Web Development

JavaScript

Numbers

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

JavaScript numbers are always stored as double precision floating point numbers, following the international IEEE 754 standard.

This format stores numbers in 64 bits, where the number (the fraction) is stored in bits 0 to 51, the exponent in bits 52 to 62, and the sign in bit 63:

Value (aka Fraction/Mantissa)	Exponent	Sign
52 bits (0 - 51)	11 bits (52 - 62)	1 bit (63)

NaN - Not a Number

NaN is a JavaScript reserved word indicating that a number is not a legal number.

Trying to do arithmetic with a non-numeric string will result in NaN (Not a Number):

```
var x = 100 / "Apple";
isNaN(x);                 // returns true because x is Not a Number
typeof NaN;              // returns "number"
var x =  2 / 0;           // x will be Infinity
var y = -2 / 0;           // y will be -Infinity
typeof Infinity;         // returns "number"
```

Number Methods

- Methods
  - .toString() : returns a number as a string.
  - .toExponential() : returns a string, with a number rounded and written using exponential notation.
  - .toFixed(): returns a string, with the number written with a specified number of decimals.
  - .toPrecision(): returns a string, with a number written with a specified length.
  - .valueOf(): returns a number as a number.
- Global Methods (can be used on all JavaScript data types)
  - Number()
  - parseFloat()
  - parseInt()
- Properties
  - .MAX_VALUE
  - .MIN_VALUE
  - .NEGATIVE_INFINITY
  - .POSITIVE_INFINITY
  - .NAN

Math Object

Math.round(x) returns the value of x rounded to its nearest integer.

Math.pow(x, y) returns the value of x to the power of y.

Math.sqrt(x) returns the square root of x.

Math.abs(x) returns the absolute (positive) value of x.

Math.floor(x) returns the value of x rounded **down** to its nearest integer.

Math.min() and Math.max() can be used to find the lowest or highest value in a list of arguments.

Math.random() returns a random number between 0 (inclusive), and 1 (exclusive)

More

Dates

The Date object lets you work with dates (years, months, days, hours, minutes, seconds, and milliseconds).

A JavaScript date can be written as a string:

Wed Oct 11 2017 16:12:12 GMT-0700 (PDT)

or as a number:

1507763532160

Dates written as numbers, specify the number of milliseconds since January 1, 1970, 00:00:00.

Creating Date Objects

The Date object lets us work with dates.

A date consists of a year, a month, a day, an hour, a minute, a second, and milliseconds.

Date objects are created with the **new Date()** constructor.

There are **4 ways** of initiating a date

```
new Date()
new Date(milliseconds)
new Date(dateString)
new Date(year, month, day, hours, minutes, seconds, milliseconds)
```

Date Methods

The **toUTCString()** method converts a date to a UTC string (a date display standard).

ex) Wed, 11 Oct 2017 23:42:00 GMT

The **toDateString()** method converts a date to a more readable format.

ex) Wed Oct 11 2017

Date Formats

There are generally 4 types of JavaScript date input formats:

Type	Example
ISO Date	"2015-03-25" (The International Standard)
Short Date	"03/25/2015"
Long Date	"Mar 25 2015" or "25 Mar 2015"
Full Date	"Wednesday March 25 2015"

Example

Date Methods

Date methods let you **get** and **set** date values (years, months, days, hours, minutes, seconds, milliseconds).

EXAMPLE