Francisco Ramírez Ruiz

OBJETO PREDEFINIDO: NUMBER

15 DE OCTUBRE DEL 2018

ÍNDICE

1.-DEFINICIÓN

El objeto Number permite trabajar con valores numéricos, el cual se crea a partir de la palabra reservada Number() con el parámetro declarado de valor numérico.

Declaración con el valor 10 un objeto Number llamado número.

```
> let numero = new Number(10);
  numero
<- ▶ Number {10}
>
```

Si introducimos un número en forma de cadena lo interpreta como numérico

```
> let numero2 = new Number("10");
  numero2
<- ▶ Number {10}

> numero + numero2
<- 20
```

Sino puede ser convertido a un número devuelve NaN

```
> let numero3= new Number("efcew");
  numero3
<- ▶ Number {NaN}
```

2.-PROPIEDADES

El objeto Number incluye propiedades propias:

- **Number.MAX_VALUE** → Número más grande
- **Number.MIN_VALUE** → Número más pequeño
- **Number.NaN** → No es un número Not a Number
- **Number.NEGATIVE_INFINITY** → Infinitos negativos
- **Number.POSITIVE_INFINITY** → Infinitos positivos
- **Number.prototype** → Permite añadir nuevas propiedades y métodos

```
> Number.NaN
< NaN
> Number.MIN_VALUE
< 5e-324
> Number.MAX_VALUE
< 1.7976931348623157e+308
> Number.NEGATIVE_INFINITY
< -Infinity
> Number.POSITIVE_INFINITY
< Infinity
>
```

3.-MÉTODOS SIN PROTOTYPE

Existen métodos que posee el propio objeto Number sin la necesidad de su propiedad prototype

- **Number.isFinite()** → Determina si el valor pasado es finito
- **Number.isInteger()** → Determina si es de tipo entero
- **Number.isNaN()** → Determina si el valor pasado es NaN

3.1-Number.isFinite()

```
> //Number.isFinite

  function numero(x) {
      if (Number.isFinite(10000 / x)) {
        return 'Finito';
      }
      return 'Infinito!';
    }
< undefined
> numero(9)
< "Finito"
> numero(0)
< "Infinito!"
>
```

3.2-Number.isInteger()

```
>   //Number.isInteger()
    function prueba(a, b) {
      if (Number.isInteger(a / b)) {
        return 'Soy entero!';
      }
      return 'Soy un decimal :(';
    }
< undefined
> prueba(10, 5);
< "Soy entero!"
> prueba(5, 10);
< "Soy un decimal :("
```

3.3-Number.isNaN()

```
//Number.isNaN()
Number.isNaN(Number.NaN); // true
Number.isNaN(0 / 0)       // true
Number.isNaN(NaN);        // true


Number.isNaN(10);         // false
Number.isNaN("10");       // false
Number.isNaN("011.1100"); // false
Number.isNaN(undefined);  // false
Number.isNaN("Holaaa");   // false
Number.isNaN("NaN");      // false
Number.isNaN({});         // false
Number.isNaN(true);       // false
Number.isNaN(null);       // false
```

4.-MÉTODOS CON PROTOTYPE

El objeto Number con la propiedad prototype dispone de otros métodos útiles:

- **Number.prototype.toExponential** → Devuelve una cadena con el número en notación exponencial.
- **Number.prototype.toFixed** → Devuelve una cadena con el número en una notación de punto fijo.
- **Number.prototype.toLocaleString** → Devuelve una cadena que contiene información del número acorde al idioma.
- **Number.prototype.toPrecision** → Devuelve una cadena con el número en una notación de precisión de punto fijo (Redondea)
- **Number.prototype.toString** → Devuelve la cadena del objeto
- **Number.prototype.valueOf** → Devuelve el valor primitivo del objeto

```
let numero = new Number(5.4778);

numero.toExponential(10); // "5.4778000000e+0"
numero.toFixed(7); //.4778000"
numero.toLocaleString('en'); //"5.478" (Redondea)
numero.toLocaleString('es'); //"5,478" (Redondea)
numero.toPrecision(2); // "5.5"
numero.toString(); //"5.4778"
numero.valueOf(); // 5.4778
```

5.-UTILIDADES

5.1-VALORES A VARIABLES NUMÉRICAS

Con el objeto Number podemos asignar valores predefinidos visto anteriormente a otras variables numéricas

```
> numeroGrande = Number.MAX_VALUE;
< 1.7976931348623157e+308
>
```

5.2-MODIFICAR OTROS OBJETOS

Podemos modificar cualquier objeto convirtiéndolo a un valor numérico por ejemplo el objeto Date.

```
> let fecha = new Date('2018,10,09');
  Number(fecha);
< 1539036000000
```

5.3-MODIFICAR EL PROPIO OBJETO NUMBER

Podremos crear una pequeña descripción y asignarla a la propiedad gracias a prototype

```
> numero = new Number(10);
◂ ▶ Number {10}
> Number.prototype.pepito = "";
◂ ""
> numero.pepito = "Soy juanito";
◂ "Soy juanito"
> numero
◂ ▶ Number {10, pepito: "Soy juanito"}
```

5.4-CONVERTIR CADENAS A NÚMEROS

Permite transformar las cadenas numéricas en números incluso los números binarios y hexadecimales.

```
Number('function')     // NaN - Cadena de caracteres
Number('100efewfew')    // NaN - Cadena de caracteres
Number('10')     // 10 - Número entero
Number('10.5')    // 10.5 - Número decimal
Number('10e-3')  // 0.01 - Número exponencial convertido (10^-3)
Number('')        // 0 - Cadena vacia
Number('0x11')    // 17 - Formato hexadecimal
Number('0b11')    // 3 - Formato binario
Number('0o11')    // 9 - Formato octal
```

6.-BIBLIOGRAFíA

https://medium.freecodecamp.org/stanford-just-abandoned-java-in-favor-of-javascript-for-its-intro-cs-course-fe40543e81d8

https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number

https://www.google.es/search?q=w3schools+number&oq=w3scho&aqs=chrome.1.0j69i59j69i60l3j69i57.2778j0j7&sourceid=chrome&ie=UTF-8

https://www.w3schools.com/jsref/jsref_parseint.asp