

Sessão 07 (30/01/2026) – Módulo 02 – Fundamentos JavaScript – Aula 03 –

=====

Exercício Prático: Sistema de Validação e Processamento de Pedidos

=====

Objetivo: Trabalhar com criação de funções, parâmetros, return, funções com arrays e objetos, e composição de funções.

=====

CONTEXTO:

Você está a desenvolver um sistema de processamento de pedidos para uma loja online. Precisa criar várias funções para validar e processar os pedidos dos clientes.

=====

PARTE 1: Criar função simples de validação (Dica Utilize o método includes para análise de strings)

=====

Crie uma função chamada 'validarEmail' que:

- Receba um email como parâmetro
- Retorne true se o email contiver "@" e "."
- Retorne false caso contrário

Resultado esperado: validarEmail("joao@exemplo.com") → true

=====

PARTE 2: Função com múltiplos parâmetros e cálculo

=====

Crie uma função chamada 'calcularPrecoFinal' que:

- Receba: precoBase e percentualDesconto
- Calcule o preço com desconto aplicado
- Retorne o preço final

Exemplo: calcularPrecoFinal(100, 10) → 90 (100 - 10%)

=====

PARTE 3: Função com valor padrão

=====

Crie uma função chamada 'calcularFrete' que:

- Receba: valorPedido e regioao (padrão: "nacional")
- Se região for "nacional": frete = 5€
- Se região for "internacional": frete = 15€
- Se valorPedido >= 50€: frete grátis (retorne 0)
- Retorne o valor do frete

=====

PARTE 4: Função que retorna objeto

=====

Crie uma função chamada 'criarPedido' que:

- Receba: nomeCliente, email, valorTotal
- Retorne um objeto com:
 - id: número aleatório entre 1000-9999 (use `Math.floor(Math.random() * 9000) + 1000`)
 - cliente: nomeCliente
 - email: email
 - valor: valorTotal
 - status: "pendente"
 - data: data atual (use `new Date().toLocaleDateString()`)

=====

PARTE 5: Função que processa array

=====

Crie uma função chamada 'calcularTotalPedidos' que:

- Receba um array de objetos pedido
- Calcule a soma de todos os valores
- Retorne o total

Dica: use um loop for ou reduce

=====

PARTE 6: Função que filtra array

=====

Crie uma função chamada 'filtrarPedidosAcimaDe' que:

- Receba: array de pedidos e um valorMinimo
- Retorne apenas os pedidos com valor >= valorMinimo

Dica: use filter ou um loop com push

=====

PARTE 7: Função que valida objeto

=====

Crie uma função chamada 'validarPedido' que:

- Receba um objeto pedido
- Verifique se tem todas as propriedades: cliente, email, valor
- Verifique se email é válido (use a função validarEmail)
- Verifique se valor > 0
- Retorne true se tudo válido, false caso contrário

=====

PARTE 8: Função que usa outras funções (composição)

=====

Crie uma função chamada 'processarPedido' que:

- Receba: nomeCliente, email, valorProdutos, percentualDesconto, regioao
- Valide o email (use validarEmail)
- Se email inválido, retorne: "Erro: Email inválido"
- Calcule preço com desconto (use calcularPrecoFinal)
- Calcule o frete (use calcularFrete)
- Calcule o total (preço com desconto + frete)
- Crie e retorne o pedido (use criarPedido) com valor total

Esta função COMBINA todas as funções anteriores!

=====

PARTE 9: Arrow Function

=====

Converta a função abaixo para Arrow Function:

```
function dobrar(numero) {  
  return numero * 2;  
}
```

```
}
```

=====

PARTE 10: Desafio Final - Gerar Relatório

=====

Crie uma função chamada 'gerarRelatorio' que:

- Receba um array de pedidos
- Calcule e retorne um objeto com:
 - totalPedidos: quantidade de pedidos
 - valorTotal: soma de todos os valores (use calcularTotalPedidos)
 - ticketMedio: valorTotal / totalPedidos
 - maiorPedido: objeto do pedido com maior valor
 - menorPedido: objeto do pedido com menor valor

=====

RESULTADO ESPERADO (após completar todos os exercícios):

=====

- PARTE 1: Emails validados corretamente
- PARTE 2: Preços com desconto calculados
- PARTE 3: Frete calculado com regras corretas
- PARTE 4: Objeto pedido criado com todas propriedades
- PARTE 5: Total de pedidos calculado (245.50€)
- PARTE 6: Pedidos filtrados (apenas >= 70€)
- PARTE 7: Validações funcionando corretamente
- PARTE 8: Pedido completo processado usando todas as funções
- PARTE 9: Arrow function funcionando
- PARTE 10: Relatório com todas as estatísticas