

NUMBERS IN JAVASCRIPT

Math constants

JavaScript understands numbers in several formats, allowing you to specify numbers in hex, decimal, and octal. If a 0 precedes a number and there is no number higher than 7, the number is considered to be in octal (base 8) and if the number is preceded by 0x, then it is considered to be in hex (base 16) and can also contain characters A, B, C, D, E, F. Neither may contain decimal points.

With decimal numbers, $12e+4$ may be used to replace 12×10^4 and $12e-4$ may be used to replace 12×10^{-4} etc.

There are a few maths constants (or properties) that are held by the Math object and can be used as variables, These values are read only, you cannot assign values to them.

The available Math object constants are:

Math object constants		
Math object property	Value (approx)	Mathematical equivalent
<i>Math.E</i>	2.718	e
<i>Math.LN2</i>	0.693	ln(2)
<i>Math.LN10</i>	2.303	ln(10)
<i>Math.LOG2E</i>	1.442	$\log_2(e)$
<i>Math.LOG10E</i>	0.434	$\log_{10}(e)$
<i>Math.PI</i>	3.142	Pi
<i>Math.SQRT1_2</i>	0.707	$(\sqrt{2})^{-1}$ or $\sqrt{1/2}$
<i>Math.SQRT2</i>	1.414	$\sqrt{2}$

Precision - javascript Math.round function

The concepts of accuracy and precision are often confused.

The accuracy of a number x is given by the number of significant decimal (or other) digits to the right of the decimal point in x.

the precision of x is the total number of significant decimal (or other) digits.

The Math.round() function will round a number (up or down) to the nearest integer. So the number 123.45 would become 123 while the number 1234.5 would become 1235. When performing arithmetic the computer will return an answer to the full precision available.

Some fractions such as $1/3$ when converted to decimal go on for ever. e.g. $1/3 = 0.3333333 \dots$. Because the computer works to some fixed precision, it must round up (or down) such values. In doing so, an error is introduced, albeit usually very small for most practical purposes.

To round number to a specific number of decimal places, for example 4 first multiply the number by $1E4$, then use the `Math.round()` function to get the integer value. Then divide this answer by $1E4$ to obtain the desired level of accuracy.

```
function precision(number,place)
{
    //convert place to exponential notation
    //E.g. 3 becomes 1E3 in scientific notation (or 1000).
    //multiply number by place and round the answer
    //then divide the answer by place to get a value
    //correct to the required accuracy.
    place = eval('1e' + place);
    return Math.round(number*place)/place;
}
```



When a number is specified as 100 ± 1 it means its true value can fall in the range 99-101, while a quantity specified as $100 \pm 2\%$ has a (relative) accuracy of + or -2% (meaning its true value can fall in the range 98-102).

The number of significant figures of a multiplication, or division of two or more quantities is equal to the smallest number of significant figures for the quantities involved. E.g. $12.345 * 1.2 = 148.14$ but with rounding should be 148.1 since 1.2 is only significant to + or -0.1.

For addition or subtraction, the number of significant figures is determined with the smallest significant figure of all the quantities involved. For example, the sum $12.345 + 1.2 + 123.4567$ is 137.0017, but should be written 137.0 (with rounding), since the quantity 1.2 is significant only to ± 0.1 .

Spurious precision

Take for example a vehicle whose time it takes to travel 1Km, measured with a stop watch, is 74.6 seconds. The stop watch is only accurate to ± 0.1 seconds so the actual time could lie between 74.7 and 74.5. By dividing the distance by the time $1000\text{m}/74.6\text{s}$ we get a speed of 13.40482574m/s . But it could be $(1000/74.7)$ 13.38688086m/s or $(1000/74.5)$ 13.42281879m/s . Which is correct? Even if these 3 numbers are rounded to 2 decimal places, the answers don't agree, 13.39m/s , 13.40m/s and 13.42m/s . Only when

rounded to 1 decimal place, as indicated by the accuracy of the stop do the answers agree. e.g. 13.4m/s

Rounding and spurious precision

The concept of **significant digits** is often used in connection with rounding. Rounding to n significant digits is a more general-purpose technique than rounding to n decimal places, since it handles numbers of different scales in a uniform way.

Be careful when doing calculations on a computer to give answers that reflect the accuracy of "measurement" **not** the accuracy of the computer

When numbers are expressed in the normalized form of scientific notation, placeholder leading and trailing digits do not occur, so all digits are significant. For example, 0.00012 (two significant digits) becomes 1.2×10^{-4} , and 0.000123400 (six significant digits) becomes 1.23400×10^{-4} . The potential ambiguity about the significance of trailing zeros is eliminated with scientific notation. E.g 1200 to four significant digits is written as 1.200×10^3 , while 1200 to two significant digits is written as 1.2×10^3 .

Source : <http://www.soslug.org/node/1725>