



Folha de Exercícios — Funções (JavaScript)

Algoritmos e Estruturas de Dados | IPCA

Marta Martinho | 2025

Tema. Funções em JavaScript

Objetivo. Consolidar a aprendizagem de **funções** em JavaScript, promovendo a modularização do código, a reutilização de funções, a passagem de argumentos, o uso de valores de retorno e a implementação de funções iterativas e recursivas.

Notas importantes para JavaScript.

- Todas as entradas de `prompt()` são **strings**; converter com `Number()` quando necessário.
- Validar conversões inválidas com `Number.isNaN()`.
- Preferir igualdade estrita: `===` e `!==`.
- Para resultados decimais, usar `toFixed(2)` quando apropriado.

Exercícios (Funções — JavaScript)

1 — Soma de dois valores

Crie um programa que solicite ao utilizador dois valores numéricos e utilize a função `soma(n1, n2)` que devolve a soma dos dois valores introduzidos.

2 — Média de três valores

Implemente a função `media(n1, n2, n3)` que calcula e devolve a média de três valores. Crie um programa que solicite ao utilizador três valores e apresente a respetiva média.

3 — Média de idades (quantidade indeterminada)

Crie um programa que calcule a média de um número indeterminado de idades introduzidas pelo utilizador.

O programa deve:

- reutilizar a função `soma(n1, n2)` implementada no exercício 1;
- utilizar a função `divisao(dividendo, divisor)` para calcular a média das idades.

4 — Impressão de números de 1 até N

Crie uma aplicação que solicite ao utilizador um número e escreva na consola os números de 1 até esse valor, utilizando a função `imprimeNumeros(numero)`.

5 — Conversão de temperaturas (Celsius e Kelvin)

Crie uma aplicação que permita converter temperaturas entre as escalas Celsius e Kelvin. A aplicação recebe:

- a temperatura;
- a escala de conversão pretendida ('C' ou 'K').

Exemplo: as entradas 38 e 'K' significam converter 38 °C para Kelvin.

Implemente as funções:

- `kelvinToCelsius(temp)`
- `celsiusToKelvin(temp)`

Considere as fórmulas:

$$\text{tempC} = \text{tempK} - 273 \quad \text{tempK} = \text{tempC} + 273$$

6 — Par ou ímpar

Crie um programa que solicite um número ao utilizador e indique se esse número é par ou ímpar.

Implemente a função `verificaPar(numero)` que devolve `true` se o número for par e `false` caso contrário.

7 — Verificação de número primo

Um número primo é apenas divisível por 1 e por ele próprio. Os números 0 e 1 não são considerados primos.

Crie um programa que solicite um número ao utilizador e indique se é primo ou não, utilizando a função `verificaPrimo(numero)`.

8 — Primos num intervalo

Crie uma aplicação que receba do utilizador um intervalo de números inteiros `[inicio,fim]` e indique, para cada número desse intervalo, se é ou não primo.

Reutilize:

- a função `verificaPrimo(numero)` do exercício anterior;
- a função `mostra(numero, primo)` para escrever o resultado na consola.

9 — Somatório de 1 até N

Crie uma aplicação que solicite um número natural ao utilizador e mostre o valor do somatório de 1 até esse número.

Implemente a função `calculaSomatorio(numero)` que devolve o resultado do somatório.

10 — Fatorial (iterativo)

Implemente a função `calculaFatorial(numero)` que calcula e devolve o fatorial de um número inteiro positivo.

Exemplo:

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

11 — Fatorial (recursivo)

Implemente a função do exercício anterior de forma recursiva.

12 — Potência (iterativo)

Crie uma aplicação que calcule a potência matemática de um número, a partir da base e do expoente introduzidos pelo utilizador.

Implemente a função `potencia(base, expoente)` de forma iterativa.

13 — Potência (recursivo)

Implemente a função do exercício anterior de forma recursiva.

14 — Inverter um número

Implemente a função `inverteNumero(numero)` que recebe um número inteiro e devolve o número invertido.

Exemplo: 1234 → 4321.

15 — Capicua (palíndromo)

Desenvolva um programa que reutilize a função do exercício anterior para verificar se um número é capicua (palíndromo).

Implemente a função `capicua(numero)` que devolve `true` se o número for capicua e `false` caso contrário.

16 — Cifra de 4 dígitos

Uma empresa pretende cifrar uma sequência de 4 dígitos (`DigDigDigDig`) segundo as seguintes regras:

- inverter os dígitos do número;
- substituir cada dígito `Dig` por $(\text{Dig} + 8) \bmod 10$.

Implemente:

- uma função que receba um número de 4 dígitos e devolva o valor cifrado;
- uma função que receba um número cifrado e o decifre, devolvendo o valor original;
- um pequeno menu de escolha (por exemplo com `prompt` e `switch`) que permita ao utilizador cifrar ou decifrar um número.