

## Exercícios de Funções em Dart/Flutter

### 1. Função Simples (Sem Parâmetros, Sem Retorno)

- **Exercício:** Crie uma função chamada `exibirBoasVindas` que não recebe nenhum parâmetro e, quando chamada, simplesmente imprime a mensagem "Bem-vindo ao mundo do Flutter!" no console.

### 2. Função com Parâmetros (Sem Retorno)

- **Exercício:** Escreva uma função chamada `saudacaoPersonalizada` que aceita um parâmetro do tipo `String` chamado `nome`. A função deve imprimir no console a mensagem "Olá, [nome]! Tenha um ótimo dia."

### 3. Função com Retorno (Sem Parâmetros)

- **Exercício:** Crie uma função chamada `obterAnoAtual` que não precisa de parâmetros, mas retorna o ano atual como um valor `int`. Chame a função e armazene o resultado em uma variável, depois imprima essa variável.

### 4. Função com Parâmetros e Retorno

- **Exercício:** Desenvolva uma função chamada `somarDoisNumeros` que recebe dois parâmetros do tipo `double`, `a` e `b`. A função deve retornar a soma desses dois números. Teste-a, imprimindo o resultado de `somarDoisNumeros(10.5, 22.3)`.

### 5. Função de Seta (Arrow Function)

- **Exercício:** Converta a tarefa anterior em uma "arrow function". Crie uma função `multiplicar` que aceita dois inteiros e retorna sua multiplicação, tudo em uma única linha usando a sintaxe `=>`.

### 6. Função com Parâmetros Nomeados Opcionais

- **Exercício:** Crie uma função chamada `exibirInfoProduto` que aceite um `String` obrigatório para o nome do produto. Além disso, ela deve aceitar dois parâmetros nomeados opcionais: `preco` (um `double`) e `categoria` (uma `String`). A função deve imprimir as informações do produto, tratando os casos em que o preço ou a categoria não são fornecidos (ex: "Produto: Caneta, Preço: R\$ 2.50, Categoria: Papelaria" ou "Produto: Caderno, Preço: Não informado").

### 7. Função com Parâmetros Posicionais Opcionais

- **Exercício:** Escreva uma função chamada `criarResumo` que aceite três parâmetros `String`: `titulo`, `autor` e um parâmetro posicional opcional `resenha`. A função deve retornar uma `String` formatada. Se a resenha for fornecida, o formato é "[titulo] por [autor]. [resenha]". Se não, o formato é "[titulo] por [autor]".

### 8. Função Anônima (Closure)

- **Exercício:** Crie uma lista de `String` com alguns nomes. Utilize o método `forEach` da lista para imprimir cada nome em maiúsculas. Para isso, passe

uma função anônima para o `forEach` que receba um nome e imprima `nome.toUpperCase()`.

#### 9. Função como Parâmetro (Higher-Order Function)

- **Exercício:** Crie uma função chamada `filtrarLista` que aceite dois parâmetros: uma lista de inteiros (`List<int>`) e uma função de teste. Essa função de teste (que você passará como argumento) deve receber um `int` e retornar um `bool`. A função `filtrarLista` deve retornar uma nova lista contendo apenas os números da lista original que passam no teste.
- **Teste:** Crie uma lista `[1, 2, 3, 4, 5, 6, 7, 8]` e chame `filtrarLista` passando uma função que testa se um número é par.

```
// 1. Sem Parâmetros, Sem Retorno (void)
void funcaoSemNada() {
    print("Executando...");
}

// 2. Com Parâmetros, Sem Retorno (void)
void funcaoComParametros(String texto) {
    print("O texto é: $texto");
}

// 3. Sem Parâmetros, Com Retorno
String funcaoComRetorno() {
    return "Resultado";
}

// 4. Com Parâmetros, Com Retorno
int funcaoCompleta(int a, int b) {
    return a + b;
}
```

```
// 5. Arrow Function (versão curta da nº 4)
```

```
int arrowFunction(int a, int b) => a + b;
```

```
// 6. Parâmetros Nomeados Opcionais
```

```
// { } indicam parâmetros nomeados. '?' ou 'required' define a opcionalidade.
```

```
void funcaoNomeada(String nomeObrigatorio, {int? idade, String? cidade}) {
```

```
    print("Nome: $nomeObrigatorio");
```

```
    if (idade != null) {
```

```
        print("Idade: $idade");
```

```
    }
```

```
}
```

```
// Chamada: funcaoNomeada("Ana", cidade: "São Paulo");
```

```
// 7. Parâmetros Posicionais Opcionais
```

```
// [ ] indicam parâmetros posicionais opcionais.
```

```
void funcaoPosicional(String a, [String? b]) {
```

```
    print("Parâmetro A: $a");
```

```
    if (b != null) {
```

```
        print("Parâmetro B: $b");
    }
}

// Chamada:
funcaoPosicional("Obrigatório"); ou
funcaoPosicional("Obrigatório",
"Opcional");

// 8. Função Anônima
// É uma função sem nome, geralmente
passada como argumento.
List<int> numeros = [1, 2, 3];
numeros.forEach((numero) {
    print(numero * 2);
});

// 9. Função como Parâmetro (Higher-
Order)
// 'Function(int)' define o tipo da
função que será recebida.
void executarAcao(int valor,
Function(int) acao) {
    acao(valor);
}

// Chamada: executarAcao(10, (numero) {
print("O número é $numero"); });
```