

TUTORIAIS JAVASCRIPT

O Objeto Number

Copyright 2013 – Todos os Direitos Reservados
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

TUTORIAIS JAVASCRIPT

O Objeto Number

Introdução

O objeto **Number** é um objeto wrapper para valores numéricos primitivos. Objetos **Number** são criados com a declaração: **new Number()**, com a seguinte sintaxe:

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

O JavaScript só tem um tipo de número, que pode ter ou não casas decimais. Por exemplo:

```
var pi = 3.14;      // Com decimais;  
var x = 25;        // Sem decimais;
```

Números muito grandes ou muito pequenos podem ser escrito com notação científica. Por exemplo:

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

Todos os números em JavaScript são de 64 bits

JavaScript não é uma linguagem tipada. A despeito de muitas outras linguagens de programação, ela não define muitos tipos de números, como inteiros, short, long, float, etc. Todos os números em JavaScript são armazenados como números de ponto flutuante de 64 bits (8 bytes) na base 10.

Precisão

Os inteiros (números sem casas decimais ou notação exponencial) são considerados de precisão até 15 dígitos. O número máximo de decimais é 17, mas operações aritméticas de ponto flutuante não são 100% precisas.

Vejam um exemplo prático:

Exemplo 1

Nesse exemplo declaramos algumas variáveis de tipos diferentes e algumas expressões para você ter uma ideia de como funcionam os números em JavaScript. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js13-exemplo1.html**.

```
<!DOCTYPE html>  
<html>  
Tutorial JavaScript13 - Exemplo 1 - O Objeto Number  
<head>  
  <title>Tutorial JavaScript13 - Exemplo 1 - O Objeto Number</title><p>  
<script language="javascript">  
var x=12345678901234567890;  
document.write("<p>Apenas 17 dígitos: ");  
document.write(x + "</p>");  
//
```

```

document.write("<p>0.2 + 0.1 = ");
x=0.2+0.1;
document.write(x + "</p>");
//
document.write("<p>Multiplicando e dividindo por 10: <p>");
x=(0.2*10+0.1*10)/10;
document.write(x + "</p>");
</script>
</head>
<body>
</body>
</html>

```

2. O resultado desse código após executado será o seguinte:

```

Tutorial JavaScript13 - Exemplo 1 - O Objeto Number

Apenas 17 dígitos: 12345678901234567000

0.2 + 0.1 = 0.300000000000000004

Multiplicando e dividindo por 10:

0.3

```

Octal e Hexadecimal

JavaScript interpreta constantes numéricas como octal se elas forem precedidas por um zero, e como hexadecimal se forem precedidas por um zero e x. Por exemplo:

```

var y=0377;
var z=0xFF;

```

Obs.: Nunca escreva um número com um zero à esquerda, a menos que você queira convertê-lo para octal.

Vejamos um exemplo prático:

Exemplo 2

Nesse exemplo declaramos algumas variáveis octais e decimais e algumas operações com as mesmas para você ter uma ideia de como funcionam as constantes numéricas octal e hexadecimal em JavaScript. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js13-exemplo2.html**.

```

<!DOCTYPE html>
<html>
Tutorial JavaScript13 - Exemplo 2 - O Objeto Number
<head>
  <title>Tutorial JavaScript13 - Exemplo 2 - O Objeto Number</title><p>
<script language="javascript">
var a=0322;
var b=0xFF;
var c=0200;
var d=0xAA;

```

```
document.write("a é igual a --> " + a + "<br>");
document.write("b é igual a --> " + b + "<br>");
document.write("c é igual a --> " + c + "<br>");
document.write("d é igual a --> " + d + "<br>");
document.write("a+b é igual a --> " + a+b + "<br>");
document.write("c+d é igual a --> " + c+d + "<br>");
</script>
</head>
<body>
</body>
</html>
```

2. O resultado desse código após executado será o seguinte:

Tutorial JavaScript13 - Exemplo 2 - O Objeto Number

```
a é igual a --> 210
b é igual a --> 255
c é igual a --> 128
d é igual a --> 170
a+b é igual a --> 210255
c+d é igual a --> 128170
```

Propriedades e Métodos dos Números

Os números em JavaScript possuem as seguintes propriedades e métodos:

Propriedades:

- MAX_VALUE
- MIN_VALUE
- NEGATIVE_INFINITY
- POSITIVE_INFINITY
- NaN
- prototype
- constructor

Métodos:

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

A propriedade MAX_VALUE

A propriedade **MAX_VALUE** retorna o maior número possível em JavaScript. Essa propriedade estática tem o valor de 1.7976931348623157e+308.

Nota: Números maiores que esse são representados como **infinity**.

A propriedade MIN_VALUE

A propriedade **MIN_VALUE** retorna o menor número possível em JavaScript. Essa propriedade estática tem o valor de 5e-324.

Nota: Números menores que esse são convertidos para 0.

Vejam os exemplos práticos:

Exemplo 3

Nesse exemplo mostramos como utilizar as propriedades **MAX_VALUE** e **MIN_VALUE** do JavaScript. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js13-exemplo3.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript13 - Exemplo 3 - O Objeto Number
<head>
  <title>Tutorial JavaScript13 - Exemplo 3 - O Objeto Number</title><p>
<script language="javascript">
var a = Number.MAX_VALUE;
var b = Number.MIN_VALUE;
document.write("a é igual a --> " + a + "<br>");
document.write("b é igual a --> " + b + "<br>");
</script>
</head>
<body>
</body>
</html>
```

2. O resultado desse código após executado será o seguinte:

```
Tutorial JavaScript13 - Exemplo 3 - O Objeto Number
a é igual a --> 1.7976931348623157e+308
b é igual a --> 5e-324
```

O método toExponential()

Esse método converte um número em uma notação exponencial.

Por exemplo:

```
var num = 5.56789;
var n=num.toExponential()
```

O método toFixed()

Esse método converte um número em uma string mantendo um número específico de decimais.

Nota: Se o número desejado de decimais for maior do que o número atual, zeros são adicionados para criar o comprimento decimal desejado.

Por exemplo:

```
var num = 5.56789;
var n=num.toFixed(2);
```

Vejamos um exemplo prático:

Exemplo 4

Nesse exemplo mostramos como utilizar os métodos **toExponential()** e **toFixed()** do JavaScript. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js13-exemplo4.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript13 - Exemplo 4 - O Objeto Number
<head>
  <title>Tutorial JavaScript13 - Exemplo 4 - O Objeto Number</title><p>
<script language="javascript">
var num = 2.54309;
var a = num.toExponential();
var num = 5.56789;
var b = num.toFixed(2);
document.write("a é igual a --> " + a + "<br>");
document.write("b é igual a --> " + b + "<br>");
</script>
</head>
<body>
</body>
</html>
```

2. O resultado desse código após executado será o seguinte:

```
Tutorial JavaScript13 - Exemplo 4 - O Objeto Number
a é igual a --> 2.54309e+0
b é igual a --> 5.57
```

Exercícios de fixação

- 1) Objetos **Number** são criados com a declaração: **new Number()**. Essa declaração é:
 - a) Verdadeira
 - b) Falsa
- 2) Esse método converte um número em uma notação exponencial.
 - a) toFixed()
 - b) toPrecision()
 - c) toExponential()
 - d) toString()
- 3) Esse método converte um número em uma string mantendo um número específico de decimais.
 - a) toExponential()
 - b) toString()
 - c) toFixed()
 - d) valueOf()
- 4) A propriedade **MAX_VALUE** retorna o maior número possível em JavaScript. Essa declaração é:
 - a) Verdadeira
 - b) Falsa
- 5) Diferentemente de outras linguagens de programação JavaScript não é uma linguagem tipada. Essa declaração é:
 - a) Verdadeira
 - b) Falsa
- 6) A propriedade **MIN_VALUE** retorna o menor número possível em JavaScript, ou seja, 0 (zero). Essa declaração é:
 - a) Verdadeira
 - b) Falsa
- 7) O número máximo de decimais dos inteiros em JavaScript é:
 - a) 15
 - b) 16
 - c) 17
 - d) 18
- 8) Todos os números em JavaScript são de ____ bits:
 - a) 8
 - b) 16
 - c) 32
 - d) 64
- 9) Qual o valor da variável **c** no código:

```
var a=0322;  
var b=0xFF;  
var c=a-b;
```

- a) 45
- b) 54
- c) -45
- d) -54

10) Qual o valor das variáveis **a** e **b** no código:

```
var num = 2;  
var a = num.toFixed(3);  
var num = 5.89;  
var b = num.toFixed(5);
```

- a) 2.000 e 5.89000
- b) 2000 e 589000
- c) 2.000 e 58900.0
- d) 0.002 e 5.89000

JORGE EIDER