



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Rio de Janeiro

Disciplina
Programação Front-End I

PROF. FERNANDO TAMBERLINI ALVES

String em JavaScript

- 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

- 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

- 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      // 6  
nome[3]          // "r"  
Nome.charAt(0)   // "M"
```

String em JavaScript

- 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

- 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

- 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"`

- 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

- 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

- 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"

Array em JavaScript

- Um array é uma estrutura de dados usada para armazenar vários valores em uma única variável.
- Cada valor tem um índice numérico, começando do 0.
- Pode armazenar qualquer tipo: números, strings, objetos, funções, etc.

```
let frutas = ["maçã", "banana", "laranja"];  
let numeros = [1, 2, 3];  
let misto = [42, "texto", true];
```

String em JavaScript

- 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
```

Array no JavaScript

- 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

- 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

- 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

- 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

- 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`

Atividade Prática I

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



- Inclua uma tarefa no final da lista ao clicar no botão Adicionar;
- Depois altere para incluir no início da lista;
- Depois tente criar alguma forma de excluir a tarefa na lista;

Atividade Prática II

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

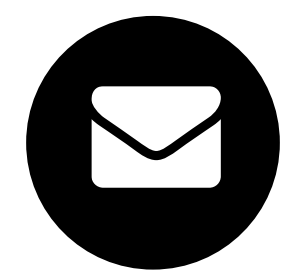


- Conte quantos caracteres, palavras e linhas no texto;
- Transforme o texto em Maiúscula, Minúscula e Primeira Letra em Maiúscula.
- Procure a quantidade de ocorrência de um texto e substitua um texto pelo outro.

Dúvidas?



Dúvidas?



fernando.alves@ifrj.edu.br



ftamberlini.dev.br