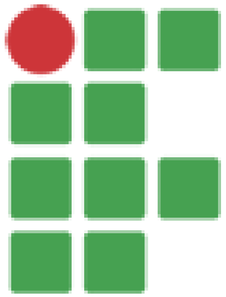


INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Disciplina

Programação Front-End I

PROF. FERNANDO TAMBERLINI ALVES

Estrutura de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

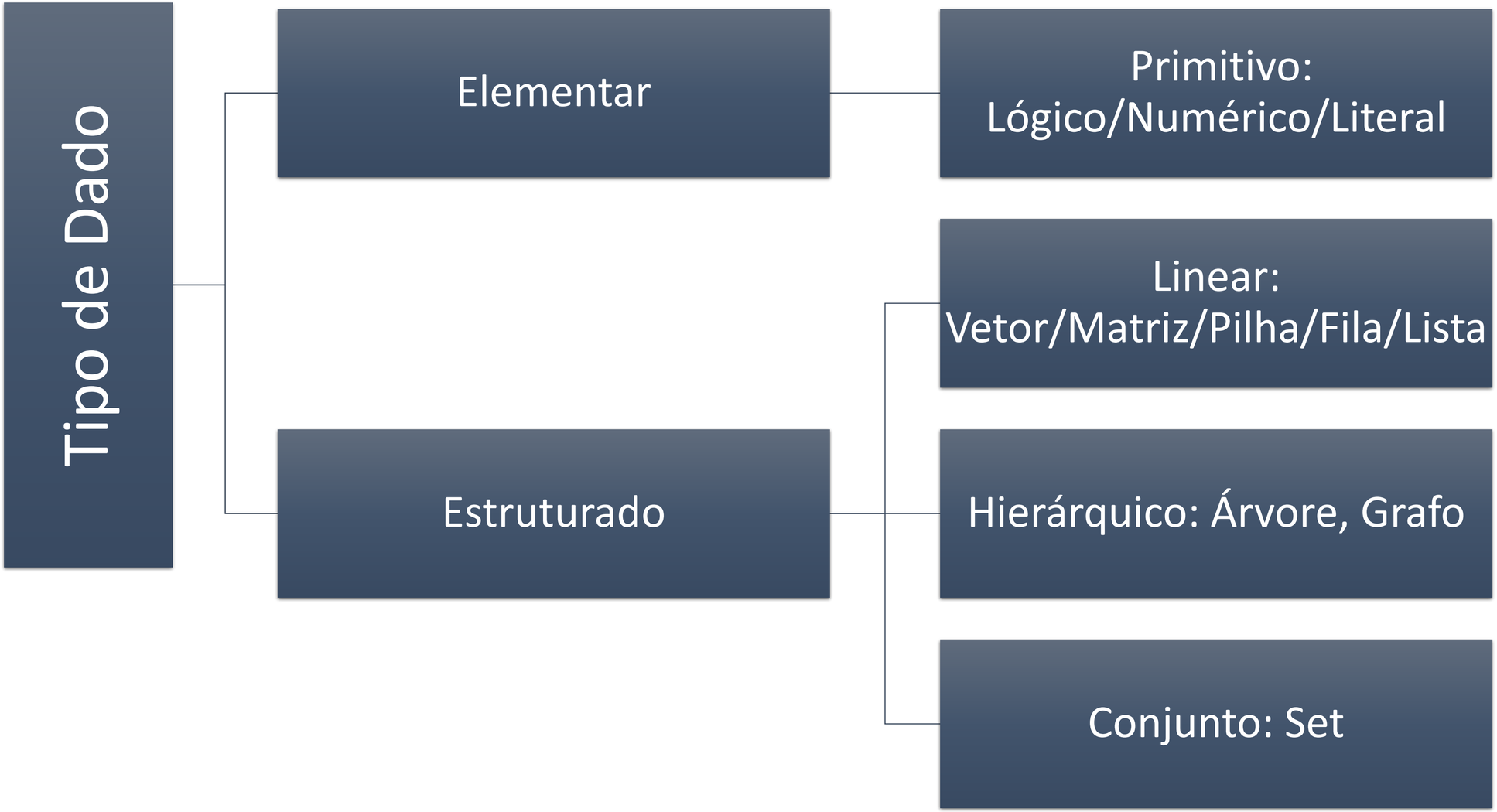
- Um conjunto de teorias e práticas responsáveis por definir a forma como os dados podem ser armazenados, representados e consequentemente manipulados.

- É nessa área de estudo que são apresentadas as formas que os algoritmos podem definir e manipular os dados.

- **Cada estrutura terá a forma de busca, inclusão e exclusão de elemento distinta.**

- **O grande desafio é utilizar a melhor estrutura para o problema em questão.**

Tipo do Dado


Tipo de Dado
Elementar
Primitivo: Lógico/Numérico/Literal
Estruturado
Linear: Vetor/Matriz/Pilha/Fila/Lista
Hierárquico: Árvore, Grafo
Conjunto: Set

Estrutura de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Linear: São estruturas onde os elementos seguem uma ordem sequencial, um após o outro.

- Principais Exemplos:
  - Vetor/Matriz (array)
    - Acesso direto por índice;
  - Lista Encadeada (linked list)
    - Elementos ligado por ponteiro;
  - Pilha (stack)
    - O último a entrar é o primeiro a sair (LIFO)
  - Fila (queue)
    - O primeiro a entrar é o primeiro a sair (FIFO)
  - Deque
    - Inserção e remoção nas duas extremidades

Estrutura de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Hierárquico: Estruturas com elementos organizados em níveis, onde cada elemento pode ter sub-elementos (filhos).

- Principais Exemplos:
  - Árvores (tree)
    - Binária, Balanceada
  - Heaps
    - O primeiro a entrar é o primeiro a sair (FIFO)

Conjuntos: Coleções sem ordem e sem elementos duplicados.

- Principais Exemplos:
  - Set/Conjuntos

Estrutura de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Hierárquico: Estruturas que usam uma função hash para mapear chaves → valores.

- Principais Exemplos:
  - Hash Table
  - Hash MapD
  - Dictionary (em Python)
  - Object literal (em JavaScript)

Hierárquico: Entidades compostas por nós (vértices) conectados por arestas. Não são hierárquicas nem lineares. Formam redes.

- Principais Exemplos:
  - Grafo direcionado
  - Grafo Ponderado

Estrutura de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Homogênea: Todos os elementos armazenados têm o mesmo tipo de dado.

Heterogênea: A estrutura pode conter diferentes tipos de dados ao mesmo tempo.

Estática: O tamanho é fixo desde o momento da criação. A memória é alocada antes de usar.

Dinâmica: O tamanho pode aumentar ou diminuir durante a execução do programa. A memória é alocada sob demanda..

Estrutura de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Homogênea: Todos os elementos armazenados têm o mesmo tipo de dado.

Heterogênea: A estrutura pode conter diferentes tipos de dados ao mesmo tempo.

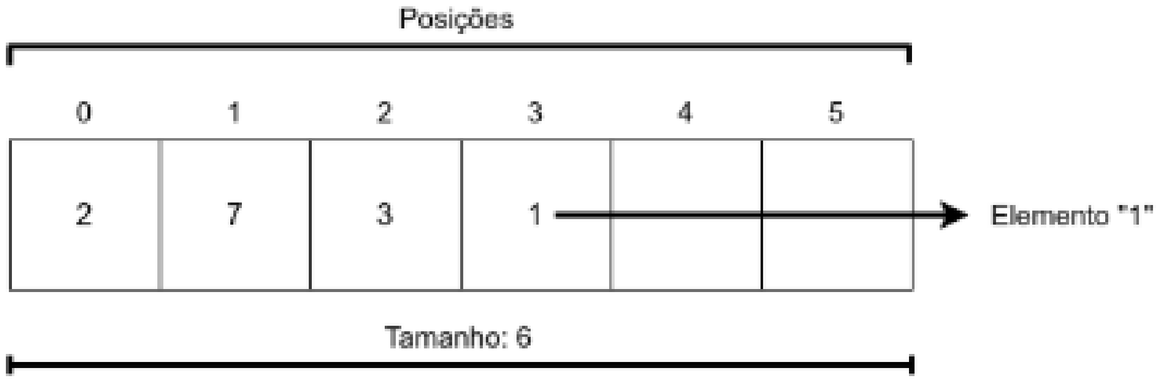
Estática: O tamanho é fixo desde o momento da criação. A memória é alocada antes de usar.

Dinâmica: O tamanho pode aumentar ou diminuir durante a execução do programa. A memória é alocada sob demanda..

Vetor – Array Unidimensional

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Elemento: é o dado que está armazenado nas posições do vetor/
- Tamanho: é a capacidade total de elementos que o vetor pode armazenar
- Índice: é o número de uma das posições que o vetor pode possuir. Caso o primeiro índice seja “0” (zero), a posição é limitada ao tamanho – 1.



Array em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Acessando Elementos do Array

```
let cores = ["azul", "verde", "vermelho"];
cores[0]; // "azul"
cores[2]; // "vermelho“
```

- Podemos alterar o valor

```
cores[1] = "amarelo"; // cores = ["azul", "amarelo", "vermelho"]
```

- Podemos verificar o tamanho do array

```
let dias = ["segunda", "terça", "quarta"];
dias.length; // retorna 3
```

Pilha

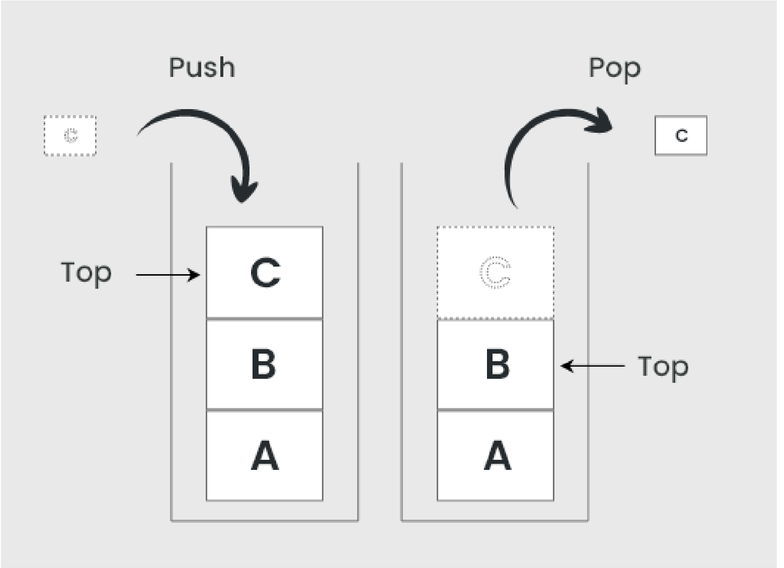
- Pilha: é uma estrutura de dados onde os elementos são dispostos, conceitualmente, um em cima do outro;

- Podemos definir o comportamento de uma pilha:

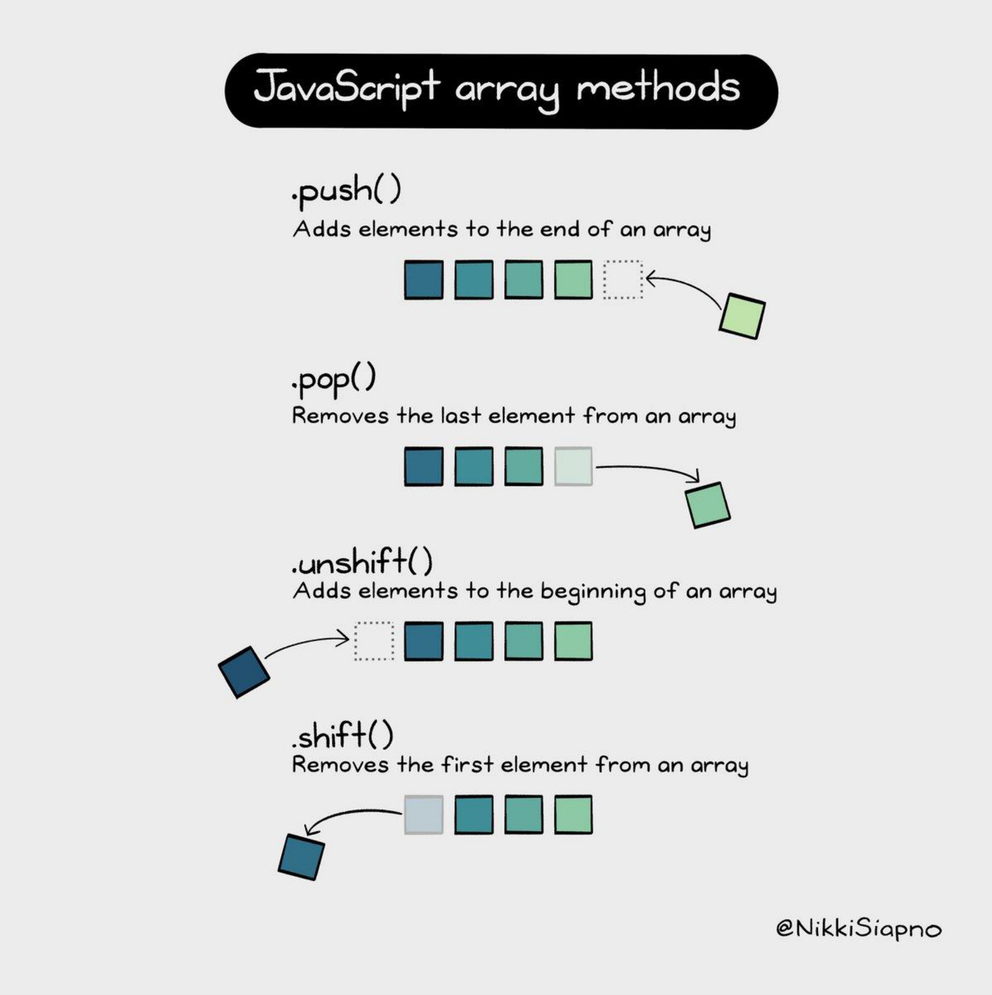
  - LIFO (last in, first out): o último a entrar é o primeiro a sair
  - FILO (first in, last out): o primeiro a entrar é o último a sair

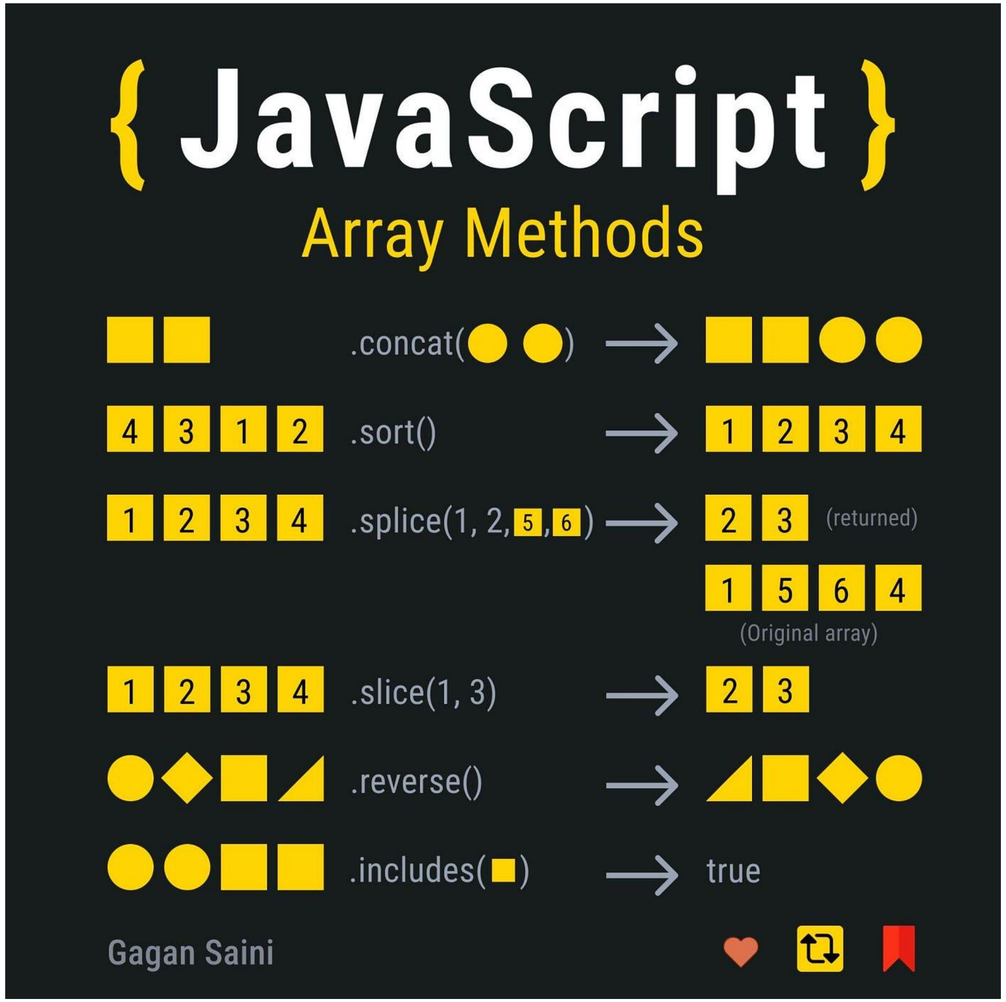
Pilha

- Operações sobre a pilha:
  - push: empilhar elemento;
  - size: retorna o tamanho da pilha
  - top: o topo da pilha
  - pop: desempilhar elemento;
  - empty: verificar se está vazia



Array no Javacript





Array no JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos de um array

  - **push(elemento)** - adiciona um elemento no final do array
    ex: let arr = [1, 2]; arr.push(3); // arr = [1, 2, 3]
  - **pop()** - remove o último elemento do array e o retorna
    ex: let arr = [1, 2, 3]; arr.pop(); // retorna 3, arr = [1, 2]
  - **shift()** - remove o primeiro elemento do array e o retorna
    ex: let arr = [1, 2, 3]; arr.shift(); // retorna 1, arr = [2, 3]
  - **unshift(elemento)** - adiciona um elemento no início do array
    ex: let arr = [2, 3]; arr.unshift(1); // arr = [1, 2, 3]
  - **concat(array2)** - retorna um novo array com a junção de dois arrays
    ex: [1, 2].concat([3, 4]) // retorna [1, 2, 3, 4]
  - **join(separador)** - junta os elementos em uma string
    ex: [1, 2, 3].join("-") // retorna "1-2-3"

Array no JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos de um array

  - **slice(início, fim)** - retorna uma cópia parcial do array, do índice início até fim (excluindo fim)
    ex: [1, 2, 3, 4].slice(1, 3) // retorna [2, 3]
  - **splice(início, qtd, ...itens)** - remove e/ou adiciona elementos
    ex: let arr = [1, 2, 3]; arr.splice(1, 1, "a") // arr = [1, "a", 3]
  - **indexOf(valor)** - retorna o índice da primeira ocorrência
    ex: [1, 2, 3].indexOf(2) // retorna 1
  - **lastIndexOf(valor)** - retorna o índice da última ocorrência
    ex: [1, 2, 2, 3].lastIndexOf(2) // retorna 2
  - **includes(valor)** - verifica se o valor existe no array
    ex: [1, 2, 3].includes(2) // retorna true
  - **forEach(callback)** - executa uma função para cada elemento
    ex: [1, 2, 3].forEach(x => console.log(x))

Array no JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos de um array

  - **map(callback)** - retorna um novo array com a função aplicada a cada item
    ex: [1, 2, 3].map(x => x * 2) // retorna [2, 4, 6]
  - **filter(callback)** - retorna um novo array com os itens que passam na condição
    ex: [1, 2, 3, 4].filter(x => x > 2) // retorna [3, 4]
  - **reduce(callback, valorInicial)** - reduz o array a um único valor
    ex: [1, 2, 3].reduce((acc, x) => acc + x, 0) // retorna 6
  - **find(callback)** - retorna o primeiro item que satisfaz a condição
    ex: [1, 2, 3].find(x => x > 1) // retorna 2
  - **findIndex(callback)** - retorna o índice do primeiro item que satisfaz a condição
    ex: [1, 2, 3].findIndex(x => x > 1) // retorna 1

Array no JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos de um array

  - **some(callback)** - retorna true se pelo menos um item satisfaz a condição
    ex: [1, 2, 3].some(x => x > 2) // retorna true
  - **every(callback)** - retorna true se todos os itens satisfazem a condição
    ex: [1, 2, 3].every(x => x > 0) // retorna true
  - **sort([função])** - ordena os itens do array (por padrão como string)
    ex: [3, 1, 2].sort() // retorna [1, 2, 3]
    ex: [10, 2].sort((a, b) => a - b) // retorna [2, 10]
  - **reverse()** - inverte a ordem dos elementos
    ex: [1, 2, 3].reverse() // retorna [3, 2, 1]
  - **flat(profundidade)** - retorna um novo array com os subarrays achatados
    ex: [1, [2, [3]]].flat(2) // retorna [1, 2, 3]

Array no JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos de um array

  - **flatMap(callback)** - faz map seguido de flat(1)
    ex: [1, 2].flatMap(x => [x, x * 2]) // retorna [1, 2, 2, 4]
  - **fill(valor, início, fim)** - preenche parte do array com um valor
    ex: [1, 2, 3].fill(0, 1, 3) // retorna [1, 0, 0]
  - **copyWithin(alvo, início, fim)** - copia parte do array para outra posição dentro dele
    ex: [1, 2, 3, 4].copyWithin(1, 2) // retorna [1, 3, 4, 4]
  - **toString()** - converte o array em string
    ex: [1, 2, 3].toString() // retorna "1,2,3"
  - **Array.isArray(valor)** - verifica se o valor é um array
    ex: Array.isArray([1, 2]) // retorna true
    ex: Array.isArray("abc") // retorna false

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- O tipo String representa texto em JavaScript.
- É uma sequência de caracteres Unicode.
- Strings são imutáveis: ao modificar uma string, uma nova é criada.
- Podem ser delimitadas por aspas simples (‘string’) ou dupla (“string“).
- As aspas simples podem ser utilizadas dentro das aspas duplas e vice-versa

```
let texto1 = “Pasta d’água";
let texto2 = “Meu nome é 'Johnny'";
let texto3 = ‘Meu nome é "Johnny"';
```

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Alguns caracteres especiais devem ser acompanhado por "\" (contrabarra ou backslash)

Code	Description
\'	Single quote
\"	Double quote
\\	Backslash
\b	Backspace
\f	Form Feed
\n	New Line
\r	Carriage Return
\t	Horizontal Tabulator

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Criando uma string

```
let nome = "Maria";
let frase = 'Olá mundo!';
let strObj = new String("João"); //não recomendado
```

- Acessando caracteres e propriedades de uma string

```
nome.length     // 5
nome[3]         // "i"
nome.charAt(0)  // "M"
```

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos da classe String

  - `charAt(posição) - retorna o caractere na posição indicada`
    `ex: "casa".charAt(2) retorna "s"`

  - `charCodeAt(posição) - retorna o código Unicode do caractere na posição`
    `ex: "ABC".charCodeAt(1) retorna 66`

  - `concat(string) - concatena duas ou mais strings`
    `ex: "bom".concat(" dia") retorna "bom dia"`

  - `includes(substring) - verifica se a string contém o texto`
    `ex: "banana".includes("na") retorna true`

  - `endsWith(sufixo) - verifica se a string termina com o texto`
    `ex: "arquivo.txt".endsWith(".txt") retorna true`

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos da classe String

  - `startsWith(prefixo) - verifica se a string começa com o texto`
    `ex: "javascript".startsWith("java") retorna true`

  - `indexOf(substring) - retorna a posição da primeira ocorrência da substring`
    `ex: "abacaxi".indexOf("a") retorna 0`

  - `lastIndexOf(substring) - retorna a posição da última ocorrência da substring`
    `ex: "banana".lastIndexOf("a") retorna 5`

  - `match(regex) - retorna as correspondências da expressão regular`
    `ex: "hoje é dia 11".match(/\d+/) retorna ["11"]`

  - `matchAll(regex) - retorna um iterador com todas as correspondências`
    `ex: [..."teste123teste456".matchAll(/\d+/g)] retorna [["123"], ["456"]]`

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos da classe String

  - `normalize() - normaliza a string para o formato Unicode padrão`
    `ex: "café".normalize() retorna "café"`

  - `padStart(tamanho, caractere) - preenche o início até alcançar o tamanho`
    `ex: "5".padStart(3, "0") retorna "005"`

  - `padEnd(tamanho, caractere) - preenche o fim até alcançar o tamanho`
    `ex: "5".padEnd(3, "0") retorna "500"`

  - `repeat(n) - repete a string n vezes`
    `ex: "ha".repeat(3) retorna "hahaha"`

  - `replace(trecho, novo) - substitui trecho por novo valor`
    `ex: "bola azul".replace("azul", "vermelha") retorna "bola vermelha"`

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos da classe String

  - `replaceAll(trecho, novo) - substitui todas as ocorrências`
    `ex: "a-b-a-b".replaceAll("a", "x") retorna "x-b-x-b"`

  - `search(regex) - busca uma expressão regular e retorna o índice da primeira ocorrência`
    `ex: "teste123".search(/\d/) retorna 5`

  - `slice(início, fim) - retorna parte da string do início ao fim (não inclui fim)`
    `ex: "abcdef".slice(1, 4) retorna "bcd"`

  - `split(separador) - divide a string em array usando o separador`
    `ex: "a,b,c".split(",") retorna ["a", "b", "c"]`

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos da classe String

  - `substring(início, fim) - retorna parte da string entre início e fim`
    `ex: "abcdef".substring(1, 4) retorna "bcd"`

  - `toLowerCase() - converte a string para minúsculas`
    `ex: "Oi".toLowerCase() retorna "oi"`

  - `toUpperCase() - converte a string para maiúsculas`
    `ex: "oi".toUpperCase() retorna "OI"`

  - `toLocaleLowerCase() - minúscula com base no idioma`
    `ex: "TÜRKÇE".toLocaleLowerCase("tr") retorna "türkçe"`

  - `toLocaleUpperCase() - maiúscula com base no idioma`
    `ex: "istanbul".toLocaleUpperCase("tr") retorna "İSTANBUL"`

String em JavaScript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Principais métodos da classe String

  - `toString()` - retorna a string (útil em objetos)
    `ex: new String("abc").toString() retorna "abc"`

  - `trim()` - remove espaços do início e fim
    `ex: " oi ".trim() retorna "oi"`

  - `trimStart()` - remove espaços apenas do início
    `ex: " oi ".trimStart() retorna "oi "`

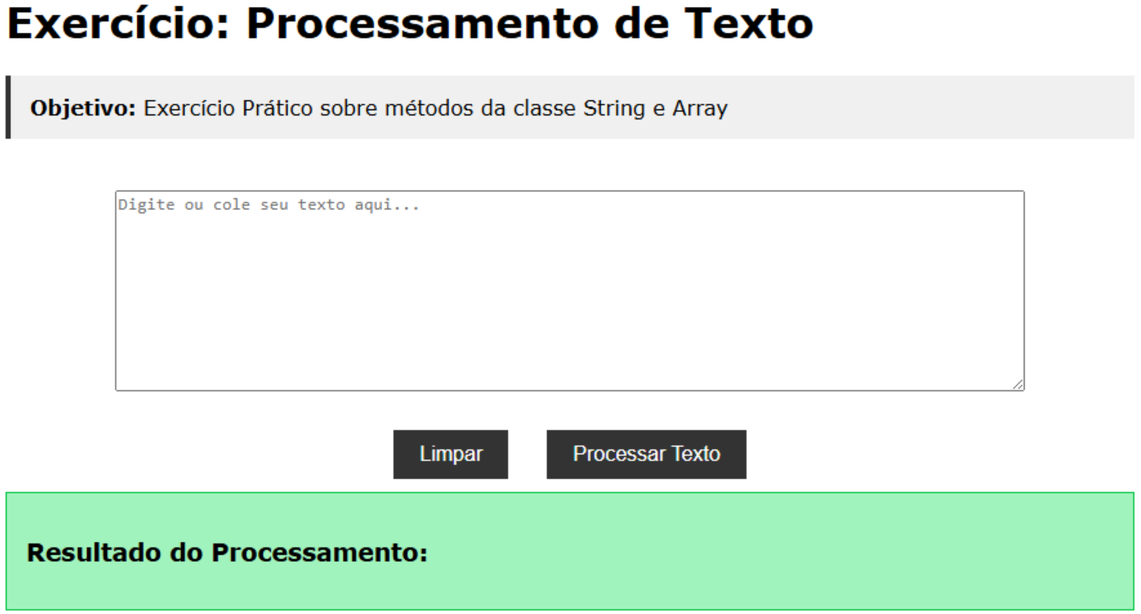
  - `trimEnd()` - remove espaços apenas do fim
    `ex: " oi ".trimEnd() retorna " oi"`

  - `valueOf()` - retorna o valor primitivo da string
    `ex: new String("abc").valueOf() retorna "abc"`

Atividade Prática I

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Utilize o código html/css disponibilizado no Github e codifique um código JavaScript que faça o seguinte processamento:



Atividade Prática II

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Instruções do Exercício:

Objetivo: Praticar o uso de métodos de String e Array em JavaScript. Ao clicar no Processar, deverá processar o texto e exibir na lista

O que você deve implementar:

1. **Botão Limpar:**
   - Deve limpar o conteúdo da textarea
   - Deve limpar a lista de resultados
2. **Botão Processar:**
   - **Texto em MAIÚSCULAS**
   - **Texto em minúsculas**
   - **Número total de caracteres** (incluindo espaços)
   - **Número de caracteres sem espaços** `Dica: use replace() com regex ou split() e join()`
   - **Número de palavras** `Dica: use split(' ') e filter() para remover strings vazias`
   - **Número de linhas** `Dica: use split('\n')`
   - **Primeira palavra do texto** (se houver)
   - **Última palavra do texto** (se houver)
   - **Palavras únicas em ordem alfabética** `Dica: use Set para remover duplicatas, depois sort()`
   - **A palavra mais longa do texto** `Dica: use reduce() ou sort() com length`

Métodos importantes para usar:

- String: `split()`, `trim()`, `replace()`, `toUpperCase()`, `toLowerCase()`, `length`
- Array: `filter()`, `map()`, `reduce()`, `sort()`, `join()`
- Outros: `new Set()`, `Array.from()`

Dúvidas?

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Dúvidas?

fernando.alves@ifrj.edu.br

ftamberlini.dev.br