



# Programa o Estruturada

## Aula 06 - Operadores JavaScript: Aritm tica

### Videoaula 07 - M todos do objeto Math



Material Did tico do Instituto Metr pole Digital - IMD  
Vers o 5.3 - Todos os Direitos reservados



## Videoaula 07 - Métodos do objeto Math

---

Outro objeto muito útil para trabalharmos com números é o objeto `Math`. Ele também possui vários métodos e propriedades interessantes. Nesta videoaula, vamos conhecer algumas delas em detalhes. São elas:

1. `ceil(x)`: retorna  $x$ , arredondado para cima para o número inteiro mais próximo
2. `floor(x)`: retorna  $x$ , arredondado para baixo para o número inteiro mais próximo
3. `trunc(x)`: retorna a parte inteira de um número  $x$
4. `round(x)`: arredonda  $x$  para o número inteiro mais próximo
5. `pow(x, y)`: retorna o valor de  $x^y$
6. `exp(x)`: retorna o valor de  $E^x$
7. `sqrt(x)`: retorna a raiz quadrada de  $x$
8. `cbrt(x)`: retorna a raiz cúbica de  $x$
9. `abs(x)`: retorna o valor absoluto de  $x$
10. `min(x, y, z, ..., n)`: retorna o número com o valor mais baixo
11. `max(x, y, z, ..., n)`: retorna o número com o valor mais alto
12. `random()`: retorna um número aleatório entre 0 e 1. Esse intervalo é fechado em 0 e aberto em 1, ou seja, ele contém números maiores ou iguais a zero e menores que um. Por esse motivo, esse método pode retornar o número 0, mas nunca retornará o número 1.

Além desses métodos, o objeto `Math` tem vários outros métodos, como `log(x)`, que retorna o logaritmo natural, isto é, na base  $E$ , de  $x$ , e métodos da trigonometria, como seno, cosseno, tangente e outros. No entanto, não os abordaremos nesta disciplina.

Além dos métodos, o objeto `Math` também possui oito constantes matemáticas. São elas:

1. `Math.E`: constante igual ao número de Euler
2. `Math.PI`: constante  $\pi$
3. `Math.SQRT2`: raiz quadrada de 2
4. `Math.SQRT1_2`: raiz quadrada de  $1/2$
5. `Math.LN2`: logaritmo natural de 2
6. `Math.LN10`: logaritmo natural de 10
7. `Math.LOG2E`: logaritmo base 2 de  $E$
8. `Math.LOG10E`: logaritmo base 10 de  $E$

Agora, você verá o uso dos métodos do objeto `Math` e de algumas de suas constantes na prática.

Então, vamos agora conhecer os métodos e constantes que temos no objeto `Math`. Temos a constante `PI`, vamos atribuí-la à variável `numero` o valor dessa constante, que a gente referencia como `Math.PI` (linha 15), e aí eu estarei sempre escrevendo o valor dessa variável `numero` nesse campo "texto" (linha 12), você pode conferir isso rolando a tela para baixo e observando o valor desse número no campo de texto (linha 56), tá certo?

#### Código 1 - 06\_8 Métodos de Math.html

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 06</title>
```

```
5 </head>
6 <body>
7   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9   <h1>Métodos de Math</h1>
10  <p id="texto"></p>
11
12  <script>
13    var numero = Math.PI;
14
15    // Arredondamentos
16
17    //numero = Math.ceil(numero);
18    //numero = Math.floor(numero);
19    //numero = Math.trunc(numero);
20    //numero = Math.round(numero);
21    //numero = Math.round(3.49);
22    //numero = Math.round(3.5);
23
24    // Potenciação e Raiz
25
26    //numero = Math.pow(5,2);
27    //numero = 5**2;
28    //numero = Math.exp(3);
29    //numero = Math.E**3;
30    //numero = Math.sqrt(25);
31    //numero = Math.cbrt(27);
32    //numero = Math.sqrt(Math.pow(5,2));
33
34    // Valores absolutos, mínimos, máximos
35    //numero = Math.abs(-100);
36    //numero = Math.min(-5,2,100,-10);
37    //numero = Math.min();
38    //numero = Math.min("olá mundo");
39    //numero = Math.max(-5,2,100,-10);
40    //numero = Math.max(3.14, 3.1415);
41    //numero = Math.max();
42    //numero = Math.max("olá mundo");
43
44    // Valores randômicos
45    //numero = Math.random();
46
47    /*
48    function randomInteger(min, max) {
49      return Math.floor(Math.random() * (max - min + 1) ) + min;
50    }
51    numero = randomInteger(0,100);
52    */
```

```
53
54     document.getElementById("texto").innerHTML = numero;
55
56     </script>
57     </body>
58 </html>
59
```

Então, você pode ver no HTML que foram utilizadas as 15 casas decimais para referenciar a constante `PI` e agora vou aplicar várias operações a essa constante, a esse valor de `PI`.

Vamos começar com `Math.ceil`, que vai arredondar esse número para 4, ou seja, ele levou o número 3.1415, aquele número que a gente tinha visto, ele arredondou para cima, para o primeiro inteiro acima dele. E se eu usar o `Math.floor`, ele vai arredondar para baixo, então ele vai levar de 3.14 para 3, o primeiro inteiro abaixo de 3.1415, que é aquela constante `PI` que visualizamos anteriormente.

### Código 2 - 06\_8 Métodos de Math.html

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 06</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Métodos de Math</h1>
10    <p id="texto"></p>
11
12    <script>
13      var numero = Math.PI;
14
15      // Arredondamentos
16
17      numero = Math.ceil(numero);
18      //numero = Math.floor(numero);
19      //numero = Math.trunc(numero);
20      //numero = Math.round(numero);
21      //numero = Math.round(3.49);
22      //numero = Math.round(3.5);
23
24      document.getElementById("texto").innerHTML = numero;
```

```
25
26 </script>
27 </body>
28 </html>
29
```

Uma outra operação que a gente pode fazer é o `Math.trunc`, o que ele faz também é "truncar", é simplesmente esquecer toda a parte decimal. E agora o `Math.round`, que é um método de arredondamento, ele é diferente, vai levar para o inteiro mais próximo, então se ele utilizar aquele  $\pi$ , vai para 3, porque 3.1415 está mais próximo de 3.

Vamos fazer um outro exemplo, com o `round` de 3.49. Se aplicarmos o arredondamento usando o `round`, o que temos também é o 3, porque 3.49 está mais próximo de 3. Porém, se nós aplicarmos o mesmo método a 3.5 (linha 19), o que teremos agora é o valor 4, porque ele considera que o 3.5 está mais próximo do inteiro 4, ok?

Então, isso é o que temos sobre arredondamento, vamos agora ver os métodos para potenciação e raízes. Temos o método `Math.pow`, que eleva o primeiro argumento ao segundo argumento, veja agora o método `pow` sendo aplicado aos argumentos 5 e 2, o que vai dar exatamente o valor de 5 elevado a 2, e o esperado você pode ver na tela, que é o valor 25. Na verdade, temos outra maneira de fazer isso, que é usando `**`, que já vimos nesta aula, então também podemos usar o `5**2`, que nos dará o mesmo resultado, 25, que é o resultado de 5 elevado a 2.

O `Math.exp` é a exponenciação, e se usarmos o número 3, ele vai resultar no E elevado a 3, que vai gerar no HTML um número com 15 casas decimais. Esse número também pode ser gerado sem usar o método `exp`, podemos usar uma outra constante que temos no objeto `Math`, que é o E, exatamente a constante de Euler, elevando isso a 3 usando `**` (`E**3`) o valor vai ser o mesmo porque essa expressão na linha 19 é exatamente igual à expressão da linha 20, ou seja, você pode usar o método `exp` e elevar a 3, ou a constante E elevado a 3, ok?

O método `Math.sqrt` extrai a raiz quadrada do número que você passar como argumento, então se você passou 25 como argumento, qual é a raiz quadrada de 25? Nós sabemos que é o número 5, que é o que veremos na tela do HTML. O método `Math.cbrt` é a raiz cúbica do número que você passar como argumento, então se você passou 27 como argumento, nós temos 3, afinal de contas, 3 elevado a 3 é 27, ou seja, a raiz cúbica de 27 é o número 3.

E, obviamente, o que a gente pode esperar é que, se nós elevarmos um número qualquer, nesse caso o 5 a 2, e tirar a raiz quadrada dele, o que esperamos é o número de volta, tá? Então, se a gente fizer um `Math.sqrt` do `Math.pow` de 5 elevado a 2, o que teremos é o número 5, ou seja, o número original.

### Código 3 - 06\_8 Métodos de Math.html

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 06</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Métodos de Math</h1>
10    <p id="texto"></p>
11
12    <script>
13      var numero = Math.PI;
14
15      // Potenciação e Raiz
16
17      numero = Math.pow(5,2);
18      //numero = 5**2;
19      //numero = Math.exp(3);
20      //numero = Math.E**3;
21      //numero = Math.sqrt(25);
22      //numero = Math.cbrt(27);
23      //numero = Math.sqrt(Math.pow(5,2));
24
25      document.getElementById("texto").innerHTML = numero;
26
27    </script>
28  </body>
29 </html>
```

Agora, veja alguns métodos que temos para valores absolutos, valores mínimos e valores máximos. Para se observar o valor absoluto de um número é necessário, simplesmente, ignorar o sinal dele, então se você aplicar o `Math.abs`, de absoluto, ao `-100`, o que teremos é o `100`, ele simplesmente ignora o sinal que é igual, obviamente, ao `Math.abs` de `100`, que, se nós ignorarmos o sinal positivo, também teremos o `100`.

#### Código 4 - 06\_8 Métodos de Math.html

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 06</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Métodos de Math</h1>
10    <p id="texto"></p>
11
12    <script>
13      var numero = Math.PI;
14
15      // Valores absolutos, mínimos, máximos
16      numero = Math.abs(-100);
17      //numero = Math.min(-5,2,100,-10);
18      //numero = Math.min();
19      //numero = Math.min("olá mundo");
20      //numero = Math.max(-5,2,100,-10);
21      //numero = Math.max(3.14, 3.1415);
22      //numero = Math.max();
23      //numero = Math.max("olá mundo");
24
25      document.getElementById("texto").innerHTML = numero;
26
27    </script>
28  </body>
29 </html>
30
```

No método `Math.min` você vai passar vários valores, e ele vai retornar o menor valor desses valores que você passar. Em nosso exemplo, temos que o menor valor é o `-10`, que é exatamente o que temos na página HTML quando imprimimos esse valor.



## Código 5 - 06\_8 Métodos de Math.html

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 06</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Métodos de Math</h1>
10    <p id="texto"></p>
11
12    <script>
13      var numero = Math.PI;
14
15      // Valores absolutos, mínimos, máximos
16      //numero = Math.abs(-100);
17      numero = Math.min(-5,2,100,-10);
18      //numero = Math.min();
19      //numero = Math.min("olá mundo");
20      //numero = Math.max(-5,2,100,-10);
21      //numero = Math.max(3.14, 3.1415);
22      //numero = Math.max();
23      //numero = Math.max("olá mundo");
24
25      document.getElementById("texto").innerHTML = numero;
26
27    </script>
28  </body>
29 </html>
30
```

Caso você não passe nenhum valor (linha 18) a expectativa é que tenhamos o `Infinity`. O valor `Infinity` é considerado o menor valor como resultado de invocar a operação `min` sem argumentos.

E caso você passe um valor que não seja um número, veja na linha 18 a *string* "olá mundo", o que nós temos é `NaN`. O menor número, na verdade, não é um número, esse valor que temos é o resultado `Not a Number` ou o `NaN`.

Agora, vamos para o outro lado, vamos para o `Math.max`. Nesse caso, retorna-se o maior valor dos argumentos passados e o maior valor é 100 (linha 18) e é exatamente o que é impresso no HTML ao utilizar esses argumentos.

#### Código 6 - 06\_8 Métodos de Math.html

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 06</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Métodos de Math</h1>
10    <p id="texto"></p>
11
12    <script>
13      var numero = Math.PI;
14
15      // Valores absolutos, mínimos, máximos
16      //numero = Math.abs(-100);
17      //numero = Math.min(-5,2,100,-10);
18      //numero = Math.min();
19      //numero = Math.min("olá mundo");
20      numero = Math.max(-5,2,100,-10);
21      //numero = Math.max(3.14, 3.1415);
22      //numero = Math.max();
23      //numero = Math.max("olá mundo");
24
25      document.getElementById("texto").innerHTML = numero;
26
27    </script>
28  </body>
29 </html>
30
```

Agora, qual seria o maior valor de 3.14 e 3.1415? Bom, o 3.14, na verdade, é 3.1400, que é menor que o 3.1415. Então, o resultado que a gente tem na tela é o esperado, ou seja, o 3.1415.

Se, no entanto, fizermos o mesmo que fizemos com o `min`, ou seja, invocar essa função sem passar nenhum parâmetro (linha 18), agora se antes tínhamos no `min` o `Infinity` e no `max` o `-Infinity`, se nós chamarmos a função `max` sem passar nenhum argumento, nós veremos na tela que o resultado é `-Infinity`.

Por fim, é de se esperar que o mesmo comportamento que tivemos para o `min` seja o comportamento para `max` quando aplicamos isso a *strings*. Então, se aplicarmos isso a "olá mundo", o que teremos é o `NaN`, o `Not a Number`.

Para terminar esta videoaula e esse exemplo, veremos como é que são gerados os números randômicos. Em JavaScript é muito simples, nós simplesmente temos que invocar o método `random` do objeto `Math`, e isso vai gerar números randômicos no intervalo de 0 sendo fechado e até 1 sendo aberto, ou seja, ele gera números aleatórios entre 0, incluindo 0 e 1, sem incluir o 1.

Em outras palavras, esse método pode retornar zeros, mas ele nunca vai retornar 1, vai sempre retornar números menores que 1. Se eu for na página HTML e carregar, cada vez que eu carregar ele vai gerar números aleatórios entre 0 e 1, ok?

Podemos definir uma função usando esse método para gerar números aleatórios entre dois números que nós queremos, ou seja, definimos o intervalo que nós desejamos. Então, por exemplo, vou criar essa função e chamá-la de `randomInteger`, que recebe um mínimo, um valor mínimo e um valor máximo e vai usar essa expressão em destaque (linha 21) para poder retornar números entre esses dois parâmetros que você passar o mínimo e o máximo, vamos aplicar o método de arredondamento para baixo, indicamos a geração do número aleatório, multiplicamos isso pelo máximo menos o mínimo mais 1 ( $\text{max} - \text{min} + 1$ ), arredondamos essa expressão e somamos com o valor mínimo, isso nos leva para dentro do intervalo desejado.

#### Código 7 - 06\_8 Métodos de Math.html

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 06</title>
5   </head>
6   <body>
```

```

7   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9   <h1>Métodos de Math</h1>
10  <p id="texto"></p>
11
12  <script>
13  var numero = Math.PI;
14
15  // Valores randômicos
16  numero = Math.random();
17
18  function randomInteger(min, max) {
19    return Math.floor(Math.random() * (max - min + 1) ) + min;
20  }
21  numero = randomInteger(0,100);
22
23  document.getElementById("texto").innerHTML = numero;
24
25  </script>
26  </body>
27  </html>
28

```

Então, só para testar essa função, o que a gente vai fazer é simplesmente ficar invocando, imprimindo na tela a chamada a essa função com os valores 0 e 100, dentro do intervalo 0 e 100. Ao acessarmos a página, veremos que estão sendo gerados números randômicos entre 0 e 100, incluindo o 100 também. Perceba que o 0 está dentro do intervalo da geração desse método, e o 100 também está dentro do intervalo da geração randômica desse método de números entre 0 e 100. Então, basta eu mudar esses parâmetros e ele vai gerar números aleatórios dentro do intervalo fechado no mínimo e no máximo, de acordo com o desejo do usuário.

Com esse exemplo, concluímos esta aula sobre números em JavaScript, na qual você pôde conhecer como JavaScript trabalha com números e objetos que oferecem funções matemáticas como funções de arredondamento, raiz quadrada, funções de trigonometria e funções de geração de números randômicos.

Agora, a fim de colocar esses conhecimentos em prática, faça as atividades propostas. Nos vemos na próxima aula. Tchau, tchau!!!