

Numbers in JavaScript

JavaScript Numbers can be **integers** (exp 3) or **whole numbers** (exp 3.14). **JavaScript Numbers** can perform **addition, subtraction, multiplication, division** and **Modulus**. Thus all arithmetic operations can be performed using **javascript numbers**.

JavaScript Supports all numbers systems, including **Binary** , **Octal**, **Decimal** and **Hexadecimal** numbers. A number starting with 0 is by-default octal in JavaScript. For example 010 is 8 in javascript.

var is used to declare numbers in JavaScript.

For fixed numbers, **const** is also used.

typeof operator can check datatype of numbers.

Define Numbers in JS

```
var a=3;           // number;
var b=3.6;         // number with decimal;
var c=2e3;         // exponential number (2000)
var d=0xa;         // number in hexadecimal
var e=010;         // number in octal
var f=0o10;        // number in octal
var g=0b100;       // number in binary
var h=NaN;         // not a number
var i=Infinity;    // number more than 10308
```

Binary, Octal, Decimal and Hexadecimal Numbers

Here is a comparison of **Binary numbers**, **Octal numbers**, **Decimal numbers** and **Hexadecimal numbers**.

Number	Binary (0-1) (2 bit)	Octal (0-7) (8 bit)	Decimal (0-9) (10 bit)	Hexadecimal (0-f) (16 bit)

0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
8	1000	10	8	8
10	1010	12	10	a
15	1111	17	15	f
16	10000	20	16	10

Numbers Method

Numbers Methods are used to convert a number to string, exponential, precision and fixed. Here are **number methods** with example.

JavaScript Number Methods		
Method	use	Example
toString()	convert number to string.	<pre>var x=6; x.toString()="6";</pre>
toLocaleString()	convert number to local string.	<pre>var x=6; x.toLocaleString()="6";</pre>
toExponential()	convert decimal to exponential notation.	<pre>var x=6; x.toExponential()="6e+0";</pre>
toPrecision(1)	convert number to Precise .	<pre>var x=1.23456; x.toPrecision(1)="1";</pre>
toPrecision(2)	convert decimal to precision 2 .	<pre>var x=1.23456; x.toPrecision(2)="1.2";</pre>
toPrecision(3)	convert decimal to precision 3 .	<pre>var x=1.23456; x.toPrecision(3)="1.23";</pre>
toFixed()	to convert a number to string with fixed decimal value	<pre>var pi=3.1416; pi.toFixed(); // "3" pi.toFixed(1); // "3.1"</pre>

```
pi.toFixed(2); // "3.14"  
pi.toFixed(3); // "3.142"
```

String to Number in javascript

Javascript variables can be strings or numbers. Like `var x="5"` is string, but `var y=5` is number. Both can be used in arithmetic operations. Except addition, all arithmetic operations are possible with x and y.

Functions to convert string to number.

Number Function

parseInt Function

parseFloat Function

Number Function

Number Function can convert string to numbers. **Number Function** can convert both floating/decimal and non-floating/integers. But if a string contains string character like alphabets or special character, **number function** will return NaN.

```
var a="100";  
var b="100.5";  
var c="100px";  
var d="abc100";
```

```
Number(a)      //100  
Number(b)      //100.5  
Number(c)      //NaN - Not a Number  
Number(d)      //NaN - Not a Number
```

Number Function Example

Value:

parseInt Function

parseInt Function can convert string to numbers, but non-floating/integers values only. **parseInt** can also convert binary, octal and hexadecimal to decimal numbers.

```
var a="100";  
var b="100.5";  
var c="100px";  
var d="abc100";  
  
parseInt(a)      //100  
parseInt(b)      //100  
parseInt(c)      //100  
parseInt(d)      //NaN - Not a Number
```

parseInt function is also used to convert **decimal to binary**, **decimal to octal** and **decimal to hexadecimal** numbers. To do this, pass second parameter in **parseInt function** as **2 for binary**, **8 for octal**, and **16 for hexadecimal**. See example.

```
var a=100;  
  
parseInt(a,2)     //4, binary number  
parseInt(a,8)     //64, octal number  
parseInt(a,10)    //100, decimal number  
parseInt(a,16)    //256, hexadecimal number
```

parseInt Example

Value:

parseFloat Function

parseFloat Function can convert string to numbers, floating/decimals and non-floating/integers both. **parseFloat** can also convert binary, octal and hexadecimal to decimal numbers.

```
var a="100";
var b="100.5";
var c="100px";
var d="100.5px";
var e="abc100";

parseFloat(a)      //100
parseFloat(b)      //100.5
parseFloat(c)      //100
parseFloat(d)      //100.5
parseFloat(e)      //NaN - Not a Number
```

parseFloat Example

Value:

isNaN, is Not a Number Function

JavaScript **isNaN** function returns a boolean value. For example, `isNaN("2")` is false and `isNaN("a")` is true. Even `isNaN(NaN)` is also true.

```
isNaN(NaN)           // returns true
isNaN(1)             // returns false
isNaN("1")           // returns false
isNaN("a")           // returns true
isNaN("2a")          // returns true
```

isNaN Example

Value:

isFinite

isFinite function tells whether a **number is finite** or not. In JavaScript, number less than $1e308$ or $1e+308$ are finite numbers. Numbers greater than $1e308$ are **Infinite**, for example, $2e308$ and more are infinite numbers in JavaScript.

```
isFinite(Infinity)   // false
isFinite(2e308)       // false
isFinite(1e308)       // true
isFinite(1e307)       // true
isFinite(123)         // true
```

isFinite Example

Value:

1. Never starts a number with 0 prefix.
2. Floating numbers can have maximum 16 characters after decimal.
3. Binary i.e. **0b100** and Octal **0o100** are not supported in all browsers.