

1 Propriedades e métodos de *Strings*

As *strings* representam textos em JavaScript e possuem muitos métodos para manipulação, busca, e transformação.

◆ Principais propriedades:

- **length** → Retorna o tamanho da string

◆ Principais métodos:

- **toUpperCase()** → Converte para letras maiúsculas.
- **toLowerCase()** → Converte para letras minúsculas.
- **includes(texto)** → Verifica se a string contém determinado texto.
- **startsWith(texto)** → Verifica se a string começa com um texto específico.
- **endsWith(texto)** → Verifica se termina com determinado texto.
- **indexOf(texto)** → Retorna a posição da primeira ocorrência do texto.
- **lastIndexOf(texto)** → Retorna a última posição de ocorrência.
- **slice(inicio, fim)** → Retorna parte da string entre as posições informadas.
- **substring(inicio, fim)** → Similar ao slice, mas sem valores negativos.
- **replace(padrao, novoTexto)** → Substitui parte do texto por outro.
- **trim()** → Remove espaços do início e do fim.
- **split(separador)** → Divide a string em partes e cria um array.
- **concat(outraString)** → Junta duas ou mais strings.
- **charAt(posicao)** → Retorna o caractere da posição indicada.
- **repeat(n)** → Repete a string várias vezes.
- **match(regex)** → Procura padrões com expressões regulares.

Exemplo:

```
let texto = " JavaScript ";  
console.log(texto.trim().toUpperCase().includes("JAVA")); // true
```

● 2 Propriedades e métodos de elementos do *DOM*

Os **elementos do DOM** representam partes do HTML.
Eles podem ser manipulados com o JavaScript para alterar o conteúdo, aparência e comportamento da página.

◆ Principais propriedades:

- **innerText** → Texto interno do elemento.
- **innerHTML** → HTML interno (com tags).
- **textContent** → Todo o texto, inclusive oculto.
- **classList** → Lista de classes CSS.
- **id** → Retorna ou define o id do elemento.
- **style** → Acessa os estilos CSS inline.
- **attributes** → Lista de atributos do elemento.
- **value** → Valor de inputs e selects.
- **src, href, alt** → Usadas em imagens e links.

◆ Principais métodos:

- **querySelector(seletor)** → Seleciona o primeiro elemento que corresponde ao seletor.
- **querySelectorAll(seletor)** → Seleciona todos os elementos correspondentes.
- **getElementById(id)** → Seleciona elemento pelo ID.
- **getElementsByClassName(nome)** → Seleciona por classe.
- **getElementsByTagName(tag)** → Seleciona por tag.
- **setAttribute(nome, valor)** → Define um atributo.
- **getAttribute(nome)** → Retorna um atributo.
- **removeAttribute(nome)** → Remove um atributo.
- **classList.add(classe)** → Adiciona uma classe CSS.
- **classList.remove(classe)** → Remove uma classe CSS.
- **classList.toggle(classe)** → Alterna (liga/desliga) uma classe.
- **appendChild(elemento)** → Adiciona um novo elemento dentro do atual.
- **removeChild(elemento)** → Remove um elemento filho.
- **replaceChild(novo, antigo)** → Substitui um elemento filho.
- **addEventListener(evento, função)** → Adiciona um evento (ex: clique).
- **removeEventListener(evento, função)** → Remove o evento.
- **focus()** → Dá foco ao elemento (como em inputs).
- **blur()** → Tira o foco.
- **click()** → Simula um clique programaticamente.

Exemplo:

```
const botao = document.querySelector('button');  
botao.classList.add('ativo');  
botao.addEventListener('click', () => console.log('Clicou!'));
```

3 Métodos para conversão de números

JavaScript permite converter valores (strings, booleanos, etc.) em números de várias formas.

◆ Conversão para número:

- **Number(valor)** → Converte qualquer valor em número.
- **parseInt(valor)** → Converte string para número inteiro.
- **parseFloat(valor)** → Converte string para número decimal.
- **+valor** → Conversão rápida (operador unário).
- **Math.floor()** → Arredonda para baixo.
- **Math.ceil()** → Arredonda para cima.
- **Math.round()** → Arredonda para o inteiro mais próximo.

◆ Conversão de número para string:

- **String(numero)** → Converte número para texto.
- **numero.toString()** → Retorna o número em formato de texto.
- **numero.toFixed(casas)** → Formata número com casas decimais.
- **numero.toPrecision(n)** → Controla o número total de dígitos.

Exemplo:

```
let n = "10.5";
console.log(parseInt(n));    // 10
console.log(parseFloat(n)); // 10.5
console.log(Number(true));   // 1
console.log((10.678).toFixed(2)); // "10.68"
```