

# JavaScript – Destructuring (Desestruturação)

## O que é Destructuring?

Destructuring é uma forma de tirar valores de objetos, arrays ou strings e guardar em variáveis de maneira simples.

---

## Destructuring de Objetos

### Exemplo básico:

```
const person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 50  
};  
  
let { firstName, lastName } = person;
```

→ Cria variáveis com os **mesmos nomes das propriedades**.

#### ◆ Ordem não importa:

```
let { lastName, firstName } = person;
```

#### ◆ Valor padrão (default):

```
let { firstName, country = "US" } = person;
```

→ Se `country` não existir, usa "US".

#### ◆ Alias (apelido para a variável):

```
let { lastName: name } = person;
```

→ `name` recebe o valor de `lastName`.

 **Destructuring não altera o objeto original.**

---

## Destructuring de Strings

Strings são iteráveis, então dá para separar letras:

```
let word = "W3Schools";  
let [a, b, c, d] = word;
```

→ Cada variável recebe um caractere.

---

## Destructuring de Arrays

### Exemplo básico:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let [fruit1, fruit2] = fruits;
```

#### ◆ Pular valores:

```
let [fruit1, , , fruit2] = fruits;
```

#### ◆ Pegar posições específicas:

```
let {[0]: first, [1]: second} = fruits;
```

---

## Rest Operator (...)

Guarda o resto dos valores em um array:

```
const numbers = [10, 20, 30, 40, 50];
const [a, b, ...rest] = numbers;
```

➡ rest = [30, 40, 50]

---

## Destructuring com Map

```
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);

for (const [key, value] of fruits) {
  console.log(key + " = " + value);
}
```

---

## Trocar valores de variáveis (Swap)

```
let firstName = "John";
let lastName = "Doe";

[firstName, lastName] = [lastName, firstName];
```

➡ Troca os valores sem variável extra.

---

## Compatibilidade

- Destructuring é **ES6**
- Funciona em todos os navegadores modernos desde **2017**