

2 - Revisando Objetos no Javascript

1. Os objetos no Javascript são estruturas flexíveis que permitem armazenar pares chave-valor:

```
const book = { title: "Eragon" }
```

2. Aos pares com valores “estáticos”, como tipos primitivos, arrays e outros objetos, damos o nome de atributos:

```
const book = { title: "Eragon", pages: 468, published: true, inStock: 20, tags: ["fantasy", "adventure", "medieval"], author: { name: "Christopher Paolini" } }
```

3. Aos pares que armazenam procedimentos, ou seja, funções, damos o nome de métodos:

```
const book = { title: "Eragon", pages: 468, published: true, inStock: 20, tags: ["fantasy", "adventure", "medieval"], author: { name: "Christopher Paolini" }, addOnStock(quantity) { this.inStock += quantity } }
```

4. Graças ao poder das funções anônimas do javascript, também podemos escrever métodos utilizando chaves que armazenam funções. Elas também são chamadas de métodos:

```
const book = { title: "Eragon", pages: 468, published: true, inStock: 20, tags: ["fantasy", "adventure", "medieval"], author: { name: "Christopher Paolini" }, addOnStock(quantity) { this.inStock += quantity }, save: function () { // Save book data to database } } console.log(book)
```

5. Um recurso que ainda não abordamos antes, mas que existe no Javascript são as **funções construtoras**, que são funções usadas para construir objetos seguindo um modelo específico:

```
function Book(title, pages, tags, author) { this.title = title
this.pages = pages this.tags = tags this.author = author
this.published = false this.inStock = 0 this.addOnStock =
function addOnStock(quantity) { this.inStock += quantity }
this.save = function () { // Save book to database } } const
author = { name: "Christopher Paolini" } const tags =
["fantasy", "adventure", "medieval"] const eragon = new
Book("Eragon", 468, tags, author) eragon.addOnStock(20) const
eldest = new Book("Eldest", 644, tags, author)
console.log(eration) console.log(eldest)
```