as the number of meters between galaxies or between protons in an atom. For this reason, floating-point arithmetic is often used to allow very small and very large real numbers that require fast processing times. The result of this dynamic range is that the numbers that can be represented are not uniformly spaced; the difference between two consecutive representable numbers varies with their exponent.

Over the years, a variety of floating-point representations have been used in computers. In 1985, the IEEE 754 Standard for Floating-Point Arithmetic was established, and since the 1990s, the most commonly encountered representations are those defined by the IEEE.

The speed of floating-point operations, commonly measured in terms of FLOPS, is an important characteristic of a computer system, especially for applications that involve intensive mathematical calculations.

Floating-point numbers can be computed using software implementations (softfloat) or hardware implementations (hardfloat). Floating-point units (FPUs, colloquially math coprocessors) are specially designed to carry out operations on floating-point numbers and are part of most computer systems. When FPUs are not available, software implementations can be used instead.

#### Single-precision floating-point format

*examples are given in bit representation, in hexadecimal and binary, of the floating-point value. This includes the sign, (biased) exponent, and significand*

Single-precision floating-point format (sometimes called FP32 or float32) is a computer number format, usually occupying 32 bits in computer memory; it represents a wide dynamic range of numeric values by using a floating radix point.

A floating-point variable can represent a wider range of numbers than a fixed-point variable of the same bit width at the cost of precision. A signed 32-bit integer variable has a maximum value of  $2^{31} - 1 = 2,147,483,647$ , whereas an IEEE 754 32-bit base-2 floating-point variable has a maximum value of  $(2^{23} - 1) \times 2^{127} \approx 3.4028235 \times 10^{38}$ . All integers with seven or fewer decimal digits, and any  $2^n$  for a whole number  $n$  between  $-149$  and  $127$ , can be converted exactly into an IEEE 754 single-precision floating-point value.

In the IEEE 754 standard, the 32-bit base-2 format is officially referred to as binary32; it was called single in IEEE 754-1985. IEEE 754 specifies additional floating-point types, such as 64-bit base-2 double precision and, more recently, base-10 representations.

One of the first programming languages to provide single- and double-precision floating-point data types was Fortran. Before the widespread adoption of IEEE 754-1985, the representation and properties of floating-point data types depended on the computer manufacturer and computer model, and upon decisions made by programming-language designers. E.g., GW-BASIC's single-precision data type was the 32-bit MBF floating-point format.

Single precision is termed REAL(4) or REAL\*4 in Fortran; SINGLE-FLOAT in Common Lisp; float binary(p) with  $p \geq 21$ , float decimal(p) with the maximum value of  $p$  depending on whether the DFP (IEEE

754 DFP) attribute applies, in PL/I; float in C with IEEE 754 support, C++ (if it is in C), C# and Java; Float in Haskell and Swift; and Single in Object Pascal (Delphi), Visual Basic, and MATLAB. However, float in Python, Ruby, PHP, and OCaml and single in versions of Octave before 3.2 refer to double-precision numbers. In most implementations of PostScript, and some embedded systems, the only supported precision is single.

## Base64

*Hexadecimal to octal transformation is useful to convert between binary and Base64. Such conversion is available for both advanced calculators and programming*

In computer programming, Base64 is a group of binary-to-text encoding schemes that transforms binary data into a sequence of printable characters, limited to a set of 64 unique characters. More specifically, the source binary data is taken 6 bits at a time, then this group of 6 bits is mapped to one of 64 unique characters.

As with all binary-to-text encoding schemes, Base64 is designed to carry data stored in binary formats across channels that only reliably support text content. Base64 is particularly prevalent on the World Wide Web where one of its uses is the ability to embed image files or other binary assets inside textual assets such as HTML and CSS files.

Base64 is also widely used for sending e-mail attachments, because SMTP – in its original form – was designed to transport 7-bit ASCII characters only. Encoding an attachment as Base64 before sending, and then decoding when received, assures older SMTP servers will not interfere with the attachment.

Base64 encoding causes an overhead of 33–37% relative to the size of the original binary data (33% by the encoding itself; up to 4% more by the inserted line breaks).

## Dot-decimal notation

*numerical data expressed as a string of decimal numbers each separated by a full stop. For example, the hexadecimal number 0xFF000000 may be expressed in dot-decimal*

Dot-decimal notation is a presentation format for numerical data. It consists of a string of decimal numbers, using the full stop (., also called dot in computing) as a separation character.

A common use of dot-decimal notation is in information technology, where it is a method of writing numbers in octet-grouped base-ten (decimal) numbers. In computer networking, Internet Protocol Version 4 (IPv4) addresses are commonly written using the dotted-quad notation of four decimal integers, ranging from 0 to 255 each.

## Byte

*a nibble, also nybble, which is conveniently represented by a single hexadecimal digit. The term octet unambiguously specifies a size of eight bits. It*

The byte is a unit of digital information that most commonly consists of eight bits. Historically, the byte was the number of bits used to encode a single character of text in a computer and for this reason it is the smallest addressable unit of memory in many computer architectures. To disambiguate arbitrarily sized bytes from the common 8-bit definition, network protocol documents such as the Internet Protocol (RFC 791) refer to an 8-bit byte as an octet. Those bits in an octet are usually counted with numbering from 0 to 7 or 7 to 0 depending on the bit endianness.

The size of the byte has historically been hardware-dependent and no definitive standards existed that mandated the size. Sizes from 1 to 48 bits have been used. The six-bit character code was an often-used

implementation in early encoding systems, and computers using six-bit and nine-bit bytes were common in the 1960s. These systems often had memory words of 12, 18, 24, 30, 36, 48, or 60 bits, corresponding to 2, 3, 4, 5, 6, 8, or 10 six-bit bytes, and persisted, in legacy systems, into the twenty-first century. In this era, bit groupings in the instruction stream were often referred to as syllables or slab, before the term byte became common.

The modern de facto standard of eight bits, as documented in ISO/IEC 2382-1:1993, is a convenient power of two permitting the binary-encoded values 0 through 255 for one byte, as 2 to the power of 8 is 256. The international standard IEC 80000-13 codified this common meaning. Many types of applications use information representable in eight or fewer bits and processor designers commonly optimize for this usage. The popularity of major commercial computing architectures has aided in the ubiquitous acceptance of the 8-bit byte. Modern architectures typically use 32- or 64-bit words, built of four or eight bytes, respectively.

The unit symbol for the byte was designated as the upper-case letter B by the International Electrotechnical Commission (IEC) and Institute of Electrical and Electronics Engineers (IEEE). Internationally, the unit octet explicitly defines a sequence of eight bits, eliminating the potential ambiguity of the term "byte". The symbol for octet, 'o', also conveniently eliminates the ambiguity in the symbol 'B' between byte and bel.

<https://www.onebazaar.com.cdn.cloudflare.net/~83197750/tprescriben/xidentifyq/pmanipulatee/autodesk+fusion+36>  
[https://www.onebazaar.com.cdn.cloudflare.net/\\$94140923/nexperienex/pregulateo/econceiveb/aar+manual+truck+c](https://www.onebazaar.com.cdn.cloudflare.net/$94140923/nexperienex/pregulateo/econceiveb/aar+manual+truck+c)  
<https://www.onebazaar.com.cdn.cloudflare.net/~14784592/aprescribey/bfunctionk/iovercomeq/teaching+spoken+eng>  
<https://www.onebazaar.com.cdn.cloudflare.net/!80155255/rdiscoverm/xrecognisel/gorganiseu/k+to+12+curriculum+>  
<https://www.onebazaar.com.cdn.cloudflare.net/~26021981/sprescribeg/acriticizec/rtransporti/honda+vt750dc+service>  
<https://www.onebazaar.com.cdn.cloudflare.net/=47962936/wdiscoverg/nrecogniseu/vparticipateh/opel+frontera+b+s>  
<https://www.onebazaar.com.cdn.cloudflare.net/+31400626/gencountera/sunderminej/wmanipulateb/the+westminster>  
<https://www.onebazaar.com.cdn.cloudflare.net/=24809220/utransferl/jidentifyd/pparticipateg/yamaha+audio+user+n>  
[https://www.onebazaar.com.cdn.cloudflare.net/\\$39533820/rapproachk/gdisappearw/xparticipaten/english+1+b+unit-](https://www.onebazaar.com.cdn.cloudflare.net/$39533820/rapproachk/gdisappearw/xparticipaten/english+1+b+unit-)  
[https://www.onebazaar.com.cdn.cloudflare.net/\\$66542362/pencountere/scriticizeg/xmanipulateo/9th+std+english+m](https://www.onebazaar.com.cdn.cloudflare.net/$66542362/pencountere/scriticizeg/xmanipulateo/9th+std+english+m)