

01

## Calculando IMC

### Transcrição

Começando deste ponto? Você pode fazer o [DOWNLOAD \(https://github.com/alura-cursos/logica-de-programacao-I/archive/aula-5.zip\)](https://github.com/alura-cursos/logica-de-programacao-I/archive/aula-5.zip) do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

---

## Índice de Massa Corporal

O IMC é o Índice de Massa Corporal, é um indicador do grau de obesidade de uma pessoa. Pode ser o primeiro sinal de que alguém está muito gordo ou muito magro. A fórmula do IMC é simples: peso dividido pelo quadrado da altura.

Será que conseguimos traduzir isso em JavaScript? Para construir o código que traduza a fórmula do IMC em nosso novo programa, criaremos uma cópia do arquivo `programa.html` e daremos um novo nome a ele, chamaremos de `imc.html`. Vamos manter as funções `pulaLinha()` e `mostra()` dentro do novo arquivo. Lembrando que todo programa que criarmos a partir de agora deve seguir uma estrutura mínima, assim será possível aproveitar o que já fizemos:

```
<meta charset="UTF-8">

<script>

function pulaLinha() {

    document.write("<br>");
    document.write("<br>");
}

function mostra(frase) {

    document.write(frase);
    pulaLinha();
}

</script>
```

Para calcular o IMC é preciso fazer o quadrado da altura e multiplicar esse valor por ele mesmo e depois utilizar o resultado e dividir pelo peso:

```
var imcFlavio = pesoFlavio / (alturaFlavio * alturaFlavio);
```

Os parênteses são necessários para que primeiro seja realizada a multiplicação e em seguida a divisão.

Mas onde estão definidas as variáveis peso e altura? Ainda não estão! É comum você ver códigos de exemplo escritos dessa forma, isto é, sem estarem completos. Se você tentar rodar um código assim, o JavaScript reclamará de que `alturaFlavio` e `pesoFlavio` não existem. Experimente!

Então, é preciso atribuir valores e imprimir o resultado do `imcFlavio`:

```
var pesoFlavio = 73;
var alturaFlavio = 1.71;
var imcFlavio = pesoFlavio / (alturaFlavio * alturaFlavio);
mostra("O IMC de Flávio é " + imcFlavio);
```

Como estamos usando a função `mostra`, ela precisa ser declarada no começo do arquivo. Lembre-se que também precisamos utilizar as tags `<script>` e `</script>` envolvendo o código JavaScript, caso contrário, ele será mostrado no navegador, mas sem outros efeitos.

Fique tranquilo! Você rapidamente se ambientará aos procedimentos necessários para construir um código! Na revisão existe uma seção, na qual o aluno poderá repetir o código que estamos escrevendo aqui fazendo o passo a passo. Agora, é importante que o aluno preste atenção e depois que a matéria estiver consolidada, parta para a prática.

Salve o arquivo `imc.html` e abra-o no navegador. Note que pela primeira vez utilizamos um número que possui casas decimais, o `1.71`. No JavaScript, usamos o ponto como delimitador da parte inteira e não a vírgula, como estamos acostumado.

Mas, e se precisarmos calcular o IMC de outra pessoa, de um amigo? Podemos criar outra variável, a `imcAmigo`, que será responsável por armazenar o resultado. Para isso, precisamos dos valores do `pesoAmigo` e `alturaAmigo`. Nossa código ficaria assim:

```
var pesoFlavio = 73;
var alturaFlavio = 1.71;
var imcFlavio = pesoFlavio / (alturaFlavio * alturaFlavio);
mostra(" O IMC de Flávio é " + imcFlavio);

var pesoAmigo = 68;
var alturaAmigo = 1.72;
var imcAmigo = pesoAmigo / (alturaAmigo * alturaAmigo);
mostra("o IMC de meu amigo é: " + imcAmigo);
```

O que o código acima tem em comum com o código anterior é a fórmula que continua a ser a mesma. E se em algum momento e por algum motivo, a [OMS \(\[https://pt.wikipedia.org/wiki/Organiza%C3%A7%C3%A3o\\\_Mundial\\\_da\\\_Sa%C3%BAde\]\(https://pt.wikipedia.org/wiki/Organiza%C3%A7%C3%A3o\_Mundial\_da\_Sa%C3%BAde\)\)](https://pt.wikipedia.org/wiki/Organiza%C3%A7%C3%A3o_Mundial_da_Sa%C3%BAde) resolver mudar a maneira como se calcula o IMC? Será preciso alterar o cálculo em todos os pontos em que ele é utilizado!

Quando um código é repetido diversas vezes, devemos sempre pensar na possibilidade de criar uma função. Utilizar funções facilita o desenvolvimento de um código, portanto, vamos declarar mais uma que será responsável por calcular o IMC. E, assim como a função `mostra`, essa nova precisará receber parâmetros para calcular o IMC. São necessários dois, o `peso` e a `altura`, que declararemos dentro de parênteses e usando uma vírgula como separadora. Observe como ficará o código:

```
function calculaImc(altura, peso) {
  var imc = peso / (altura * altura);
  mostra("IMC calculado é " + imc);
}
```

Repare que declaramos uma variável `imc` dentro da nova função. Dessa forma, essa variável só existe aí dentro e não pode ser acessada do lado de fora das chaves. É o que chamamos de **escopo** da variável. O escopo de uma variável pode ser **local** ou **global**, nesse caso, como estamos lidando com uma variável existente e acessível somente dentro da função dizemos que seu

escopo é local. Apesar de parecer estranho, essa é uma maneira útil de organizar o código. Afinal, quem vai usar a função não precisa e nem quer saber como o cálculo é feito. Com a função declarada, já podemos utilizá-la e o código ficará:

```
var pesoFlavio = 73;  
var alturaFlavio = 1.71;  
calculaImc(alturaFlavio, pesoFlavio);
```

```
var pesoAmigo = 68;  
var alturaAmigo = 1.72;  
calculaImc(alturaAmigo, pesoAmigo);
```

É possível passar os valores da altura e peso diretamente para a função, sem a necessidade de criar variáveis para isso:

```
calculaImc(1.71, 73);  
calculaImc(1.72, 68);
```

Salve seu arquivo `imc.html` e veja o resultado!

