

03

## Criando funções

### Transcrição

Outra boa prática que poderemos implementar no código é separar os blocos de código em blocos menores, em **funções**. Mostramos como um bloco grande de código é responsável por fazer muitas coisas.

No arquivo `calcula-imc.js`, temos um `for` responsável por pegar os dados dos pacientes, fazer a validação do peso e da altura e, depois, calcular o IMC e inserir os dados na tabela. Ou seja, temos três funcionalidades dentro do mesmo bloco.

Essas funcionalidades, como o cálculo do IMC, são necessárias em outros pontos do nosso código no `form.js`. Quando estamos adicionando um paciente na tabela por meio do formulário, atualmente, a linha fica sem o resultado do cálculo do IMC.

Ou seja, precisaremos calcular o IMC também no `form.js`, não só no código que atualmente está preso no `calcula-imc.js`. Queremos **reaproveitar** o código, e isto será possível se chamarmos uma função que calcule o IMC em `form.js`. A função `calculaImc()` receberá `peso` e `altura` como parâmetros:

```
nomeTd.textContent = nome;
pesoTd.textContent = peso;
alturaTd.textContent = altura;
gorduraTd.textContent = gordura;
imcTd.textContent = calculaImc(peso, altura);
```

A função também poderá ser chamada em `calcula-imc.js`, reaproveitando assim a lógica de cálculo do IMC. Vamos criá-la em seguida, e ela já retornará o IMC com duas casas decimais:

```
function calculaImc(peso, altura) {
  var imc = 0;

  imc = peso / (altura * altura);

  return imc;
}
```

Depois, vamos chamá-la também dentro da validação do `peso` e `altura` que antecede a função `calculaImc()`:

```
if ( alturaEhValida && pesoEhValido) {
  var imc = calculaImc(peso, altura);
  tdImc.textContent = imc.toFixed(2);
}

function calculaImc(peso, altura){
  var imc = 0;

  imc = peso / (altura * altura);
```

```
    return imc;
}
```

No browser, veremos que o cálculo será realizado e o resultado será exibido na tabela.

Nome	Peso(kg)	Altura(m)	Gordura Corporal(%)	IMC
Paulo	100	2.00	10	25.00
João	80	1.72	40	27.04
Erica	54	1.64	14	20.08
Douglas	85	1.73	24	28.40
Tatiana	48	1.55	19	19.98

Mas faremos um pequeno ajuste no código: em vez de passarmos `toFixed()` em `if`, iremos movê-lo para o retorno da função `calculaImc()`. Com as alterações, o nosso código fica assim:

```
if ( alturaEhValida && pesoEhValido ) {
    var imc = calculaImc(peso, altura);
    tdImc.textContent = imc;
}
}

function calculaImc(peso, altura){
    var imc = 0;

    imc = peso / (altura * altura);

    return imc.toFixed(2);
}
```

A vantagem de termos uma função separada é que conseguimos executar o `calculaImc()` no arquivo `form.js`:

```
nomeTd.textContent = nome;
pesoTd.textContent = peso;
alturaTd.textContent = altura;
gorduraTd.textContent = gordura;
imcTd.textContent = calculaImc(peso,altura);
```

Pode-se afirmar que o conteúdo de texto do `imcTd` será o retorno da função `calculaImc()`. Neste caso, também teremos `peso`, `altura`, e queremos calcular o IMC. Em seguida, a variável `pacienteTr` também receberá o `imcTd` quando formos adicioná-la na `tr`.

```
pacienteTr.appendChild(nomeTd);
pacienteTr.appendChild(pesoTd);
pacienteTr.appendChild(alturaTd);
pacienteTr.appendChild(gorduraTd);
pacienteTr.appendChild(imcTd);
```

Agora, ao adicionarmos um paciente na tabela pelo formulário, uma nova linha será inserida, com o valor do IMC calculado e exibido na tabela. Para exemplificar, vamos experimentar cadastrar os dados do paciente Alvaro .

Com isto, conseguiremos extrair a responsabilidade de se calcular o IMC, que estava presa no `calcula-imc.js`, fazendo o reuso desse código. Logo, é uma boa prática extraírmos um código que queremos reusar em outros locais, para funções externas.

Nome	Peso(kg)	Altura(m)	Gordura Corporal(%)	IMC
Paulo	100	2.00	10	25.00
João	80	1.72	40	27.04
Erica	54	1.64	14	20.08
Douglas	85	1.73	24	28.40
Tatiana	48	1.55	19	19.98
Pedro	100	2.0	20	25.00
Alvaro	60	1.60	5	23.44

Nós conseguimos reutilizar o código para se calcular o IMC e, após movermos a função, separamos a responsabilidade de se fazer este cálculo. Uma boa prática de programação é extraírmos ações que queremos reutilizar - como o cálculo do IMC - para funções externas. Fica mais essa dica de melhorias do código!