

Téc em Desenvolvimento  
de Sistemas Bilíngue

Desenvolver Código

Orientado a Objetos

UC4 | Prof. Leonardo de Souza

# Estruturas de Dados

Imagine que você tem uma prateleira cheia de livros desorganizados. Quando precisa encontrar um livro específico, é como procurar uma agulha no palheiro, certo?

# Estruturas de Dados

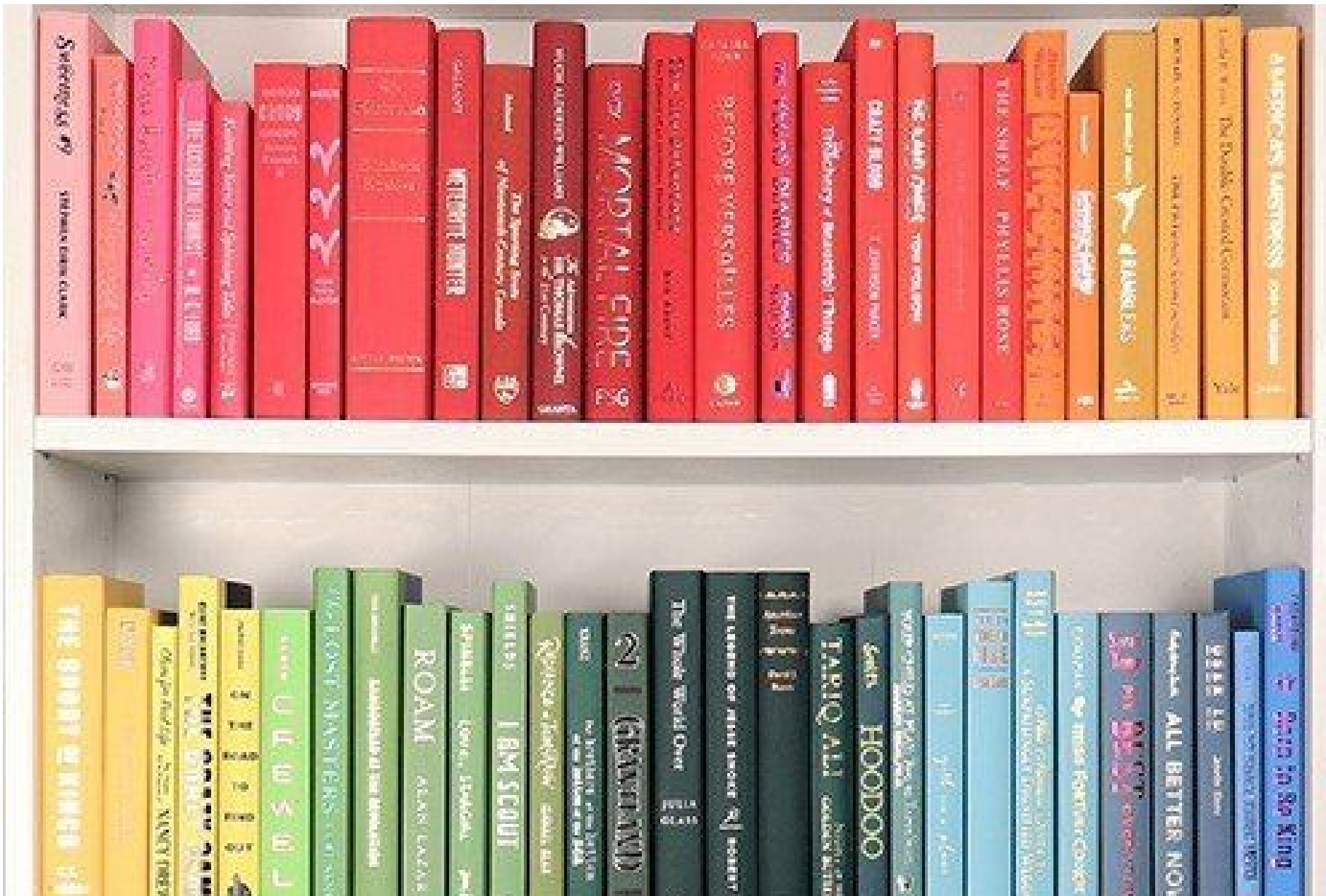


Difícil achar  
algo aqui,  
não?

# Estruturas de Dados

Agora, pense em uma maneira de organizar esses livros para que você possa encontrá-los facilmente quando quiser ler ou estudar.

# Estruturas de Dados



Muito melhor!

## Estruturas de Dados

Pergunta: como podemos organizar os livros de uma biblioteca?

# Estruturas de Dados

Estruturas de dados são como diferentes maneiras de organizar seus livros: você pode classificá-los por gênero, autor ou até mesmo por ordem alfabética.

# Estruturas de Dados

Cada estrutura de dados tem suas próprias vantagens e desvantagens, e escolher a certa pode tornar sua vida muito mais fácil quando se trata de encontrar e gerenciar seus livros.



# Estruturas de Dados

Existe um recurso que nos permite percorrer cada livro em nossa estante, nos ajudando a encontrar e ler cada um deles de forma eficiente.

É aí que entra o...

Estruturas de Dados

For

## For

Vamos pensar na nossa prateleira de livros novamente. Digamos que você queira verificar todos os livros para ver quais deles são de ficção científica. Uma maneira de fazer isso é olhar um por um, certo? Bem, o loop for é como um assistente eficiente que faz isso de maneira automática.

For

Com ele, podemos criar uma instrução que diz algo como:

**'Pegue o primeiro livro, veja se é de ficção científica. Se sim, ótimo, se não, pegue o próximo livro.'**

E ele continua fazendo isso até verificar todos os livros na prateleira.

```
for (inicialização; condição; incremento/decremento) {  
    // bloco de código a ser repetido  
}
```

# For

**Inicialização:** É onde você inicializa o contador ou variável de controle do loop. Geralmente, você define uma variável e atribui um valor inicial a ela. Esta parte é executada apenas uma vez, no início do loop.

## For

**Condição:** É a condição que determina se o loop deve continuar ou não. Enquanto essa condição for verdadeira, o bloco de código dentro do loop será executado. Se a condição for falsa, o loop termina e a execução continua após o loop.

# For

**Incremento/Decremento:** É onde você altera o valor da variável de controle do loop. Isso geralmente envolve aumentar ou diminuir o valor da variável. Essa parte é executada após cada iteração do loop.



# For

**Bloco de código:** É o conjunto de instruções que serão executadas a cada iteração do loop. Essas instruções podem ser qualquer coisa que você queira repetir várias vezes, como manipulação de dados, chamadas de função, etc.

## For

Importante: **ITERACÃO** é o nome que se dá a cada “volta” que o loop dá!

```
// Lista de livros
let livros: string[] = ["Fundação", "Neuromancer", "Orgulho e Preconceito", "Duna"];

// Verificar se cada livro é de ficção científica
for (let i = 0; i < livros.length; i++) {
  if (livros[i] === "Fundação" || livros[i] === "Neuromancer" || livros[i] === "Duna") {
    console.log(`${livros[i]} é um livro de ficção científica.`);
  } else {
    console.log(`${livros[i]} não é um livro de ficção científica.`);
  }
}
```

# For

Executa uma iteração com início e término determinados.

Não precisa estar ligada a um array.

O iterador pode ser de qualquer tipo.

Estruturas de Dados

For In

## For In

Vamos pensar novamente na nossa biblioteca de livros. Digamos que você queira achar um determinado autor dentro de um determinado gênero. Uma maneira de fazer isso é olhar cada livro individualmente, certo?

## For In

Com o loop `for...in`, podemos criar uma instrução que diz algo como: 'Pegue o primeiro livro EM ficção científica, veja quem é o autor.

Depois, pegue o próximo livro.' E ele continua fazendo isso até verificar todos os livros de ficção.

```
// Definindo um objeto representando os livros de ficção com seus respectivos autores
let ficcao = {
  "Fundação": "Isaac Asimov",
  "Duna": "Frank Herbert"
};

// Iterando sobre os livros de ficção e exibindo os autores
for (let livro in ficcao) {
  console.log(`Autor do livro "${livro}": ${ficcao[livro]}`);
}
```



## For In

Executa uma iteração SEMPRE com base em um array  
Sempre utiliza um iterador do tipo string como contador.  
Ex.: "0", "1", "2", "3",..

Utiliza uma const como base pois recria a variável em cada iteração e não permite a sua alteração

Estruturas de Dados

For Of

## For Of

Ainda no exemplo da nossa biblioteca, nós podemos percorrer nossos livros de outra forma, bem mais simples.

```
// Array representando os títulos dos livros na prateleira
let meusLivros = ["Fundação", "Duna", "Orgulho e Preconceito"];

// Iterando sobre os títulos dos livros e exibindo-os
for (let livro of meusLivros) {
  console.log(livro);
}
```

## For In

Executa uma iteração SEMPRE com base em um array.

Não possui um iterador. Sempre processa diretamente os itens do array. Ou seja, não usamos índice aqui.

Utiliza uma const como base pois recria a variável em cada iteração e não permite a sua alteração

Estrutura de Dados

For Each

## For Each

Agora imagine que queremos ler todos os livros de um determinado autor. Existe um modo de percorrer (ler) todos os livros desejados e executar uma ação para cada um através do `forEach`.

```
// Array representando os títulos dos livros na prateleira
let tolkienLivros = ["Senhor dos Aneis", "O Hobbit", "Contos Inacabados"];

// Iterando sobre os elementos do array usando forEach
tolkienLivros.forEach(livro => {
  console.log(`Eu já li ${livro}`);
});
```



## For In

Executa uma iteração SEMPRE com base em um array.

Não possui um iterador. Sempre processa diretamente os itens do array.