

# ATIVIDADE SOBRE VARIÁVEIS, ARRAYS E FUNÇÕES EM JAVASCRIPT

## PARTE 1: PERGUNTAS TEÓRICAS

1. Defina o método map em JavaScript.
2. Qual é a principal diferença entre o método map e o método forEach? 3. Explique como o método map pode ser usado para transformar um array. Dê um exemplo.
4. O que é o escopo de uma variável? Diferencie entre escopo global e escopo local.
5. O que são dados mutáveis e imutáveis? Dê exemplos de cada um. 6. Descreva o que são dados persistentes e dados voláteis. Dê exemplos. 7. Explique a diferença entre tipos de dados primitivos e tipos de dados de referência. 8. O que significa a integridade dos dados? Como ela pode ser garantida?

## RESPOSTAS:

1. O método **map** em JavaScript é utilizado para criar um novo array a partir de um array existente, aplicando uma função de callback a cada elemento do array original. O novo array é do mesmo tamanho que o original, mas os elementos são o resultado da aplicação da função de callback.
2. **map**: Retorna um novo array com os resultados da função aplicada a cada elemento.  
**forEach**: Itera sobre os elementos do array, mas não retorna um novo array. Ele é utilizado principalmente para executar efeitos colaterais (como modificar elementos do DOM ou logar no console).

3. 

```
const numbers = [1, 2, 3, 4];  
const doubled = numbers.map(number => number * 2);  
console.log(doubled); // [2, 4, 6, 8]
```

Nesse exemplo, o método **map** é utilizado para criar um novo array **doubled** onde cada número é o dobro do número original.

4. O escopo de uma variável define onde essa variável está acessível no código. Em JavaScript, existem dois tipos principais de escopo:

- **Escopo global:** Variáveis declaradas fora de qualquer função têm escopo global e podem ser acessadas de qualquer lugar no código.
- **Escopo local:** Variáveis declaradas dentro de uma função têm escopo local e só podem ser acessadas dentro dessa função.

5. **Dados mutáveis:** São aqueles que podem ser alterados após a sua criação. Em JavaScript, objetos e arrays são exemplos de dados mutáveis.  
javascript.

**Dados imutáveis:** São aqueles que não podem ser alterados após a sua criação. Em JavaScript, strings e números são exemplos de dados imutáveis.

6. **Dados persistentes:** São armazenados de forma que permanecem disponíveis mesmo após o término de um programa ou após o desligamento de um dispositivo. Exemplos incluem arquivos em disco, bancos de dados e cookies.

**Dados voláteis:** São armazenados temporariamente e são perdidos quando o programa termina ou o dispositivo é desligado. A memória RAM é um exemplo de armazenamento de dados voláteis.

7. **Tipos de dados primitivos:** São dados que são armazenados diretamente no local de memória atribuído à variável. Exemplos em JavaScript incluem `number`, `string`, `boolean`, `null`, `undefined`, `symbol`, e `bigint`.

**Tipos de dados de referência:** São dados que são armazenados como referências em vez do próprio valor. Isso significa que a variável armazena um ponteiro para o local onde o objeto ou array está na memória. Exemplos incluem `object`, `array`, `function`.

8. Integridade dos dados refere-se à precisão, consistência e confiabilidade dos dados ao longo do seu ciclo de vida. Isso pode ser garantido por meio de boas práticas de gerenciamento de dados, incluindo:

- **Validação de entrada de dados:** Garantir que os dados inseridos sejam válidos e consistentes.

- **Controle de acesso:** Restringir o acesso aos dados a usuários autorizados. ●

**Backups regulares:** Manter cópias de segurança dos dados para evitar perdas. ●

**Transações atômicas:** Em bancos de dados, garantir que as operações sejam completadas inteiramente ou revertidas para evitar dados inconsistentes.