

```
1  /*
2    01
3
4    - Declare uma constante que recebe o seu nome;
5    - Ela deve ter um escopo global.
6  */
7
8  const name = "Marcelo"
9
10 for (let i = 0; i < 5; i++) {
11   console.log(`dentro do bloco de codigo: ${name}`)
12 }
13
14 console.log(`Fora do bloco de codigo: ${name}`)
15
16 /*
17    02
18
19    - Funções também são blocos de código;
20    - Crie uma função que ao ser invocada, exibe sua idade no console;
21    - Dentro da função, declare uma let que armazena a sua idade;
22    - Exiba sua idade no console, através da invocação da função;
23    - Do lado de fora da função exiba no console o valor da let que foi
    declarada
24    dentro da função;
25    - Um erro com a mensagem "Uncaught ReferenceError: SUA_VARIÁVEL is not
    defined" será exibido no console;
26    - Você sabe por que isso aconteceu?
27    Sim sei, foi pq a let está no escopo de bloco da função
28  */
29
30
31 const showingAge = function () {
32   let age = 41
33   console.log(age)
34 }
35
36 showingAge()
37
38 //console.log(age)
39
40 /*
41    03
42
43    - Crie um objeto "car" com as seguintes propriedades e métodos:
44      - name, que recebe o nome do carro;
45      - brand, que recebe a marca do carro;
46      - colors, que recebe 3 cores para o carro;
47      - isRunning, que recebe um boolean indicando se o carro está em
    movimento.
48      Inicialmente, deve receber um boolean indicando que o carro não está em
    movimento;
49
50    - run, que é um método que faz o carro andar e retorna a mensagem
    "O NOME_DO_CARRO está em movimento";
51
52    - stop, que é um método que faz o carro parar e retorna a mensagem
    "O NOME_DO_CARRO está parado";
53
54    - getColorsMessage, que é um método que retorna a mensagem
    "O NOME_DO_CARRO está disponível nas cores COR_01, COR_02 e COR_03".
55  */
56
57 let car = {
```

```
58   name: "Fusca",
59   brand: "VW",
60   colors: ["verde", "amarelo", "azul"],
61   isRunning: false,
62   run () {
63     this.isRunning = true
64     return `0 ${this.name} está em movimento.`
65   },
66   stop () {
67     this.isRunning = false
68     return `0 ${this.name} está parado.`
69   },
70   getColorsMessage () {
71     return `${this.name} está disponível nas cores ${this.colors[0]},
72     ${this.colors[1]} e ${this.colors[2]}`
73   }
74 }
75 /*
76   04
77
78   - Faça o carro andar e exiba no console se ele realmente está em movimento.
79 */
80
81 console.log(car.run())
82 console.log(car.isRunning === true)
83
84 /*
85   05
86
87   - Faça o carro parar e exiba no console se ele realmente está parado.
88 */
89
90 console.log(car.stop())
91 console.log(car.isRunning === false)
92
93 /*
94   06
95
96   - Exiba, no console, a mensagem com as cores do carro.
97 */
98
99 console.log(car.getColorsMessage())
100
101 /*
102   07
103
104   - Exiba, no console, a mensagem "0 carro é um MARCA_DO_CARRO
105   NOME_DO_CARRO";
106   - Utilize a notação de colchetes para acessar as propriedades do carro.
107 */
108 console.log(`0 carro é um ${car['brand']} ${car['name']}`)
109
```