

# MATERIAL EXTRA

## Tipos Estruturados



growdev

## TIPOS ESTRUTURADOS

No JavaScript, **tipos estruturados** são formas de **organizar e agrupar dados** de maneira que possam ser facilmente compreendidos e manipulados. Esses tipos ajudam a organizar informações complexas que não poderiam ser representadas de forma simples, como números ou textos isolados. Os dois principais tipos estruturados em JavaScript são os **objetos** e os **arrays**, que funcionam como recipientes para armazenar dados.

### Objetos

Os objetos podem ser comparados a uma "caixa de informações" onde cada item tem um nome específico e um valor associado.

Imagine que você está descrevendo uma pessoa em um papel; você provavelmente anotaria informações como o nome, idade, e profissão. Em JavaScript, cada uma dessas informações seria uma propriedade dentro de um objeto. Por exemplo, a propriedade "nome" poderia ter o valor "Ana", a propriedade "idade" poderia ter o valor "30", e assim por diante. Os objetos permitem que você agrupe essas informações relacionadas em uma estrutura única, facilitando o acesso e a manipulação delas em conjunto.

Eles funcionam como uma **coleção de pares chave-valor**, onde cada chave é uma string (ou símbolo) que identifica uma propriedade, e o valor pode ser de qualquer tipo de dado, incluindo números, strings, arrays, funções e até outros objetos.

Essa capacidade de agrupar dados relacionados torna os objetos extremamente úteis para representar entidades do mundo real, como uma pessoa, um carro, ou até mesmo um produto em uma loja. A estrutura dos objetos facilita a organização dos dados, permitindo acesso e modificação eficientes.

## Criando Objetos

Existem várias maneiras de criar objetos em JavaScript. A forma mais comum é usando a notação de objeto literal. Aqui está um exemplo:

```
const pessoa = {  
  nome: "Carlos",  
  idade: 28,  
  profissao: "Desenvolvedor de Software",  
  possuiCNH: false  
}
```

Neste exemplo, criamos um objeto chamado **pessoa** com várias propriedades: **nome**, **idade**, **profissao**, e **possuiCNH**. Para cada propriedade foi atribuído um valor de tipos distintos de acordo com os dados de uma pessoa.

## Acessando propriedades de um objeto

As propriedades de um objeto podem ser acessadas usando a notação de ponto ou colchetes.

```
console.log(pessoa.nome) // "Carlos"  
console.log(pessoa.idade) // 28  
console.log(pessoa['profissao']) // "Desenvolvedor de Software"  
console.log(pessoa["possuiCNH"]) // false
```

## Modificando Objetos

Os objetos em JavaScript são dinâmicos, o que significa que você pode adicionar, modificar ou remover propriedades a qualquer momento.

```
// Adicionando uma nova propriedade  
pessoa.altura = 1.75
```

```
// Modificando uma propriedade existente
pessoa.idade = 29

// Removendo uma propriedade
delete pessoa.profissao

console.log(pessoa)
// Resultado:
// { nome: 'Carlos', idade: 29, possuiCNH: false, altura: 1.75 }
```

## Arrays

Os arrays funcionam como uma **lista ordenada de itens**, como uma fila de pessoas ou uma lista de compras. Cada item na lista tem uma posição específica, chamada de **índice**, que **começa do zero**. Isso significa que o primeiro elemento de um array está na posição 0, o segundo na posição 1, e assim por diante.

Os arrays são versáteis e podem armazenar dados de diferentes tipos, incluindo números, strings, objetos, e até mesmo outros arrays, o que os torna ideais para representar coleções de dados que precisam ser processadas sequencialmente, além de permitir que você faça coisas como adicionar novos valores ao final da lista ou substituir valores de qualquer posição da lista.

## Criando Arrays

Existem várias maneiras de criar arrays em JavaScript. A forma mais comum é usar colchetes [ ].

```
const frutas = ["maçã", "banana", "laranja"]
const notas = [8.5, 9.3, 10]
```

Nesse exemplo, criamos dois arrays chamados **frutas** e **notas**. O primeiro contém três strings e o segundo três dados numéricos.

## Acessando elementos de um array

Podemos acessar os elementos desse array usando seus **índices** (posições).

```
console.log(frutas[0]) // "maçã"
console.log(frutas[1]) // "banana"
console.log(frutas[2]) // "laranja"

console.log(notas[0]) // 8.5
console.log(notas[1]) // 9.3
console.log(notas[2]) // 10
```

## Tamanho de listas

Para saber o tamanho de um array em JavaScript, você pode usar a propriedade `.length`. Essa propriedade retorna o número de elementos presentes no array. O valor retornado é um número inteiro que representa a quantidade de itens armazenados.

```
const frutas = ["maçã", "banana", "laranja"];
const notas = [];

console.log(frutas.length); // 3
console.log(notas.length); // 0
```

## Modificando Arrays

Os arrays em JavaScript são dinâmicos, o que significa que você pode modificar os valores de índice a qualquer momento, bem como adicionar novos valores ao final da lista.

```
const frutas = ["maçã", "banana", "laranja"];
const notas = [8.5, 9.3, 10];

frutas[1] = "uva";
notas[notas.length] = 9.8;

console.log(frutas); // [ 'maçã', 'uva', 'laranja' ]
console.log(notas);  // [ 8.5, 9.3, 10, 9.8 ]
```

## Conclusão

Esses dois tipos estruturados, objetos e arrays, tornam-se ferramentas essenciais para qualquer pessoa que está começando a programar em JavaScript, porque ajudam a transformar dados desorganizados em formatos lógicos e acessíveis.

Objetos são ótimos para quando você quer representar algo com várias características, como uma pessoa ou um produto, enquanto arrays são ideais para listas de itens semelhantes. Juntos, eles facilitam muito o processo de armazenamento, organização e manipulação de dados, o que é fundamental para criar programas e aplicações que funcionem de maneira eficiente.

# BONS ESTUDOS

