



IMPLEMENTAÇÃO DE PILHAS EM JAVASCRIPT

ESTRUTURA DE DADOS

CST em Desenvolvimento de Software Multiplataforma



PROF. Me. TIAGO A. SILVA



LIVRO DE REFERÊNCIA DA DISCIPLINA

- **BIBLIOGRAFIA BÁSICA:**

- GRONER, Loiane. **Estrutura de dados e algoritmos com JavaScript**: escreva um código JavaScript complexo e eficaz usando a mais recente ECMAScript. **São Paulo: Novatec Editora, 2019.**

- **NESTA AULA:**

- **Capítulo 4**



PARA SOBREVIVER AO JAVASCRIPT

Non-zero value



null



0



undefined



PILHA

*Exemplo de Implementação em
JavaScript, sem o uso de funções
nativas*

IMPLEMENTAÇÃO DE UMA PILHA

- Estrutura interna (**items**): Usamos um objeto para armazenar os elementos da pilha, com chaves representando as posições.
 - **adicionar(elemento)**: Insere o elemento no topo da pilha, na posição correspondente ao tamanho atual. Em seguida, incrementa o tamanho da pilha.
 - **remove()**: Remove e retorna o elemento do topo da pilha, ajustando o tamanho após a remoção. Verifica se a pilha está vazia antes de tentar remover.
 - **topo()**: Retorna o elemento no topo da pilha sem removê-lo.
 - **estaVazia()**: Verifica se a pilha está vazia, retornando **true** ou **false**.
 - **tamanhoPilha()**: Retorna o número de elementos na pilha.
 - **limpar()**: Remove todos os elementos da pilha, redefinindo os valores de **items** e tamanho.
- Essa implementação simula o comportamento básico de uma pilha, sem recorrer a métodos nativos como **push()** e **pop()**, e realiza o gerenciamento de dados e índices de maneira manual.

```
1 class MinhaPilha {
2
3     constructor() {
4         this.items = {}; // Usamos um objeto para armazenar os elementos da pilha
5         this.tamanho = 0; // Mantemos o controle do tamanho da pilha
6     }
7
8     // Adiciona um elemento ao topo da pilha
9     adicionar(elemento)
10    {
11        this.items[this.tamanho] = elemento; // Insere o elemento na posição atual do tamanho
12        this.tamanho++; // Incrementa o tamanho
13    }
14
15    // Remove e retorna o elemento do topo da pilha
16    remover()
17    {
18        if (this.tamanho === 0) {
19            return undefined; // Se a pilha estiver vazia, retorna undefined
20        }
21
22        const ultimoItem = this.items[this.tamanho - 1]; // Pega o item no topo da pilha
23        delete this.items[this.tamanho - 1]; // Remove o item do topo
24        this.tamanho--; // Decrementa o tamanho
25
26        return ultimoItem; // Retorna o item removido
27    }
28 }
```

```
28
29 // Retorna o elemento no topo da pilha sem removê-lo
30 topo()
31 {
32     if (this.tamanho === 0) {
33         return undefined; // Se a pilha estiver vazia, retorna undefined
34     }
35     return this.items[this.tamanho - 1]; // Retorna o item no topo
36 }
37
38 // Verifica se a pilha está vazia
39 estaVazia()
40 {
41     return this.tamanho === 0; // Verifica se o tamanho da pilha é zero
42 }
43
44 // Retorna o número de elementos na pilha
45 tamanhoPilha()
46 {
47     return this.tamanho; // Retorna o tamanho da pilha
48 }
49
50 // Limpa a pilha
51 limpar()
52 {
53     this.items = {}; // Reseta os itens
54     this.tamanho = 0; // Reinicializa o tamanho
55 }
56 }
```

PILHA – COMO USAR

```
57
58 // Exemplo de uso
59 let minha_variavel = new MinhaPilha();
60
61 minha_variavel.adicionar(10);
62 minha_variavel.adicionar(20);
63 minha_variavel.adicionar(30);
64
65 console.log(minha_variavel.topo()); // Saída: 30 (Elemento no topo)
66
67 console.log(minha_variavel.remover()); // Saída: 30 (Remove o elemento do topo)
68 console.log(minha_variavel.topo()); // Saída: 20 (Agora o topo é 20)
69
70 console.log(minha_variavel.tamanhoPilha()); // Saída: 2 (Dois elementos restantes)
```


EXERCÍCIOS

Use a classe implementada acima para resolver os exercícios

EXERCÍCIO 1

- O grande feiticeiro Eldrin vive em uma torre cheia de livros de feitiços. Ele sempre coloca os novos livros no topo da pilha e só pode pegar o que está no topo primeiro.
- Implemente um sistema para ajudá-lo a organizar seus feitiços com as seguintes funcionalidades:
 - Adicionar um novo feitiço no topo da pilha.
 - Remover o feitiço do topo (Eldrin usou e guardou em outro lugar).
 - Ver qual feitiço está no topo sem removê-lo.
 - Verificar se não há mais feitiços na pilha.

EXERCÍCIO 2

- O Capitão Barbaruiva tem um baú cheio de tesouros, mas ele só consegue pegar o último item que colocou.
- Crie um programa para ajudá-lo a:
 - Guardar um novo tesouro no baú.
 - Retirar o último tesouro guardado.
 - Olhar qual é o último tesouro sem pegá-lo.
 - Saber se o baú está vazio.

RESOLUÇÃO DO EXERCÍCIO

Só veja após tentar fazer 😊

RESOLUÇÃO DO EXERCÍCIO 1

```
1  class SpellStack {
2      constructor() {
3          this.spells = [];
4      }
5
6      // Adiciona um novo feitiço no topo
7      addSpell(spell) {
8          this.spells.push(spell);
9          console.log(`Feitiço "${spell}" adicionado!`);
10     }
11
12     // Remove e retorna o feitiço do topo
13     removeSpell() {
14         if (this.isEmpty()) {
15             console.log("Nenhum feitiço na torre!");
16             return;
17         }
18         console.log(`Feitiço "${this.spells.pop()}" removido!`);
19     }
20 }
```

RESOLUÇÃO DO EXERCÍCIO 1

```
21 // Mostra o feitiço no topo sem removê-lo
22 peekSpell() {
23     if (this.isEmpty()) {
24         console.log("Nenhum feitiço para ver!");
25         return;
26     }
27     console.log(`Feitiço no topo: "${this.spells[this.spells.length - 1]}"`);
28 }
29
30 // Verifica se a pilha de feitiços está vazia
31 isEmpty() {
32     return this.spells.length === 0;
33 }
34 }
```

RESOLUÇÃO DO EXERCÍCIO 1

```
36  const tower = new SpellStack();
37  tower.addSpell("Bola de Fogo");
38  tower.addSpell("Escudo Arcano");
39  tower.peekSpell(); // "Escudo Arcano"
40  tower.removeSpell(); // "Escudo Arcano" removido
41  tower.removeSpell(); // "Bola de Fogo" removido
42  tower.removeSpell(); // Nenhum feitiço na torre!
```

OBRIGADO!

- Encontre este **material on-line** em:
 - Slides: Plataforma Microsoft Teams
- Em caso de **dúvidas**, entre em contato:
 - **Prof. Tiago:** tiago.silva238@fatec.sp.gov.br

