

# Módulo 6 – Arrays e Objetos em JavaScript

## Objetivo do módulo

Aprender a armazenar e organizar múltiplos dados de forma eficiente usando arrays e objetos.

### 1. Arrays

Um **array** é uma lista que armazena múltiplos valores em uma única variável.

Exemplo:

```
let frutas = ["Maçã", "Banana", "Laranja"];
console.log(frutas[0]); // Maçã
console.log(frutas[2]); // Laranja
```

#### 1.1 Operações básicas com arrays

- Adicionar elemento no final: `push()`

```
frutas.push("Uva");
```

- Remover elemento do final: `pop()`

```
frutas.pop();
```

- Adicionar elemento no início: `unshift()`

```
frutas.unshift("Manga");
```

- **Remover elemento do início:** `shift()`

```
frutas.shift();
```

## 1.2 Percorrendo arrays

Usamos laços para acessar todos os elementos.

Exemplo com `for`:

```
for(let i = 0; i < frutas.length; i++) {  
    console.log(frutas[i]);  
}
```

Exemplo com `forEach`:

```
frutas.forEach((fruta) => {  
    console.log(fruta);  
});
```

## 2. Objetos

Um **objeto** é uma coleção de **propriedades**, cada uma com um **nome (chave)** e um **valor**.

Exemplo:

```
let pessoa = {  
    nome: "Ana",  
    idade: 25,  
    cidade: "São Paulo"  
};  
  
console.log(pessoa.nome); // Ana  
console.log(pessoa.idade); // 25
```

## 2.1 Alterando valores

```
pessoa.idade = 26;  
pessoa.cidade = "Rio de Janeiro";
```

## 2.2 Adicionando novas propriedades

```
pessoa.profissao = "Programadora";
```

## 2.3 Percorrendo objetos

Usamos `for...in` para percorrer todas as propriedades:

```
for (let chave in pessoa) {  
    console.log(chave + ": " + pessoa[chave]);  
}
```

## 3. Arrays de objetos

É comum combinar arrays e objetos para organizar informações complexas.

Exemplo:

```
let alunos = [  
    { nome: "Ana", idade: 20 },  
    { nome: "Carlos", idade: 22 },  
    { nome: "Mariana", idade: 21 }  
];  
  
alunos.forEach((aluno) => {  
    console.log(aluno.nome + " tem " + aluno.idade + " anos");  
});
```

## 4. Boas práticas

- Use arrays para **listas de dados relacionados**.
- Use objetos para **representar entidades com características diferentes**.
- Combine arrays e objetos para **estruturas de dados mais complexas**.

## 5. Próximos passos

No próximo módulo você aprenderá sobre **funções avançadas e manipulação de arrays**, incluindo métodos como `map`, `filter` e `reduce`, que tornam seu código mais poderoso e funcional.

Pratique criando arrays e objetos de sua própria rotina, como uma lista de filmes, tarefas ou contatos, e teste percorrendo, alterando e adicionando dados.