

Exercícios de TypeScript – B4F

Tópicos abordados:

- ✓ O que é TypeScript
- ✓ Tipos de dados
- ✓ Type Annotation e Type Inference
- ✓ Tipos any, unknown, void
- ✓ Arrays, Objetos
- ✓ Type Aliases
- ✓ Interfaces
- ✓ Enums
- ✓ Union Types
- ✓ Funções com parâmetros e retorno

Parte 1 – Tipos básicos e Type Annotation

1. Crie variáveis com os seguintes tipos: string, number, boolean, null, undefined. Atribua valores e utilize type annotation para cada uma delas.
2. Escreva uma função que recebe dois números e retorna a sua soma. Use type annotation nos parâmetros e no retorno.
3. Escreva uma função que imprime uma mensagem no console. Declare o tipo de retorno como void.
4. Crie uma variável com o tipo any e tente atribuir valores de tipos diferentes. Em seguida, mostre os problemas que podem ocorrer.
5. Declare uma variável com tipo unknown e tente utilizá-la em uma função. Use o type guard (typeof) para garantir segurança antes de usar.
6. Escreva uma função que recebe um valor e imprime se é string, number ou boolean. Use type inference dentro da função.

Parte 2 – Arrays, Objetos, Type Aliases e Interfaces

7. Crie um array de números e uma função que retorna a média desses números. Use anotações corretas para o array e a função.
8. Crie um objeto com propriedades nome, idade e ativo. Crie um type alias para esse objeto e utilize-o na anotação.
9. Use interface para definir a estrutura de um Produto com: nome, preço e categoria. Crie uma função que recebe um produto e retorna uma string formatada com as informações.
10. Crie uma função que recebe um array de objetos do tipo Produto e retorna apenas os produtos com preço acima de 100.
11. Crie um type alias chamado Pessoa com propriedades nome, email e telefone. Depois, crie um array de pessoas e exiba os dados de cada uma no console.

12. 12. Transforme o type alias Pessoa do exercício anterior em uma interface.
13. 13. Combine dois tipos (Pessoa e Endereço) usando & e crie um objeto que contenha os dois.

Parte 3 – Enums, Union Types e Funções

14. 14. Crie um enum com os dias da semana. Escreva uma função que recebe um dia e retorna se é final de semana ou dia útil.
15. 15. Crie um enum com os níveis de acesso: Admin, User, Guest. Crie uma função que recebe o nível e retorna a descrição de permissões.
16. 16. Crie uma função que recebe um parâmetro do tipo string | number. Se for string, retorna o comprimento. Se for number, retorna o quadrado do número.
17. 17. Crie uma função que recebe dois parâmetros opcionais (ambos números). Se os dois forem fornecidos, retorna a soma. Se nenhum, retorna 0.
18. 18. Crie uma função que recebe um array de números e uma função de callback. A função deve aplicar o callback a cada número e retornar um novo array.
19. 19. Crie um type alias ou interface para representar uma ContaBancária. A conta deve ter: número, saldo e método depositar(valor: number).
20. 20. Crie um formulário com tipo interface FormularioContato com campos: nome, email, mensagem. Crie uma função enviarFormulario(form: FormularioContato): void que imprime os dados.