

# 1 Operadores e funções matemáticas

---

*Neste capítulo você verá como transformar expressões matemáticas que utilizam operadores aritméticos (+, -, ×, ÷) e funções matemáticas (seno, cosseno, logaritmo etc.) em linhas de programa executáveis pelo computador. Verá também como escrever na tela o resultado de cálculos com a precisão ou o número de algarismos significativos desejado e como colocar comentários (textos ignorados pelo interpretador) no seu programa.*

Abra o seu editor de textos predileto, escreva as seguintes linhas e salve o arquivo com uma extensão `.html`:

## Listagem:

```
<script>  
document.write("Meu primeiro script");  
</script>
```

Para utilizar HTML/CSS/JavaScript, assim como qualquer outra linguagem de programação, você precisa de um editor de textos que salve o documento sem qualquer formatação adicional. O Bloco de Notas (Windows), o Gedit (Linux) e vários outros disponíveis gratuitamente na internet (EditPad, TextPad etc.) são exemplos de editores que fazem isso automaticamente. Se você insistir em utilizar algo como o Word ou o LibreOffice, ao salvar escolha "texto sem formatação" para o tipo de arquivo e coloque explicitamente a extensão HTML no nome do arquivo.

Clique no arquivo para visualizá-lo no navegador. Você deverá ver abrir uma janela do seu navegador padrão com a frase "Meu primeiro script" escrita nela. Pronto, você fez o seu primeiro script em JavaScript!

Os scripts são sempre antecedidos por `<script>` e sucedidos por `</script>`, que podem aparecer múltiplas vezes e em vários pontos de um mesmo documento HTML.

Vamos incrementá-lo um pouco, introduzindo algumas variáveis. Experimente digitar as seguintes linhas, salve o arquivo e visualize-o no navegador:

## Listagem:

```
<script>
var a = 2;
var b = 3;
var c = a + b;
document.write("A soma de " + a + " com " + b + " resulta em " + c);
</script>
```

## Resultado:

A soma de 2 com 3 resulta em 5

O comando `var` é utilizado para declarar variáveis em JavaScript e o ponto-e-vírgula é utilizado para indicar que o comando foi terminado. O JavaScript não vai reclamar se você esquecer de usar o `var` nem o ponto-e-vírgula, mas é altamente recomendável que você sempre os utilize, pois em algumas situações específica sua ausência pode levar a erros difíceis de identificar.

Diferentemente de muitas outras linguagens de programação (C, C++, Java etc.) em JavaScript não se especifica o tipo da variável (inteiro, ponto flutuante (real), booleano etc.). O JavaScript vai decidir e modificar o tipo de variável conforme o contexto em tempo de execução. Mas você deve ter clareza disso e de como usa suas variáveis pois, dependendo do problema, essa característica pode ser o céu ou o inferno, também levando a erros difíceis de identificar.

O método `write()` do objeto `document`, acessado com o operador ponto (`.`), escreve algo no documento HTML quando ele está sendo carregado. A instrução `document.write()` recebe como argumento uma expressão literal, que pode ser formada por constantes, que são colocadas entre aspas simples ou duplas, ou por variáveis ou combinações de constantes e variáveis. No exemplo acima o sinal de mais (`+`) é utilizado para concatenar (juntar) constantes e variáveis em literais.

---

### Sua vez... (1-1)

Modifique o exemplo anterior de modo que defina as variáveis `x0`, `v0`, `a` e `t`, atribuindo-lhes os valores 0 m, 0 m/s, 9,8 m/s<sup>2</sup> e 1 s, respectivamente, e calcule o valor de  $x = x_0 + v_0 \cdot t + a \cdot t^2 / 2$ . Para ter certeza de que seu programa está correto, experimente outros valores e verifique se o resultado é o esperado.

---

Em JavaScript, assim como em virtualmente todas as outras linguagens de programação, o ponto (`.`) e não a vírgula (`,`) é utilizado como separador decimal. Certifique-se que no programa o valor atribuído à aceleração é 9.8 e não 9,8.

O objeto `Math` do JavaScript dispõe várias constantes pré-definidas e operações matemáticas básicas. O script abaixo calcula o logaritmo natural (na base  $e$ ) de  $\pi$  e divide-o pelo seno de  $e$ :

## Listagem:

---

```
<script>
var a = Math.PI;
var b = Math.E;
var c = Math.log(a)/Math.sin(b);
document.write("log(" + a + ")/sin(" + b + ")= " + c);
</script>
```

## Resultado:

```
log(3.141592653589793)/sin(2.718281828459045)= 2.7867137873975154
```

Uma grande fonte de confusão na formatação da informação a ser apresentada é a quantidade e localização das aspas (") e sinais de concatenação (+) na formação da expressão final. Olhe atentamente os exemplos anteriores e certifique-se de que entendeu perfeitamente o uso desses sinais para delimitar e juntar as constantes e as variáveis.

No exemplo acima foi utilizado novamente o operador "." para dar acesso a propriedades (`Math.PI`, `Math.E`) e métodos (`Math.log()`, `Math.sin()`) do objeto `Math`. Esta é a forma geral de acessar propriedades e métodos dos objetos em JavaScript.

O JavaScript, como toda linguagem de programação, oferece todas as funções matemáticas fundamentais. Dentre as funções disponíveis no objeto `Math` estão:

`Math.abs(x)` - valor absoluto (módulo) do argumento.  
`Math.acos(x)` - arco-cosseno do argumento.  
`Math.asin(x)` - arco-seno do argumento.  
`Math.atan(x)` - arco-tangente do argumento.  
`Math.atan2(y, x)` - ângulo cuja tangente é  $y/x$ .  
`Math.ceil(x)` - menor inteiro maior ou igual ao argumento.  
`Math.exp(x)` - retorna  $e = 2.71\dots$  elevado a  $x$ .  
`Math.floor(x)` - maior inteiro menor ou igual ao argumento.  
`Math.max(x, y)` - maior dos dois argumentos.  
`Math.min(x, y)` - menor dos dois argumentos.  
`Math.pow(x, y)` - valor de  $x$  elevado a  $y$ .  
`Math.random(x)` - retorna um número aleatório entre 0 e 1 inclusive.  
`Math.round(x)` - valor arredondado do argumento.

O método `Math.log(x)` retorna o logaritmo natural de  $x$ , que, em matemática, é usualmente representado por  $\ln(x)$ , o que às vezes causa certa confusão. O logaritmo de um número na base 10, pode ser obtido utilizando-se o método `Math.log10(x)` ou a fórmula de mudança de base:

$$\log(x) = \ln(x) / \ln(10)$$

Em um programa, esse cálculo poderia ser implementado como algo do tipo:

```
var x = 2;
var y = Math.log10(x);
var z = Math.log(x)/Math.log(10);
```

## Sua vez... (1-2)

Faça um script que calcule os valores do seno, o cosseno, tangente, raiz quadrada, logaritmo na base  $e$  e logaritmo na base 10 de  $x = 2$ . Confira seus resultados com o que você obtém com uma calculadora de mão.

---

---

O JavaScript tem dois métodos para lidar com a maneira como são apresentados os dígitos de um número, os métodos `Number.toFixed(n)` e `Number.toPrecision(n)`. Os dois métodos transformam um objeto `Number`, que pode ser uma constante ou uma variável, em um literal. No caso do `toFixed(n)`, o número é transformado em um literal com *n* algarismos depois do ponto decimal; no caso do `toPrecision(n)`, em um literal com *n* algarismos significativos. O script a seguir ilustra a sua aplicação e efeitos.

### Listagem:

```
<script>
var x = Math.sqrt(2);
document.write(x.toFixed(0) + "<br>");
document.write(x.toFixed(2) + "<br>");
document.write(x.toFixed(4) + "<br>");
document.write(x.toPrecision(1) + "<br>");
document.write(x.toPrecision(2) + "<br>");
document.write(x.toPrecision(4) + "<br>");
</script>
```

### Resultado:

```
1
1.41
1.4142
1
1.4
1.414
```

---

---

## Sua vez... (1-3)

Experimente ver e analisar o que acontece quando você altera a primeira linha do exemplo anterior para `var x = Math.sqrt(1000*2)` e para `var x = Math.sqrt(0.001*2)`.

---

---

Estes métodos podem ser utilizados com números especificados explicitamente:

### Listagem:

```
<script>
document.write((100).toFixed(0) + "<br>");
document.write((100).toFixed(2) + "<br>");
document.write((100).toFixed(4) + "<br>");
document.write((100).toPrecision(1) + "<br>");
document.write((100).toPrecision(2) + "<br>");
document.write((100).toPrecision(4) + "<br>");
</script>
```

## Resultado:

```
100
100.00
100.0000
1e+2
1.0e+2
100.0
```

Nos exemplos anteriores, você deve ter notado que no final de cada linha a literal "<br>" é adicionada. Trata-se de um marcador (*tag*) do HTML que faz com que o navegador quebre a linha (é uma abreviação de *break*). Experimente removê-lo e veja o que acontece. O documento onde o JavaScript opera e apresenta seus resultados é sempre um documento HTML e veremos aos poucos vários outros marcadores utilizados para sua estruturação e formatação.

Constantes, variáveis e funções podem ser combinadas em fórmulas complexas. O exemplo a seguir implementa a fórmula de Torricelli para o cálculo da velocidade no movimento retilíneo uniformemente variado e a fórmula para o cálculo da força elétrica entre um elétron e um próton separados de 1 nanometro ( $10^{-9}$  m).

## Listagem:

```
<script>
// Equação de Torricelli
// utilizada para calcular a velocidade final
// a partir da velocidade inicial, da aceleração
// e da distância percorrida pelo objeto
var vo = 0;
var a = 9.8;
var Dx = 10;
var v = Math.sqrt(vo*vo + 2*a*Dx);
document.write("<i>v</i> = " + v.toFixed(2) + " m/s <br>");

/*
Lei de Coulomb
utilizada para calcular a força
entre duas cargas elétricas q1 e q2
separadas de uma distância d
*/
var q1 = -1.6e-19;
var q2 = +1.6e-19;
var d = 1e-9;
var F = 8.99e9 * q1 * q2 / Math.pow(d,2);
document.write("<i>F</i> = " + F.toPrecision(3) + " N <br>");
</script>
```

Números muito grandes ou muito pequenos são escritos em potências de 10, como a carga do elétron, que é  $1.6 \times 10^{-19}$  C. Em linguagens de programação potências de 10 são em geral indicadas por um "e" entre a mantissa e o expoente. Nesta notação a carga do elétron fica  $1.6e-19$  e a constante elétrica  $8.99 \times 10^9$  fica  $8.99e9$ . Ao executar o script, você deverá obter os seguintes resultados:

```
v = 14.00 m/s
F = -2.30e-10 N
```

Note o uso de `//` para indicar um comentário. Qualquer texto colocado na linha (e apenas naquela linha) após as duas barras será ignorado pelo interpretador. Este tipo de comentário aplica-se linha a linha. Para comentar diversas linhas sucessivas utiliza-se `/* ... */`. Este esquema de incorporar comentários é o mesmo utilizado em C/C++/Java e outras linguagens delas derivadas.

As seqüências `<i>...</i>` indicam o marcador HTML que coloca o texto entre eles em *itálico*, que é como normalmente aparecem variáveis escalares em textos técnicos e científicos. Os marcadores HTML sempre são identificados por estarem entre "<" e ">" e, com raras exceções, andam aos pares, indicando o início e o fim do texto ou outro elemento marcado.

No caso da equação de Torricelli o quadrado da velocidade inicial foi calculado utilizando a função `Math.pow(x, y)` e no caso da lei de Coulomb o quadrado da distância foi calculado fazendo `(d*d)`. As duas formas são equivalentes. Não esqueça dos parênteses, pois o resultado seria desastroso.

---

---

### Sua vez... (1-4)

Uma das maiores fontes de frustração de novatos em programação e marcação de informação é que os computadores são **absolutamente** intolerantes aos mais pequenos erros de digitação. Para acostumar seus olhos, dedos e cérebro a essa intolerância, um bom exercício é reescrever, "brincar", com um código que funciona e ver se ele continua funcionando.

Modifique o exemplo anterior de modo que:

- a maneira de fazer comentários seja invertida (isto é, use `/*...*/` para comentar a equação de Torricelli e `//` para comentar a lei de Coulomb)
- a equação de Torricelli e as variáveis associadas sejam trocadas pela equação horária do MRU e que a variável calculada seja o tempo. Isto é, dados  $x$ ,  $x_0$  e  $v_0$  calcular  $t$  (`var t = (x-x0)/v0`).
- no exemplo da lei de Coulomb sejam dados os valores das cargas ( $q_1$  e  $q_2$ ) e da força ( $F$ ), e o programa calcule o valor da distância ( $d$ ).

---

---

A expressão `Math.pow(v0, 2) + 2*a*Dx` é passada como argumento para a função `Math.sqrt()`. Quando muitos aninhamentos são necessários, eventualmente fica mais claro dividir o cálculo de expressões como estas em várias linhas, evitando confusões no número de parênteses. Assim, a expressão:

```
var x = Math.pow(Math.sqrt(Math.sin(Math.atan(a*a+b*b))), 1/2);
```

Pode ser reescrita equivalentemente como:

```
var tmp1 = a*a + b*b;
var tmp2 = Math.atan(tmp1);
var tmp3 = math.sin(tmp2);
```

```
var tmp4 = Math.sqrt(tmp3);  
var x = Math.pow(tmp4, 1/2)
```

---

### Sua vez... (1-5)

Faça um programa que, dada a equação de 3o. grau:

$$x^3 + a_1x^2 + a_2x + a_3 = 0$$

calcule suas soluções:

$$x_1 = S + T - \frac{1}{3}a_1$$

$$x_2 = -\frac{1}{2}(S + T) - \frac{1}{3}a_1 + \frac{1}{2}i\sqrt{3}(S - T)$$

$$x_3 = -\frac{1}{2}(S + T) - \frac{1}{3}a_1 - \frac{1}{2}i\sqrt{3}(S - T)$$

onde:

$$Q = \frac{3a_2 - a_1^2}{9}$$

$$R = \frac{9a_1a_2 - 27a_3 - 2a_1^3}{54}$$

$$S = \left(R + \sqrt{Q^3 + R^2}\right)^{1/3}$$

$$T = \left(R - \sqrt{Q^3 + R^2}\right)^{1/3}$$

Confira os resultados verificando se:

$$-a_1 = x_1 + x_2 + x_3$$

$$a_2 = x_1x_2 + x_2x_3 + x_3x_1$$

$$-a_3 = x_1x_2x_3$$

Note que  $i = \sqrt{-1}$  aparece explicitamente em  $x_2$  e  $x_3$  e que se  $Q^3 + R^2$  for negativo, sua raiz quadrada é imaginária. Isso não deixa o problema exatamente mais "difícil", mas mais trabalhoso, pois as partes reais e imaginárias dos cálculos precisam ser tratadas separadamente.

---

### Exercícios

---

Escreva scripts que calculem o valor das seguintes expressões com a precisão indicada na resposta:

1. Equação de movimento para o MRU:

$$x = x_0 + v_0 t + (1/2)at^2$$

para  $x_0 = 3.00$  m,  $v_0 = 5.00$  m/s,  $a = -9.80$  m/s<sup>2</sup>,  $t = 2.00$  s. *Resposta:* -6.60.

2. Amplitude de uma onda transversal:

$$y = y_0 \cos(2\pi(x/\lambda - ft))$$

para  $y_0 = 1.00 \times 10^{-2}$  m,  $f = 1.00 \times 10^5$  Hz,  $\lambda = 0.033$  m,  $x = 2.00$  m,  $t = 2.00$  s.  
*Resposta:* -0.0079.

3. Campo elétrico de uma carga em um ponto do espaço:

$$E_x = kqx/(x^2 + y^2 + z^2)^{3/2}$$

$$E_y = kqy/(x^2 + y^2 + z^2)^{3/2}$$

$$E_z = kqz/(x^2 + y^2 + z^2)^{3/2}$$

para  $k = 8.99 \times 10^9$  N·m<sup>2</sup>/C,  $q = 1.602 \times 10^{-19}$  C e  $x = 1.00$  nm,  $y = 2.00$  nm e  $z = 3.00$  nm (1 nm =  $1 \times 10^{-9}$  m). *Resposta:*  $E_x = 2.75 \times 10^7$  N/C,  $E_y = 5.50 \times 10^7$  N/C,  $E_z = 8.25 \times 10^7$  N/C.

4. Distribuição normal ou gaussiana:

$$g = Ae^{-(x-x_0)^2/(2\sigma^2)}$$

para  $A = 10$ ,  $x_0 = 5.0$ ,  $\sigma = x_0/4$ ,  $x = 2.5$ . *Resposta:* 1.4.

5. Função de onda do estado fundamental de átomos com um elétron:

$$\Psi = (1/\sqrt{\pi})(Z/a_0)^{3/2}e^{-Zr/a_0}$$

para  $Z = 2$ ,  $a_0 = 5 \times 10^{-11}$  m,  $r = 2a_0$ . *Resposta:*  $8 \times 10^{13}$  m<sup>-3/2</sup>.