

# Lógica de Programação com Java Script

Othon Oliveira

SENAC - PROA



# Sumário

- 1 Entrada de dados
- 2 Funções Especiais
- 3 Calculos com um pouco de desafios
- 4 Calculos com mais um pouco de desafios
- 5 Java Scritp exibindo mensagens diretamente no navegados
- 6 HTML com Java Scritp



# Introdução a Lógica de Programação com Java Script

## Entrada de dados em Java Script

### Leitura de dados através de prompt

- a. O Java Script trata a entrada de dados como tipo texto (string)



# Introdução a Lógica de Programação com Java Script

## Entrada de dados em Java Script

### Leitura de dados através de prompt

- a. O Java Script trata a entrada de dados como tipo texto (string)
- b. Uma maneira de converter os dados a entrada em número: `parseInt()`



# Introdução a Lógica de Programação com Java Script

## Entrada de dados em Java Script

### Leitura de dados através de prompt

- a. O Java Script trata a entrada de dados como tipo texto (string)
- b. Uma maneira de converter os dados a entrada em número: `parseInt()`
- c. A função também trata se os dados não forem numéricos: (NaN)



# Introdução a Lógica de Programação com Java Script

## Entrada de dados em Java Script

### Leitura de dados através de prompt

- a. O Java Script trata a entrada de dados como tipo texto (string)
- b. Uma maneira de converter os dados a entrada em número: `parseInt()`
- c. A função também trata se os dados não forem numéricos: (NaN)
- d. Dessa forma, mostrada acima, a entrada de dados já tem um tratamento especial



# JavaScript

## Introdução

As funções `parseInt()` e `parseFloat()` são usadas para converter strings em valores numéricos, mas com algumas diferenças importantes.

Além dessas, existem outras funções que também podem ser usadas para converter valores ou realizar outras operações.

Vamos entender quando usar `parseInt`, `parseFloat` e outras funções relevantes:



## parseInt() - Conversão para Inteiro:

- Use parseInt() quando você quiser converter uma string em um valor inteiro
- Qualquer parte da string que não seja um número será ignorada
- Exemplo: parseInt("123") retorna 123





## parseFloat() - Conversão para Ponto Flutuante:

- Use parseFloat() quando você quiser converter uma string em um valor de ponto flutuante.
- Essa função lida com números decimais.
- Exemplo: parseFloat("3.14") retorna 3.14



## Number() - Conversão Genérica:

- A função `Number()` pode ser usada para converter tanto para inteiro quanto para ponto flutuante.
- Ela lida bem com números decimais e números em notação científica.
- Exemplo: `Number("42")` retorna 42, `Number("3.14")` retorna 3.14.



## isNaN() - Verificação de NaN:

- 'isNaN()' verifica se um valor é NaN (Not-a-Number).
- Pode ser usado para validar se uma conversão de string para número foi bem-sucedida.
- Exemplo: `isNaN(parseInt("abc"))` retorna true.



## toFixed() - Arredondamento Decimal:

- A função `toFixed()` é usada para arredondar um número para um número específico de casas decimais.
- Exemplo: `const numero = 3.14159; numero.toFixed(2)` retorna `"3.14"`.



## Math.floor() e Math.ceil() - Arredondamento para Baixo/Cima:

- Math.floor() arredonda um número para baixo para o inteiro mais próximo.
- Math.ceil() arredonda um número para cima para o inteiro mais próximo.
- Exemplo: Math.floor(4.8) retorna 4, Math.ceil(4.2) retorna 5.



## Math.random() - Geração de Números Aleatórios:

- `Math.random()` gera um número decimal aleatório entre 0 (inclusive) e 1 (exclusivo).
- Pode ser usado com outros métodos matemáticos para gerar números aleatórios em faixas específicas.
- Exemplo: `Math.random()` retorna um número aleatório entre 0 e 1.



# Cálculos em JavaScript

## Usando Funções de Entrada

```
// Exemplo 1: Soma de Dois Números  
const num1 = parseFloat(prompt("Digite o primeiro número:"));  
const num2 = parseFloat(prompt("Digite o segundo número:"));  
const soma = num1 + num2;  
console.log("A soma é:", soma);
```

Como se executa (rodar) esse código?



```
// Exemplo 3: Multiplicação de Números  
const num1 = parseFloat(prompt("Digite o primeiro número:"));  
const num2 = parseFloat(prompt("Digite o segundo número:"));  
const multiplicacao = num1 * num2;  
console.log("A multiplicação é:", multiplicacao);
```

Como se executa (rodar) esse código?





# Fórmulas Matemáticas

## Média Aritmética

A média aritmética é calculada pela soma de todos os valores dividida pelo número de valores.

$$\text{Média Aritmética} = \frac{\text{Soma dos Valores}}{\text{Número de Valores}}$$



## Apresentando a Média Aritmética

A média aritmética é uma medida de centralidade que representa o valor médio de um conjunto de números.

Para calcular a média aritmética:

- 1 Some todos os valores do conjunto.
- 2 Divida a soma pelo número total de valores no conjunto.

Isso é expresso pela fórmula:

$$\text{Média Aritmética} = \frac{v_1 + v_2 + v_3 + \dots + v_N}{N}$$

onde  $v_i$  é cada valor individual do conjunto e  $N$  é o número total de valores.



## Exemplo de Uso da Média Aritmética

Vamos calcular a média aritmética de um conjunto de valores.

Vamos calcular a média aritmética dos valores 10, 15, 20, 25 e 30.

```
\begin{verbatim}const valor1 = 10, valor2 = 15, valor3 = 20;
const valor4 = 25, valor5 = 30;

const soma = valor1 + valor2 + valor3 + valor4 + valor5;
const quantidade = 5;
const media = soma / quantidade;

console.log("Soma:", soma);
console.log("Média Aritmética:", media);
```

Pergunta: Onde deverei "colar" esse código, para que funcione do jeito que está?

## Apresentando a Média Ponderada

A média ponderada é uma medida de centralidade que leva em consideração diferentes pesos atribuídos a cada valor no cálculo da média.

Para calcular a média ponderada:

- 1 Multiplique cada valor pelo seu respectivo peso.
- 2 Some os produtos obtidos.
- 3 Divida a soma pelo total dos pesos.

Isso é expresso pela fórmula:

$$\text{Média Ponderada} = \frac{v_1 \cdot p_1 + v_2 \cdot p_2 + v_3 \cdot p_3 + \dots}{p_1 + p_2 + p_3 + \dots}$$

onde  $v_i$  é cada valor individual do conjunto e  $p_i$  é o peso correspondente a esse valor.

## Exemplo de Uso da Média Ponderada

Vamos calcular a média ponderada das notas de um aluno.

```
const nota1 = 7.5, peso1 = 2;
```

```
const nota2 = 9.0, peso2 = 3;
```

```
const produto1 = nota1 * peso1;
```

```
const produto2 = nota2 * peso2;
```

```
const somaDosProdutos = produto1 + produto2;
```

```
const somaDosPesos = peso1 + peso2;
```

```
const mediaPonderada = somaDosProdutos / somaDosPesos;
```

```
console.log("Nota 1:", nota1);
```

```
console.log("Peso 1:", peso1);
```

```
console.log("Nota 2:", nota2);
```

```
console.log("Peso 2:", peso2);
```



# Cálculos Desafiante em JavaScript

## Desafio 1: Média Ponderada

```
// Desafio: Calcular média ponderada de notas
const nota1 = parseFloat(prompt("Digite a nota 1:"));
const peso1 = parseFloat(prompt("Digite o peso da nota 1:"));
const nota2 = parseFloat(prompt("Digite a nota 2:"));
const peso2 = parseFloat(prompt("Digite o peso da nota 2:"));

const mediaPonderada = (nota1 * peso1 + nota2 * peso2) / (peso1 + peso2);
console.log("Média ponderada:", mediaPonderada);
```

Pergunta: Onde deverei "colar" esse código, para que funcione do jeito que está?



## Apresentando o Cálculo de Juros Simples

Os juros simples são uma forma de cálculo de juros onde o valor dos juros é calculado apenas sobre o valor inicial do investimento.

Para calcular os juros simples:

- 1 Multiplique o principal pelo percentual de taxa de juros.
- 2 Multiplique o resultado pelo período.

Isso é expresso pela fórmula:

$$\text{Montante} = \text{Principal} + \text{Principal} \times \frac{\text{Taxa de Juros}}{100} \times \text{Período}$$

onde o montante é o valor acumulado após um determinado período, o principal é o valor inicial do investimento, a taxa de juros é dada em percentual e o período é o número de vezes que os juros são aplicados.

## Exemplo de Cálculo de Juros Simples

Vamos calcular o montante com juros simples de um empréstimo.

```
const principal = 5000;  
const taxaDeJuros = 10; // 10%  
const periodo = 2;  
  
const montante = principal + (principal * taxaDeJuros / 100 *  
  
console.log("Principal:", principal);  
console.log("Taxa de Juros:", taxaDeJuros, "%");  
console.log("Período:", periodo, "anos");  
console.log("Montante com Juros Simples:", montante.toFixed(2))
```





## Apresentando o Cálculo de Juros Compostos

Os juros compostos são uma forma de cálculo de juros onde o valor dos juros é calculado sobre o montante acumulado em cada período.

Para calcular os juros compostos:

- 1 Multiplique o montante anterior pela taxa de juros (em decimal).
- 2 Some o montante inicial com o valor dos juros calculados.

Isso é expresso pela fórmula:

$$\text{Montante} = \text{Principal} \times \left( 1 + \frac{\text{Taxa de Juros}}{100} \right)^{\text{Período}}$$

onde o montante é o valor acumulado após um determinado período, a taxa de juros é dada em percentual e o período é o número de vezes que os juros são aplicados.

# Cálculos Desafiante em JavaScript

## Desafio 2: Juros Compostos

```
// Desafio: Calcular montante com juros compostos
const principal=parseFloat(prompt("Digite o valor principal:"))
const taxaDeJuros=parseFloat(prompt("Digite a taxa de juros (%"))
const anos = parseInt(prompt("Digite o número de anos:"));

const montante = principal * Math.pow(1 + taxaDeJuros / 100, anos)
console.log("Montante com juros compostos:", montante.toFixed(2))
```

Pergunta: Onde deverei "colar" esse código, para que funcione do jeito que está?



# Cálculos Desafiantes em JavaScript

## Desafio 3: Fatorial

Vamos calcular o fatorial de um número inteiro e exibir o resultado em uma janela. Exemplo:  $4! = 4 \cdot 3 \cdot 2 \cdot 1$

```
const numero = parseInt(prompt("Digite um número inteiro:"));

let fatorial = 1;
for (let i = 1; i <= numero; i++) {
  fatorial *= i;
}

const resultado = `Fatorial de ${numero} é ${fatorial}`;
alert(resultado);
```

Atenção: Este slide contém coisas **"perigosas"**, uso de estruturas de repetição, que não vimos ainda!!



## Usando document.write()

O método `document.write()` é utilizado para exibir conteúdo diretamente no navegador.

É útil para mostrar resultados, mensagens ou saídas de cálculos no contexto da página web.



## Exemplo de document.write()

Vamos usar o `document.write()` para mostrar uma mensagem na página.

```
document.write("Olá, mundo!");
```



# Usando HTML para mostrar código Java Script

comandos "Alert" e Console log

```
<script>  
    alert("Bem vindo ao mundo Java Script!")  
    console.log("Olá, Mundo JavaScript")  
</script>
```



## Exemplo de Cálculo de Juros Simples

Aqui está um exemplo de cálculo de juros simples em HTML:

```
<!DOCTYPE html>
<html>
<head> <title>Cálculo de Juros Simples</title> </head>
<body>
  <h1>Calculadora de Juros Simples</h1>
  <script>
    const principal = 5000;
    const taxaDeJuros = 10; // 10%
    const periodo = 2;
    const montante = principal + (principal * taxaDeJuros / 100 * periodo);
    const resultado = 'Montante com Juros Simples: $$' + montante;
    document.write(resultado);
  </script>
</body>
</html>
```



## Exemplo de Cálculo de Juros Simples

Aqui está um exemplo de cálculo de juros simples em HTML com prompts de comando:

```
<!DOCTYPE html>
<html><head>
  <title>Cálculo de Juros Simples</title></head>
<body>
  <h1>Calculadora de Juros Simples</h1>
  <script>
    const principal = parseFloat(prompt("Digite o valor principal"));
    const taxaDeJuros = parseFloat(prompt("Digite a taxa de juros"));
    const periodo = parseFloat(prompt("Digite o período (anos)"));
    const montante = principal + (principal * taxaDeJuros / 100 * periodo);
    const resultado = `Montante com Juros Simples: $$${montante}`;
    document.write(resultado);
  </script>
</body>
```

