20. JavaScript Number Object

JavaScript has only one type of number.

Numbers can be written with, or without decimals.

20.1. JavaScript Numbers

JavaScript numbers can be written with, or without decimals:

Example

```
    var pi=3.14;     // Written with decimals
    var x=34;        // Written without decimals
```

Extra large or extra small numbers can be written with scientific (exponent) notation:

Example

```
     var y=123e5;     // 12300000
    var z=123e-5;    // 0.00123
```

20.2. All JavaScript Numbers are 64-bit

JavaScript is not a typed language. Unlike many other programming languages, it does not define different types of numbers, like integers, short, long, floating-point etc.

All numbers in JavaScript are stored as 64-bit (8-bytes) base 10, floating point numbers.

a) Precision

Integers (numbers without a period or exponent notation) are considered accurate up to 15 digits.

The maximum number of decimals is 17, but floating point arithmetic is not always 100% accurate:

Example

```
    var x=0.2+0.1;
```

b) Octal and Hexadecimal

JavaScript interprets numeric constants as octal if they are preceded by a zero, and as hexadecimal if they are preceded by a zero and x.

Example

```
var y=0377;
var z=0xFF;
```

Never write a number with a leading zero, unless you want an octal conversion.

20.3. Number Properties and Methods

Properties:

- MAX VALUE
- MIN VALUE
- NEGATIVE INFINITY
- POSITIVE INFINITY
- NaN
- prototype
- constructor

Methods:

- toExponential()
- toFixed()
- toPrecision()
- toString()
- valueOf()

20.4. Complete Number Object Reference

For a complete reference of all the properties and methods that can be used with the Number object, go to our Complete Number Object Reference.

The reference contains both descriptions and examples, for each property and method.

The Number object is an object wrapper for primitive numeric values.

Number objects are created with new Number().

Syntax

```
var num = new Number(value);
```

Note: If the value parameter cannot be converted into a number, it returns NaN (Not-a-Number).

Number Object Properties

Property	Description
constructor	Returns the function that created the Number object's prototype
MAX_VALUE	Returns the largest number possible in JavaScript
MIN_VALUE	Returns the smallest number possible in JavaScript
NEGATIVE_INFINITY	Represents negative infinity (returned on overflow)
NaN	Represents a "Not-a-Number" value
POSITIVE_INFINITY	Represents infinity (returned on overflow)
prototype	Allows you to add properties and methods to an object

Number Object Methods

Method	Description
toExponential(x)	Converts a number into an exponential notation
toFixed(x)	Formats a number with x numbers of digits after the decimal point
toPrecision(x)	Formats a number to x length
toString()	Converts a Number object to a string
valueOf()	Returns the primitive value of a Number object