

# Funções Nativas, Datas e Manipulação de Dados (4h)

Neste módulo, não vamos apenas criar nossas funções, mas dominar as ferramentas poderosas que o JavaScript já traz prontas. Vamos aprender a manipular números, textos, tempo e listas de dados como profissionais.

## 1. O "Canivete Suíço" Matemático (Math)

O JavaScript possui um objeto global chamado `Math` para operações matemáticas complexas. Não precisamos instanciá-lo, basta usar.

### Principais Métodos

Método	Descrição	Exemplo
<b>Arredondamento</b>		
<code>Math.round(n)</code>	Arredonda para o inteiro mais próximo.	$4.7 \rightarrow 5$
<code>Math.ceil(n)</code>	Arredonda sempre para <b>cima</b> (teto).	$4.1 \rightarrow 5$
<code>Math.floor(n)</code>	Arredonda sempre para <b>baixo</b> (chão).	$4.9 \rightarrow 4$

### Cálculos

<code>Math.pow(x, y)</code>	Potência (x elevado a y).	$2^3 \rightarrow 8$
<code>Math.sqrt(x)</code>	Raiz Quadrada.	$\sqrt{25} \rightarrow 5$
<code>Math.abs(x)</code>	Valor Absoluto (remove o sinal negativo).	$-10 \rightarrow 10$

### Analíticos

<code>Math.min(...)</code>	Retorna o menor número da lista.	<code>Math.min(5, 1, 9) → 1</code>
----------------------------	----------------------------------	------------------------------------

Math.max(...)	Retorna o maior número da lista.	Math.max(5, 1, 9) → 9
---------------	----------------------------------	--------------------------

## Gerando Números Aleatórios (Muito útil para jogos!)

O `Math.random()` gera um número decimal entre 0 e 1 (ex: 0.453...). Para torná-lo útil, precisamos de um truque matemático:

```
// Gerar um número inteiro aleatório entre 1 e 10
const aleatorio = Math.floor(Math.random() * 10) + 1;
console.log(aleatorio);
```

## 2. Manipulação de Strings (Textos)

Strings são cadeias de caracteres. No dia a dia, precisamos limpar, formatar e verificar textos vindos de formulários.

```
const mensagem = " JavaScript é Incrível! ";

// 1. Tamanho (inclui espaços)
console.log(mensagem.length); // 28

// 2. Limpeza (Remove espaços do início e fim - Trim)
console.log(mensagem.trim()); // "JavaScript é Incrível!"

// 3. Maiúsculas e Minúsculas (Normalização)
const textoLimpido = mensagem.trim();
console.log(textoLimpido.toUpperCase()); // "JAVASCRIPT É INCRÍVEL!"
console.log(textoLimpido.toLowerCase()); // "javascript é incrível!"

// 4. Substituição (Replace)
console.log(textoLimpido.replace("Incrível", "Poderoso")); // "JavaScript é Poderoso!"

// 5. Cortar texto (Slice)
// slice(inicio, fim)
console.log(textoLimpido.slice(0, 10)); // "JavaScript"

// 6. Transformar em Array (Split)
const tags = "JS,Frontend,React";
console.log(tags.split(",")); // ["JS", "Frontend", "React"]

// 7. Verificar conteúdo (Includes)
```

```
console.log(textoLimpo.includes("Java")); // true
```

---

## 3. Dominando o Tempo (Date)

O objeto `Date` controla datas e horários.

**A Pegadinha do Mês:** No JavaScript, os meses começam em **0 (Janeiro)** e vão até **11 (Dezembro)**.

### Criando Datas

```
const agora = new Date(); // Data e hora exata do momento  
const dataEspecifica = new Date("2025-12-25"); // Natal de 2025
```

### Obtendo Informações (Getters)

```
console.log(agora.getFullYear()); // Ano (ex: 2025)  
console.log(agora.getMonth()); // Mês (0 a 11) - Lembre-se de somar +1 para exibir  
console.log(agora.getDate()); // Dia do mês (1 a 31)  
console.log(agora.getDay()); // Dia da semana (0 = Domingo, 1 = Segunda...)
```

### Formatando Datas (O jeito moderno: Intl)

Em vez de montar a string na mão, use o formatador internacional:

```
const opcoes = { year: 'numeric', month: 'long', day: 'numeric' };  
console.log(agora.toLocaleDateString('pt-BR', opcoes));  
// Saída: "7 de fevereiro de 2025"
```

---

## 4. Arrays (Listas de Dados)

Arrays são estruturas fundamentais para armazenar coleções. Pense neles como uma lista de compras.

### Operações Básicas

```
const techs = ["HTML", "CSS", "JS"];  
  
// Adicionar  
techs.push("React"); // No final -> ["HTML", "CSS", "JS", "React"]
```

```
techs.unshift("Git"); // No início -> ["Git", "HTML", "CSS", "JS", "React"]

// Remover
techs.pop(); // Remove o último ("React")
techs.shift(); // Remove o primeiro ("Git")

// Encontrar índice
const posicaoJS = techs.indexOf("JS"); // 2
```

---

## 5. Métodos Avançados de Arrays (High Order Functions)

É aqui que o JavaScript brilha. Esses métodos são essenciais para React, Node.js e manipulação de dados.

### map () - Transforma os dados

Cria um **novo array** do mesmo tamanho, modificando os itens.

```
const numeros = [10, 20, 30];
const dobro = numeros.map(num => num * 2);

console.log(dobro); // [20, 40, 60]
```

### filter () - Filtra os dados

Cria um novo array **apenas com os itens que passam no teste**.

```
const idades = [15, 22, 18, 40, 12];
const maioresDeldade = idades.filter(idade => idade >= 18);

console.log(maioresDeldade); // [22, 18, 40]
```

### reduce () - Reduz a um único valor

Soma totais, agrupa dados, etc.

```
const precos = [10.50, 20.00, 5.00];
// acumulador começa em 0
const total = precos.reduce((acumulador, valorAtual) => acumulador + valorAtual, 0);

console.log(total); // 35.5
```

## `sort()` - Ordenação (Atenção!)

Por padrão, o `sort()` ordena como texto. Para números, precisamos de uma função auxiliar.

```
const lista = [10, 1, 5, 20];

// Errado (ordem alfabética): [1, 10, 20, 5]
// Certo (ordem numérica):
lista.sort((a, b) => a - b);

console.log(lista); // [1, 5, 10, 20]
```

---

## 6. Matrizes (Arrays Multidimensionais)

Matrizes são arrays dentro de arrays. Imagine uma tabela do Excel ou um tabuleiro de xadrez.

```
// Matriz 3x3
const tabela = [
  [1, 2, 3], // Linha 0
  [4, 5, 6], // Linha 1
  [7, 8, 9] // Linha 2
];

// Acessando o número 6 (Linha 1, Coluna 2)
console.log(tabela[1][2]);

// Percorrendo a matriz inteira
for (let i = 0; i < tabela.length; i++) {
  for (let j = 0; j < tabela[i].length; j++) {
    console.log(`Elemento [${i}][${j}] = ${tabela[i][j]}`);
  }
}
```

---

## Exercícios Práticos ("Mão na Massa")

### Desafio 1: O Formatador de Nomes

Crie um array com nomes mal formatados: [" João ", " MaRIA ", " PEDRO "].

Use `map()` e métodos de string para criar um novo array onde todos os nomes estejam:

1. Sem espaços extras.

2. Apenas com a primeira letra maiúscula (Capitalizados).

*Resultado esperado:*

## Desafio 2: Estatísticas de Notas

Dado o array de notas: [7.5, 8.0, 4.5, 9.0, 3.5, 6.0].

1. Use `filter` para criar um array apenas com os **Aprovados** (nota  $\geq 6$ ).
2. Use `reduce` para calcular a **Média da turma** (soma de todas as notas dividida pelo total de alunos).
3. Use `Math.max` e `Math.min` (dica: use o spread operator . . .) para exibir a maior e menor nota.

## Desafio 3: Gerenciador de Notas (Matriz)

Crie uma matriz que represente 3 alunos e suas 3 notas:

```
const notasAlunos = [  
  [8, 7, 9], // Aluno A  
  [4, 5, 6], // Aluno B  
  [9, 10, 9] // Aluno C  
];
```

Use loops aninhados para calcular a média de cada aluno e exibir no console:

"Aluno 0: Média 8.0 - Aprovado"

"Aluno 1: Média 5.0 - Reprovado"

---

## Dicas de Solução (Desafio 1)

```
const nomesBaguncados = [" joão", "MaRIA ", " PEDRO "];  
  
const nomesFormatados = nomesBaguncados.map(nome => {  
  const nomeLimpo = nome.trim().toLowerCase();  
  // Pega a 1ª letra maiúscula + o resto da palavra  
  return nomeLimpo.charAt(0).toUpperCase() + limpo.slice(1);  
});
```

```
console.log(nomesFormatados);
```

## Dicas de Solução (Desafio 2)