Number & Boolean

Number

- **JavaScript Numbers are Always 64-bit Floating Point.**

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

- **Precision**

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

```
Example :
        Case 1 : let x = 999999999999999;   // x will be 999999999999999
                 when <=15 digits
        Case 2 : let y = 9999999999999999;  // y will be 10000000000000000
                 when >=15 digits
```

- **Automatic Type Conversion**

  While doing operations on two Strings, sometimes it automatically converts them to Number for the desired output.

```
Example :    let x = "10";
             let y = "2";
             let z = x / y ;   // z=5

             x = “5” , y = “2” ;
             z = x*y ;   // z = 10

            x  = “5” , y = “2” ;
            z = x - y ;   // z = 3
```

But “+” concatenates the Strings.

```
Let x = “10” , y =”20”;
Let z  = x + y ; // z=”1020”
```

Methods

1. **tostring( ) :** Converts from Number to String

Example :

```
let x = (25).tostring();
console.log(typeof(x));  // String
 console.log(x);             // 25
```

2. **toFixed( ) :** returns a string, with the number written with a specified number of decimals

Example :

```
let num = 10.126 ;
num.toFixed(0) ;  // 10
 num.toFixed(3) ;  //10.126
```

3. **toPrecision( ) :** returns a string, with a number written with a specified length

Example :

```
let num = 10.58 ;
num.toPrecision(3) ;   // 10.6
num.toPrecision(2) ;   // 11
num.toPrecision(5) ;   //10.580
```

Variables to Numbers

- **Number( ) :** Converts from a variable to number
- **parseInt( ) :** Parses its argument and returns an integer
- **parseFloat( ) :** Parses its argument and returns a floating point number

Number( )

```
Examples :  Number(“10”);    // Returns 10 , string -> number
            Number(“10.5”); // Returns 10.5
            Number(“Coding Ninjas ”); // Returns NaN
```

If the number cannot be converted, NaN (Not a Number) is returned.

parseInt()

- parses a string and returns a whole number, does not contains decimals.
- Spaces are allowed.
- Only the first number is returned.

```
Examples : parseInt("-5");            // returns -5
           parseInt("-5.25");         //returns -5 , whole numbers only
           parseInt("10 20 30");   // returns 10
           parseInt("10 Coding Ninjas");   // returns 10
           parseInt("Coding Ninjas 10");   // returns NaN
```

parseFloat()

- parses a string and returns a number, **decimals are also included**
- Spaces are allowed.
- Only the first number is returned.

```
Examples :     parseFloat("100");        // returns 100
               parseFloat("5.25");       // returns 5.25
               parseFloat("10 20 30");  // returns 10
```

Properties Of Number

Property	Description
MAX_VALUE	Returns the largest number possible in JavaScript
MIN_VALUE	Returns the smallest number possible in JavaScript
POSITIVE_INFINITY	Returns infinity (returned on overflow)
NEGATIVE_INFINITY	Returns negative infinity (returned on overflow)
NaN	Represents a "Not-a-Number" value

Examples : **For maximum and minimum values of Number**

```
let temp = Number.MAX_VALUE  ;
let temp = Number.MIN_VALUE  ;
```

For positive and negative Infinity

```
let temp = Number.POSITIVE_INFINITY  ;
let temp = Number.NEGATIVE_INFINITY  ;
```

For NaN - Not a Number

```
let temp = Number.NaN ;
let temp  = 1 / "Coding Ninjas";  // temp will be NaN
```

Number Properties are not used on variables

- Properties are only on the object **Number**
- Using temp.MAX_VALUE, where temp is a variable, expression, or value, will return **undefined**

Example :

```
let temp  = 5 ;
temp.MAX_VALUE ;   // returns undefined
Number.MAX_VALUE ; // Correct way
```

Boolean

Boolean holds two types of values: **true** or **false**

Boolean( ) Function :

The boolean( ) function is used to find out if an expression (or a variable) is true or not.

Example :

```
Boolean(5>4) ; // returns true
Boolean("Coding Ninjas"> "Coding");  // returns true
```

Predefined Boolean to the data types

Number : If the number is 0, then it is false by default; else, true

```
Boolean(0);  // FALSE
Boolean(1);  // TRUE
Boolean(-2);  // TRUE
```

String: If the String is empty, then it is false by default; else, true

```
Boolean(“”);  // FALSE
Boolean(“Coding Ninjas”);  // TRUE
Boolean(“ ”);  // TRUE because space(_)
```

Undefined : For undefined, it is false by default

```
let temp ;
Boolean(temp);  // FALSE
```

Null : For null, it is false by default

```
let temp = null  ;
Boolean(temp);  // FALSE
```

NaN: For NaN, it is false by default

```
let temp = 1 / “Coding Ninjas”  ;
Boolean(temp);  // FALSE
```

Datatypes that will be covered in the following module :

- **Arrays**
- **Strings**
- **Objects**