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?

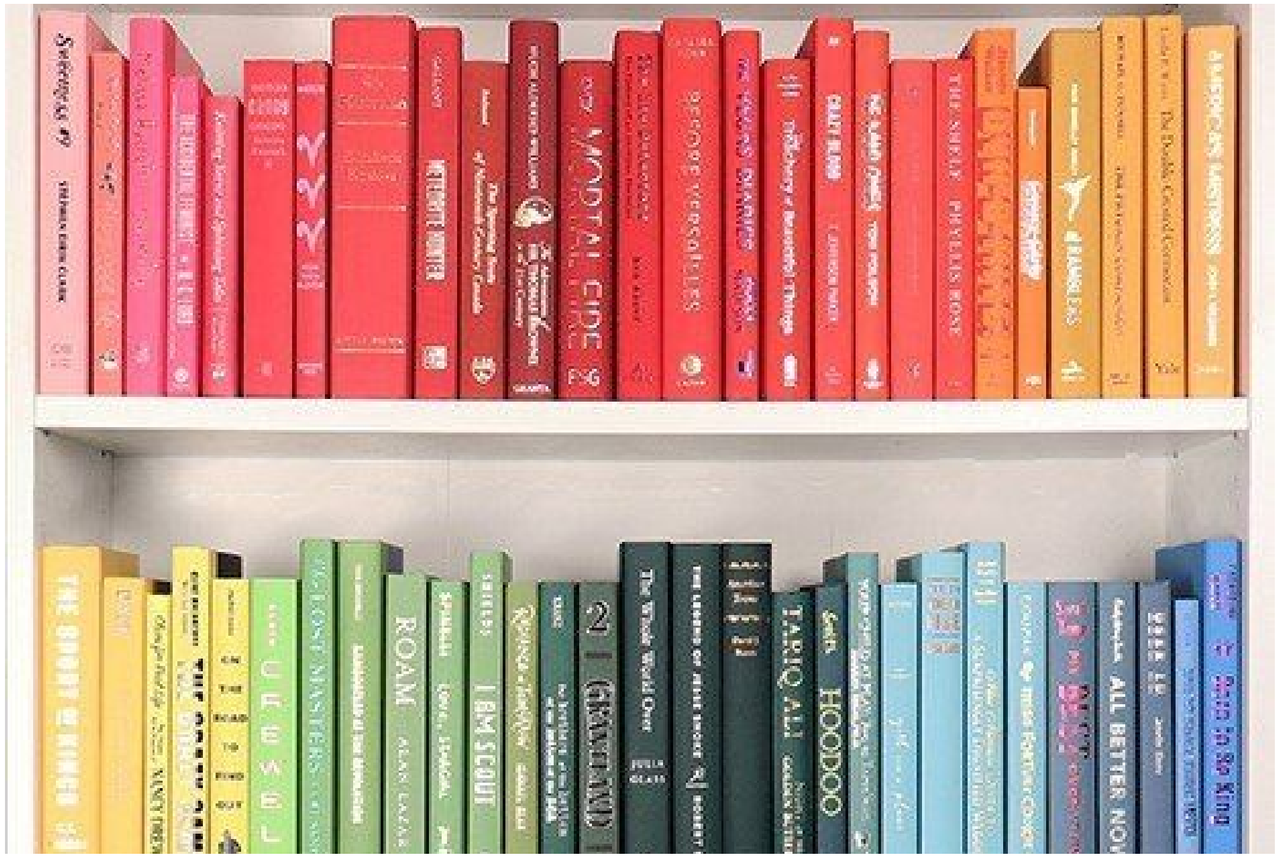
Estruturas de Dados

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!

```typescript
// 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.

```javascript
// 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

- O for-in itera sobre as propriedades de um objeto.
- É usado principalmente para iterar sobre as chaves de um objeto.
- Não é recomendado para iterar sobre arrays, pois pode incluir propriedades herdadas e não numéricas.

For In

Não é recomendado utilizar o forIn para percorrer arrays

```
const arr = [1, 2, 3];

for (const index in arr) {
    console.log(typeof index); // Imprime '0', '1', '2'
}
```

Quando usar

- É útil quando você precisa acessar as chaves e os valores associados de um objeto durante a iteração.

Porque usar

- Sintaxe simples
- Fornece tanto as chaves quanto os valores

Estruturas de Dados

For Of

For Of

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

```javascript
// 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

- O for-of é uma maneira de iterar sobre os valores de um objeto iterável, como arrays, strings, mapas, conjuntos, etc.
- É mais intuitivo e limpo do que for-in, pois itera apenas sobre os valores e não sobre as chaves.
- Não é possível usar for-of em objetos não-iteráveis.

Quando usar

- Quando você precisa apenas iterar sobre os valores dos elementos.
- Quando você precisa apenas dos valores e não precisa se preocupar com o índice ou com a referência ao array original.

Porque usar

- Fornece uma sintaxe mais limpa e concisa.

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.

```
let tolkienLivros = ['Senhor dos Aneis', 'O Hobbit', 'Contos Inacabados'];

tolkienLivros.forEach(livro => {
    console.log(`Eu já li ${livro}`);
});
```

```
let tolkienLivros = ['Senhor dos Aneis', 'O Hobbit', 'Contos Inacabados'];

tolkienLivros.forEach(function(livro) {
    console.log(`Eu já li ${livro}`);
});
```

For Each

- Executa uma determinada função para cada elemento do array.
- É uma maneira mais conveniente e segura de iterar sobre os elementos de um array em comparação com for-in.
- Não é possível usar o forEach em objetos não-iteráveis (ou seja, só se usa em arrays). Pode ser usado em objetos mas não retorna valor dos atributos.

For Each

Não utiliza um iterador como contador, mas sim passa cada elemento do array para a função de callback (uma função que é passada como argumento para outra função)

For Each

Essa função de callback aceita até 3 parâmetros:

- element: O valor do elemento atual sendo processado no array. (ex: "carro")
- index: O índice do elemento atual sendo processado no array.("ex:0")
- array: O array original sobre o qual forEach() foi chamado.(ex:["carro", "moto", "bicicleta"])

```
const arr = ["carro", "moto", "bicicleta"];

arr.forEach((element, index, array) => {
    console.log(`Elemento ${element} no índice ${index}`);
    console.log('Array original:', array);
});
```

Quando usar

- É usado quando você precisa iterar sobre os valores dos elementos e também precisa acessar o índice ou o array original durante a iteração.
- É útil quando você precisa realizar uma operação em cada elemento do array e precisa acessar o índice ou o array original durante essa operação.

Porque usar

- Fornece uma maneira conveniente de fazer algo com cada elemento do array.
- Permite passar uma função de callback que pode receber até três argumentos: o valor do elemento, o índice do elemento e o array original.

Estrutura de Dados

RESUMO

forIn

- Use quando estiver iterando sobre as propriedades e valores de um objeto.
- Evite usar em arrays.

forOf

- Use quando estiver iterando sobre os valores de uma estrutura de dados iterável, como um array.
- Ideal quando você só precisa dos valores dos elementos e não precisa se preocupar com o índice ou com a referência ao array original.

forEach

- Use quando estiver iterando sobre os valores de uma estrutura de dados iterável, como um array, mas precisa realizar uma operação em cada elemento do array e precisa acessar o índice ou o array original durante essa operação.