



Lógica de Programação - Aula 2

Variáveis, Tipos de Dados e Boas Práticas em JavaScript

Instrutor: MSc Sandir Rodrigues Campos | Data: 02/02/2026

Objetivos da Aula

O que você vai aprender hoje?

1. Diferenciar `console.log()` e `document.write()`;
2. Declarar variáveis com `var`, `let` e `const`;
3. Entender escopo e boas práticas;
4. Identificar tipos de dados primitivos;
5. Usar o operador `typeof`;
6. Resolver problemas práticos.

Fundamentos: O que é Programação?

Programação é a arte de **dar instruções a um computador.**

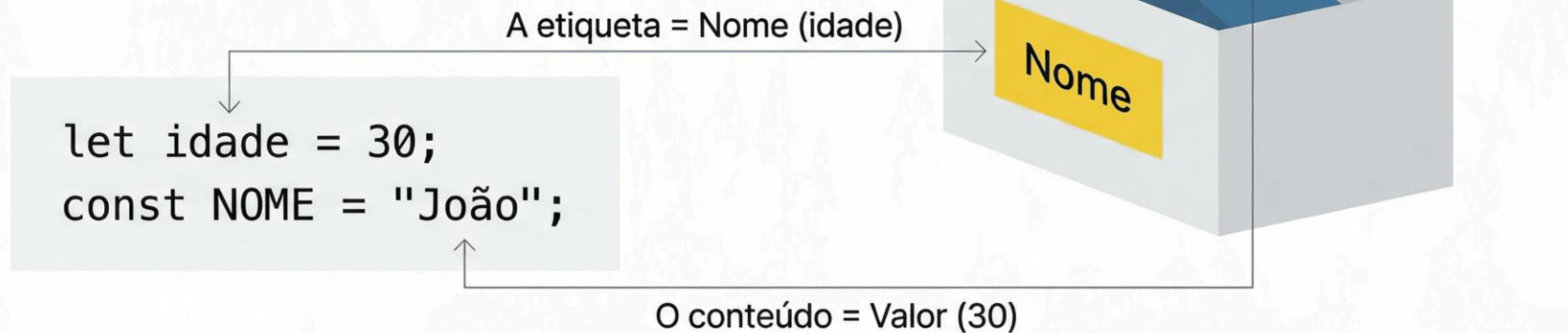
Assim como você segue uma receita para fazer um bolo, um computador segue um conjunto de instruções.

Automatizar tarefas • Resolver problemas • Criar ferramentas



O que são Variáveis?

Uma variável é um espaço na memória do computador reservado para guardar um valor.



Slide 6: Tipos de Dados Primitivos

Tipos de Dados: O que você pode guardar?

Os tipos de dados definem o que pode ser armazenado nas variáveis.



String (Texto)

Sequência de caracteres entre aspas.

"João da Silva", 'Olá'



Number (Número)

Inteiros ou decimais.

25, 3.14, -10



Boolean (Booleano)

Verdadeiro ou Falso.

true, false

Slide 7: Exibindo Informações - console.log()

Exibindo Informações: console.log()

A ferramenta essencial para desenvolvedores.



Conteúdo Principal:

- **Definição:** Exibe mensagens no **console do navegador**.
- **Quando usar:**
 - Debug e rastreamento de erros
 - Verificar valores de variáveis
 - Comunicação técnica (dev-to-dev)

```
console.log('Esta mensagem aparece no console');  
console.log('Valor de idade:', 30);
```

Dica de Atalho:

- **Windows/Linux:** F12 ou Ctrl+Shift+I
- **Mac:** Cmd+Option+I

Exibindo Informações: **document.write()**

Definição: Exibe mensagens **diretamente na página HTML.**

Quando usar:

- Para exibir conteúdo **ao usuário final**
- Para criar elementos HTML **dinamicamente**
- Para interação com o usuário

```
document.write('<h1>Bem-vindo!</h1>');  
document.write('<p>Esta mensagem  
aparece na página.</p>');
```



Slide 9: Comparação - `console.log()` vs `document.write()`

Qual usar? `console.log()` vs `document.write()`

Aspecto	<code>console.log()</code>	<code>document.write()</code>
Destino	Console do navegador	Página HTML
Público	Desenvolvedores	Usuários finais
Uso	Debug e testes	Conteúdo visível
Visibilidade	Oculto por padrão	Visível na página

✓ **Regra de Ouro: Use `console.log()` para debug.
Use `document.write()` para exibir ao usuário.**

Slide 10: Declarando Variáveis - var, let e const

Três formas, comportamentos diferentes.

Três formas, comportamentos diferentes.

var

✗ Evitar

Escopo global/função.

Hoisting confuso.

Comportamento antigo.

let

✓ Use quando mudar

Escopo de bloco.

Permite reatribuição.

Seguro.

const

✓ Use por padrão

Escopo de bloco.

Não muda (constante).

Mais seguro.

Exemplo Prático: Declarando Variáveis

```
// VAR (evitar)
var nome = 'João';
nome = 'Maria'; // Permitido
console.log(nome); // Saída: Maria

// LET (use quando precisar mudar)
let idade = 30;
idade = 31; // Permitido
console.log(idade); // Saída: 31

// CONST (use por padrão)
const CPF = '123.456.789-00';
// CPF = '000.000.000-00'; // ✗ ERRO!
console.log(CPF); // Saída: 123.456.789-00
```

Slide 12:

Escopo: Onde a Variável é Acessível?

Escopo de Variáveis

Analogia



Global (Sala de Estar)

Qualquer pessoa na casa pode ver o sofá.
(Acessível a todos)

Local (Quarto)

Apenas você pode acessar seu travesseiro.
(Privado/Restrito)

No Código

- **Escopo Global**
Variável acessível em **todo o programa**.
- **Escopo Local**
Variável acessível apenas dentro de um **bloco específico { ... }**.

Regra de Ouro

Use `let` e `const` para criar variáveis com escopo de bloco (mais seguro).

Descobrimos o Tipo: Operador typeof

O operador **'typeof'** retorna uma **string** com o tipo de **dado** da variável.



```
const nomeCompleto = 'Ana Silva';  
const anoNascimento = 1990;  
const ehAluna = true;  
  
console.log(typeof nomeCompleto); // Saída: "string"  
console.log(typeof anoNascimento); // Saída: "number"  
console.log(typeof ehAluna); // Saída: "boolean"
```

Por que é importante

- Ajuda a debugar erros
- Valida dados antes de processar
- Entende o que o programa está manipulando

Boas Práticas

Escrevendo código limpo e seguro.

- ✓ **Use `const` por padrão**
Mais seguro, evita mudanças acidentais.
- ✓ **Use `let` quando a variável vai mudar**
Necessário para loops e contadores.
- ✗ **Nunca use `var`**
Comportamento confuso e imprevisível.
- ✓ **Nomes descritivos**
 - ✓ `const idadeAluno = 25;`
 - ✗ `const a = 25;`
- ✓ **Constantes em MAIÚSCULA**
`const TAXA_JUROS = 0.05;`

Exercício 1: Informações do Usuário



Enunciado (Texto Principal)

Crie uma página HTML que declare três variáveis: nome, cidade e profissao.

Formato especificado:

"Olá, meu nome é [nome], moro em [cidade] e sou [profissao]."

Dicas

- Use ``const`` para declarar as variáveis
- Use ``document.write()`` para exibir na página
- Use concatenação de *strings* com ``+``

Exercício 2:

Calculadora de Idade

1. Declare uma variável **anoNascimento** com seu ano de nascimento.
2. Declare outra variável **anoAtual** com o ano atual.
3. Calcule sua idade e armazene em uma variável **idade**.
4. Exiba a idade no console com **console.log()**.

Dicas

- Use operações aritméticas (-)
- Use **console.log()** para exibir no console
- Exemplo: **idade = 2026 - 1995;**

Slide 17: Exercício 3 - Verificador de Tipo

Conteúdo Principal (Enunciado):

Para cada uma das variáveis, use o operador `typeof` para obter o tipo de dado com o operador `typeof`.

```
let produto = "Notebook";  
let preco = 4999.90;  
let disponivel = true;
```

Tarefa: Para cada variável, use `console.log()` para exibir o tipo de dado correspondente usando o operador `typeof`.

Dicas

- Use o operador `typeof`
- Exiba uma mensagem clara no console
- Exemplo:
`console.log(typeof produto);`

Exercício 4:

let vs const

1. Declare uma variável **let** chamada **pontuacao** com valor **100**
2. Reatribua um novo valor de **150** para **pontuacao**
3. Exiba o valor final no console
4. Declare uma variável **const** chamada **JOGADOR** com seu nome
5. Tente reatribuir um novo nome (observe o erro)

Dicas

- **let** permite reatribuição
- **const** não permite reatribuição
- O erro esperado:
TypeError:
Assignment to
constant variable



Desafio: Calculadora de Compras

1. Declare `precoProduto` com valor 129.99
2. Declare `quantidade` com valor 3
3. Calcule o `subtotal` (`preço × quantidade`)
4. Declare `const DESCONTO_PERCENTUAL = 0.10` (10%)
5. Calcule o `valorDoDesconto` (`subtotal × percentual`)
6. Calcule o `precoFinal` (`subtotal - desconto`)
7. Exiba no console: Subtotal, Valor do Desconto e Preço Final

Dicas

- Combine múltiplos conceitos da aula
- Use operações aritméticas
- Exiba mensagens claras no console

Troubleshooting: Erros Comuns

Seção 1: ReferenceError

ReferenceError: minhaVariavel is not defined

Causa: Tentou usar uma variável que não foi declarada.

✓ **Solução:** Declare a variável com ``let`` ou ``const`` antes de usar.

Seção 2: TypeError

TypeError: Assignment to constant variable

Causa: Tentou mudar o valor de uma variável ``const``.

✓ **Solução:** Use ``let`` se a variável precisar mudar de valor.

Slide 21: Troubleshooting - Mais Erros

Mais Erros Comuns

SyntaxError

SyntaxError: Unexpected token

- **Erro:** SyntaxError: Unexpected token
- **Causa:** String sem fechar as aspas.
- **Solução:** Feche todas as aspas: "texto" ou 'texto'.

Confundindo Tipos

```
let r1 = "10" + 5; // "105"  
let r2 = 10 + 5;   // 15
```

- **Causa:** Misturar strings com números.
- **Solução:** Verifique os tipos com typeof.

Slide 22: Resumo da Aula

Resumo: Os Pontos-Chave

- ✓ **Exibir informações:** `console.log()` vs `document.write()`
- ✓ **Declarar variáveis:** **const** (padrão) vs **let** vs **var**
- ✓ **Tipos de dados:** String, Number, Boolean
- ✓ **Verificar tipos:** Operador **typeof**
- ✓ **Escopo:** Global vs Local
- ✓ **Boas práticas:** Nomes descritivos & **const** por padrão

Slide 23: Próximos Passos

Próximos Passos



Recursos para Aprofundamento

- MDN Web Docs: JavaScript
- W3Schools: JavaScript Tutorial



Próxima Aula

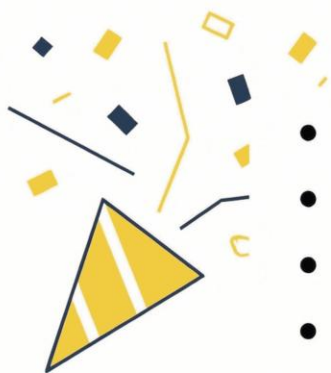
- Estruturas Condicionais (if/else)
- Loops (for, while)
- Funções em JavaScript



Desafio até a próxima aula

- Resolva todos os exercícios propostos
- Tente criar suas próprias variáveis e operações
- Explore o console do navegador

Parabéns! Você completou a Aula 2!



Você agora entende:

- Como **armazenar** informações em **variáveis**
- As diferenças entre ``var``, ``let`` e ``const``
- Como trabalhar com tipos de dados
- Como exibir informações ao usuário



Próxima etapa: Aplicar esses conceitos em projetos práticos!

Dúvidas? Abra o console (F12) e comece a experimentar!