

Objeto Number

Assim com os demais objetos, a duas formas de declará-lo:

Literal:

```
var x = 123;
```

Formal:

```
var x = new Number('123');
```

No objeto number teremos algumas propriedades estáticas, que nada são que propriedade na qual o valor fica no Objeto construtor e não no objeto construído.

Propriedade MAX_VALUE e MIN_VALUE

Ambas as propriedades são estáticas, elas retornam o maior número que o JS consegue manipular e o menor número que ele consegue manipular.

Método toFixed

Ele é responsável por informar quantos números decimais terá após o '.'.

Exemplo:

```
var x = 1234.8623  
  
x.toFixed(2)  
  
//a função irá retornar 1234.87
```

Método toPrecision

O método toPrecision retorna o número com a quantidade de 'casas' desejadas.

Exemplo:

```
var x = 123456

x.toPrecision(2);

//Será retornado 12
```

Método toString

O método toString pode tanto retornar um número como string como mudar a base do número.

Exemplo de sua utilização:

```
var x = 123

x.toString() //Irá retornar o número, mas como string

x.toString(número) //Caso passe um número como parâmetro, será convertido para
//a base desse número, exemplo a seguir.

x.toString(2) //Irá retornar o número 123, mas em base binária.
```

Método toExponential

O método toExponential serve para trabalhar com notação científica.

Exercício 01

Fazer um programa que calcule a média de dois número e retorne apenas com duas casas decimais.

Resolução:

```
var $inputNota1 = document.querySelector('#nota01');
var $inputNota2 = document.querySelector('#nota02');
var $inputMedia = document.querySelector('#media');
var $btnCalcular = document.querySelector('.btncalcular');

$btnCalcular.addEventListener('click', function(){
    var nota01 = parseFloat($inputNota1.value);
```

```
        var nota02 = parseFloat($inputNota2.value);

        var media = calcularMedia(nota01, nota02);

        $inputMedia.value = media
    })

    function calcularMedia(n1, n2){
        var media = (n1 + n2) / arguments.length;
        media = media.toFixed(2);

        return media;
    }
```