

```
1 // ### Java Script - FUNÇÕES ###
2 //Capítulo: 03-02 (FUNÇÕES)
3
4 //FUNÇÃO => DECLARAÇÃO PADRÃO
5 function soma1(num1, num2) {
6     return num1 + num2;
7 }
8
9 //FUNÇÃO => DECLARADA COMO UMA CONSTANTE
10 //A função "function" agora é referenciada pela variável soma2.
11 const soma2 = function(num1, num2) {
12     return num1 + num2;
13 }
14
15 // FUNÇÃO ANÔNIMA - Expressão Lambda
16 // Esta função irá receber como argumento dois números (num1 e num2)
17 // Não precisa da palavra "function"
18 // => (ARROW) indicando que esta função irá levar a operar o que estiver
19 // dentro do corpo { }.
20 const soma3 = (num1, num2) => {
21     return num1 + num2;
22 }
23
24 // CONTINUANDO O CASO DA FUNÇÃO ANÔNIMA
25 // Quando dentro do corpo da função anônima houve apenas uma linha,
26 // conforme no exemplo "soma3", não é necessário montar o corpo da função
27 // { return num1 + num2; } ou seja
28 // pode-se colocar A EXPRESSÃO DE RETORNO, imediatamente a direita da flecha (arrow)
29 // Assim:
30 const soma4 = (num1, num2) => num1 + num2;
31
32 const dobro1 = function(num){
33     return num * 2;
34 }
35
36 //FUNÇÃO ANÔNIMA:
37 const dobro2 = num => num * 2;
38
39 // FUNÇÃO QUE RETORNA "UNDEFINED" - NÃO TEM "RETURN"
40
41 function mostrarPreco(preco){
42     console.log("PREÇO = ", preco)
43 }
44
45 // VARIÁVEIS DEFINIDAS DENTRO DO ESCOPO DA FUNÇÃO NÃO VAZAM ESCOPO
46
47 function areaCirculo(raio){
48     const pi = 3.14;
49     return pi * raio * raio;
50 }
51
52 // FUNCTION HOISTING: Funções no JS são "MOVIDAS" para cima pelo motor do JS.
53
54
55 function teste(x) {
56     console.log("TESTE = " + x);
57 }
58
59
```

```
60 //FUNÇÕES DE ALTA ORDEM: Funções que podem ser passadas como argumento.
61
62 // E TAMBÉM PODEM SER RETORNADAS COMO RETORNO DE FUNÇÕES.
63 // Uma FUNÇÃO pode RETORNAR uma outra FUNÇÃO.
64
65 function triplo(num) {
66     return num * 3;
67 }
68
69 function aplicar(f, num){
70     const result = f(num);
71     console.log("RESULTADO = " + result);
72 }
73
74 aplicar(triplo, 10);
75 aplicar(dobro1, 10);
76
77
78
```