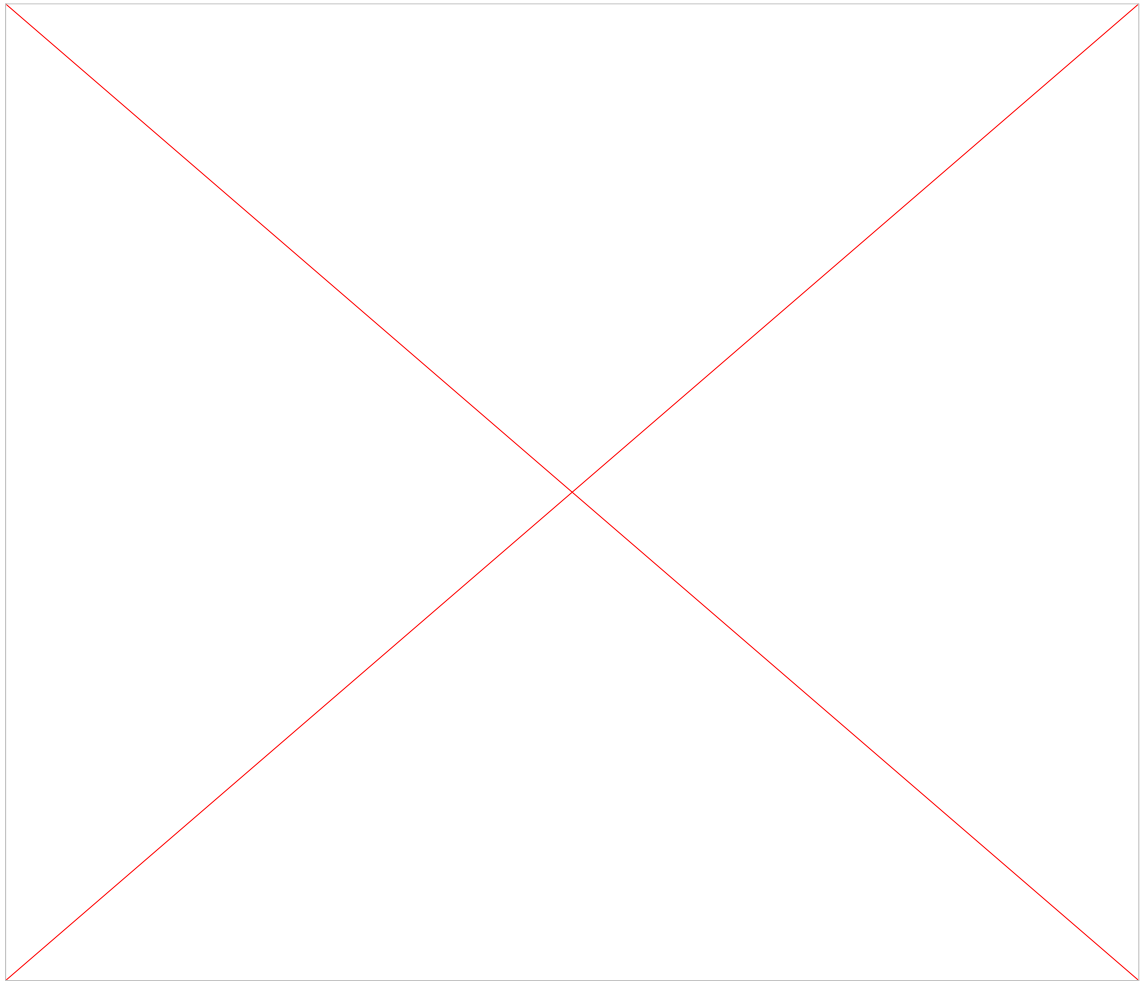
Introdução a Lógica de Programação com JavaScript

Professor: Renato William R. de Souza
IFCE Campus Boa Viagem
Curso de Análise e desenvolvimento de Sistemas
Turma - 2024.1

Criação de páginas WEB

- HTML - Responsavél pelo Conteúdo.
- CSS - Apresentação tornando essas páginas atraentes, definindo a aparência visual.
- JavaScript é a linguagem de programação que adiciona interatividade e dinamismo.



Exemplos

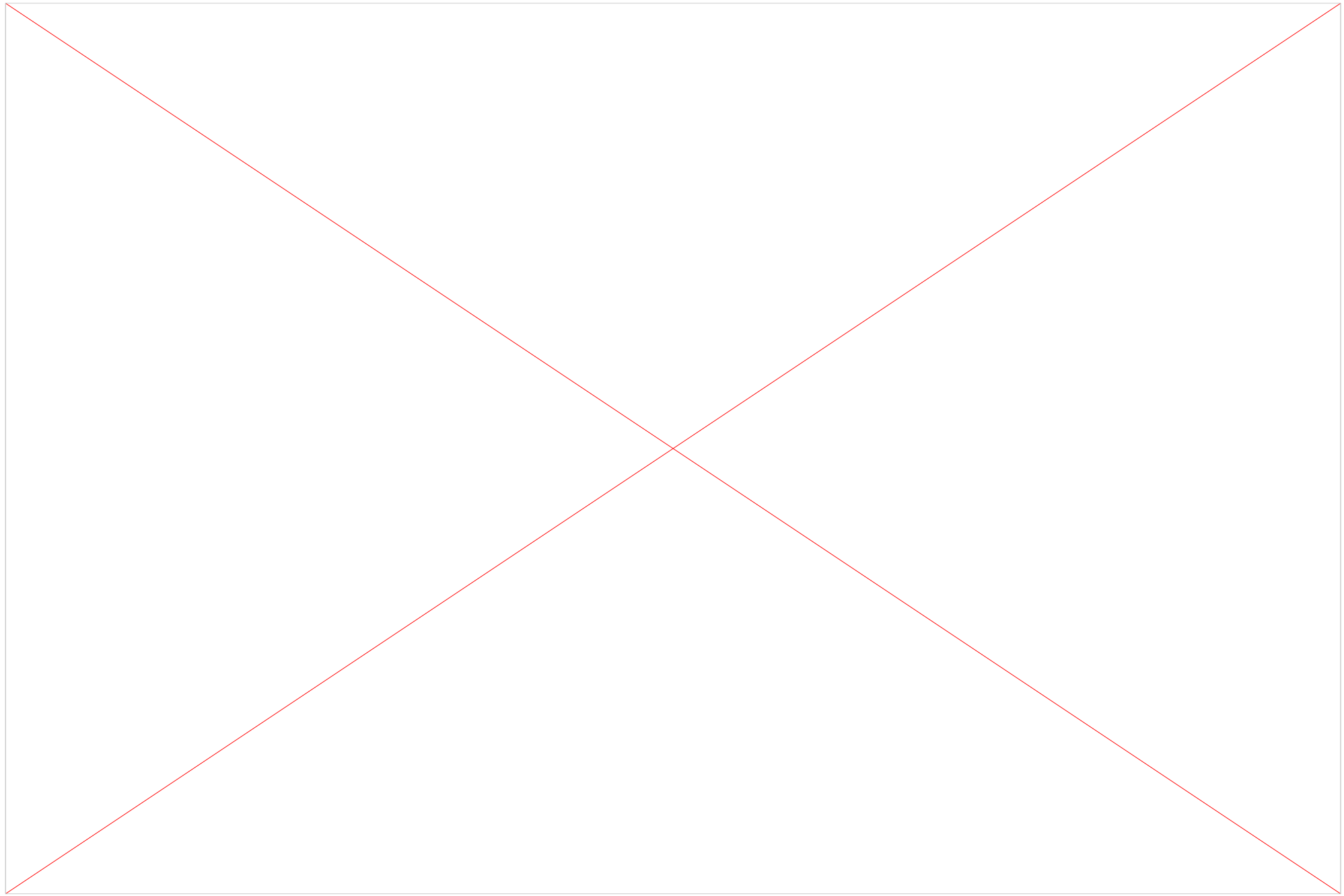
- Validação de Formulários;
- Galeria de Fotos;
- Janelas de Avisos;
- Banners Rotativos;
- Janelas de Publicidade;
- Menu Pop up

O que é JavaScript?

- JavaScript é uma linguagem de programação de alto nível, interpretada e baseada em protótipos.
- É uma das principais tecnologias da web, ao lado de HTML e CSS, e é usada para criar páginas web interativas e dinâmicas.
- Originalmente desenvolvido para rodar no lado do cliente (navegador do usuário), o JavaScript agora também é amplamente utilizado no lado do servidor, graças ao Node.js.

Onde o JavaScript é usado hoje?

- Desenvolvimento Web (Front-end) - React, Angular e Vue.js ;
- Desenvolvimento Web (Back-end) - Node.js;
- Aplicativos Móveis - React Native e Ionic;
- Desenvolvimento de Jogos - Babylon.js e Three.js;
- Aplicações Desktop - Electron;
- Internet das Coisas (IoT) - Johnny-Five e plataformas como Node-RED;



Processo de desenvolvimento WEB

- Há linguagens que rodam do lado cliente e e que rodam no lado servidor
  - Java Script é utilizado pricipalmente para rodar do lado cliente
- O que significa rodar do lado cliente?
  - Significa que o próprio navegador web deve conter as funcionalidades capazes de interpretar o código JavaScript e executá-lo.
- Rodando do lado servidor
  - São executadas em um programa instalado no servidor WEB.

Paradigma

- Suporta os elementos de sintaxe de programação estruturada da linguagem C
  - (por exemplo, if, while, switch)
- Baseadas em Objetos
  - Isso permite que os desenvolvedores modelem conceitos do mundo real e suas interações de forma mais intuitiva e modular.
- Tipagem Dinâmica
  - você pode atribuir valores de diferentes tipos à mesma variável ao longo do tempo sem precisar declarar explicitamente o tipo da variável.

Editores de Código JavaScript

1. Visual Studio Code (VS Code)
2. Sublime Text
3. Atom
4. WebStorm
5. Brackets
6. Notepad++
7. Editores on-line - https://www.w3schools.com/

Onde escrever o código JavaScript

```
1  <html>
2
3      <head>
4
5          <title>Meu primeiro Código JavaScript </title>
6
7          <script>
8            alert("Bem-Vindo ao Mundo JavaScript!")
9            console.log("Meu primeiro programa...")
10         </script>
11     </head>
12
13     <body>
14
15     </body>
16
17 </html>
```

JavaScript INLINE

Segunda Maneira de Escrever um código JavaScript

pagina.html

```
<html>
  <head>
    <title>Aulas de Javascript</title>
    <script src="arquivo.js"></script>
  </head>
  <body>

  </body>
</html>
```

arquivo.js

```
var nome  = "André Fontenelle";
var idade = 20;
var fumante = false;
```

Variáveis e Constantes

Em JavaScript, variáveis e constantes são usadas para armazenar dados que podem ser manipulados ao longo do programa.

As palavras-chave ***var, let e const*** são usadas para declarar variáveis e constantes, cada uma com suas próprias características.

VAR

A palavra chave VAR é utilizada geralmente quando se utiliza o escopo Global

```
<script>
  {
      var meuNome = "Renato William";
  }
alert(meuNome)
```

CONST

Aqui é utilizado para valores que não serão alterados, ou seja, são valores que não irão mudar ao longo do código - São Constantes

Funcionam no escopo Local

```
<script>

  {

    const meuNome = "Renato William";
    let minhaAltura = 1.70

    alert(meuNome)


  }

</script>
```

LET

Usado para declarar variáveis com escopo de bloco

```
<script>
  {
    let minhaAltura = 1.70
    alert(minhaAltura)

  }
```

Regras Básicas para Nomeação de Variáveis

- Caracteres Válidos:
  - Pode conter letras (a-z, A-Z), números (0-9), underscores (_) e cifrões ($).
  - Deve começar com uma letra, underscore ou cifrão. Não pode começar com um número.
- Case Sensitivity:
  - JavaScript diferencia maiúsculas de minúsculas, então nome e Nome são variáveis diferentes.
- Palavras Reservadas:
  - Não pode usar palavras reservadas de JavaScript como var, let, const, if, else, for, while, function, entre outras, como nomes de variáveis.

Exemplos de Nomes Válidos e Inválidos

Válidos

```javascript
let nome;
let _nome;
let $nome;
let nomeCompleto;
let nome1;
let $nome_2;
```

Inválidos

```javascript
let 1nome;        // Não pode começar com um número
let nome-completo; // Hífen não é permitido
let function;     // Palavra reservada
```

Convenções de Nomeação

1. Camel Case:
    1. A primeira palavra começa com letra minúscula e cada palavra subsequente começa com letra maiúscula.
    2. Exemplo: minhaVariavel, nomeCompleto, idadeDoUsuario
2. Pascal Case:
    1. Todas as palavras começam com letra maiúscula. Usada frequentemente para nomear classes.
    2. Exemplo: MinhaClasse, NomeCompleto, IdadeDoUsuario
3. Snake Case:
    1. Todas as palavras são minúsculas e separadas por underscores. Usada menos frequentemente em JavaScript.
    2. Exemplo: minha_variavel, nome_completo, idade_do_usuario
4. Kebab Case:
    1. Todas as palavras são minúsculas e separadas por hífens. Não é usada para nomear variáveis em JavaScript, mas é comum para nomes de arquivos e URLs.
    2. Exemplo: minha-variavel, nome-completo, idade-do-usuario

Boas Práticas

- Nomes Descritivos:
  - Use nomes descritivos que indiquem claramente o propósito da variável.
  - Exemplo: let numeroDeUsuarios = 5; em vez de let n = 5;
- Use Constantes para Valores Fixos:
  - Use const para declarar variáveis que não mudarão.
  - Exemplo: const PI = 3.14;
- Evite Abreviações Confusas:
  - Evite usar abreviações que não sejam amplamente reconhecidas.
  - Exemplo: let idadeUsuario em vez de let idUsr;

Boas Práticas - Escopo Claro

- Use **let e const** para manter o escopo de suas variáveis claro e evitar problemas de escopo global.

```
if (true) {
  let localVar = 'escopo de bloco';
  console.log(localVar); // 'escopo de bloco'
}
// console.log(localVar); // ReferenceError: localVar is not defined
```

Exemplos Práticos

Nomeação Correta

```javascript
let idade = 25;
let nomeDoUsuario = 'João';
const TAXA_DE_CONVERSAO = 0.85;
```

Nomeação Incorreta

```javascript
let 25anos; // Inválido: começa com número
let nome-usuario; // Inválido: contém hífen
let var; // Inválido: palavra reservada
```

Entrada de Dados com prompt( )

- O prompt é uma função nativa do JavaScript usada para solicitar uma entrada do usuário através de uma janela de diálogo.
- Quando o prompt é chamado, ele exibe uma caixa de diálogo com uma mensagem opcional e um campo de entrada de texto onde o usuário pode digitar um valor.
- O valor inserido pelo usuário é retornado como uma string. Se o usuário clicar em "Cancelar" ou fechar a caixa de diálogo, o prompt retornará null.

Sintaxe

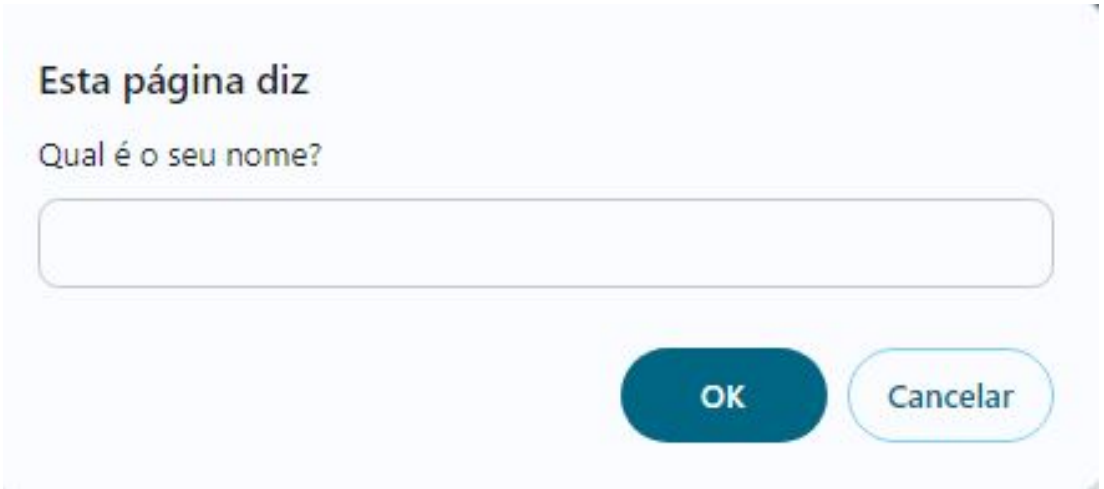
mensagem (opcional): O texto que será exibido na caixa de diálogo.

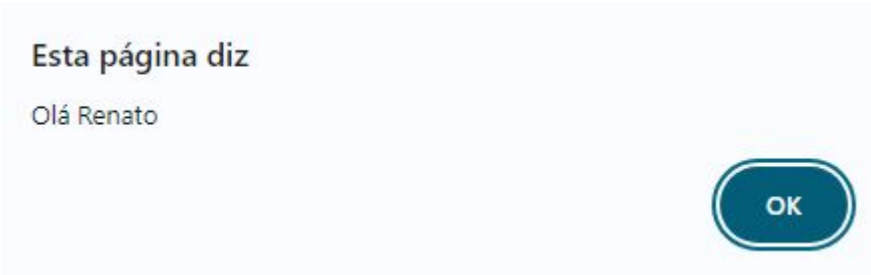
valorPadrão (opcional): Um valor padrão que será exibido no campo de entrada de texto.

```
let valor = prompt(mensagem, valorPadrão);
```

Exemplo

```
1  <meta charset="UTF-8">
2  <script>
3    const nome = prompt("Qual é o seu nome?")
4    alert("Olá " + nome)
5  </script>
6
```





Comentários em JavaScript

Comentários em JavaScript são usados para explicar o código, torná-lo mais legível, e evitar a execução de algumas partes do código durante testes.

Existem dois tipos de comentários em JavaScript: comentários de linha única e comentários de múltiplas linhas.

```
// Este é um comentário de linha única
let x = 5; // Esta é uma variável que recebe o valor 5
```

```
/*
  Este é um comentário de múltiplas linhas.
  Pode ser usado para comentários mais longos
  ou para desativar blocos de código.
*/
let y = 10;

/*
  Este bloco de código está desativado:
  let a = 1;
  let b = 2;
  console.log(a + b);
*/
```

Usos Comuns de Comentários

- Explicação de Código:
  - Comentários ajudam a explicar o que o código faz, especialmente para outras pessoas que possam ler seu código no futuro ou para você mesmo quando revisar o código após algum tempo.

```
// Função para calcular a soma de dois números
function soma(a, b) {
  return a + b;
}
```

Usos Comuns de Comentários

- Desativar Código Temporariamente:
  - Comentários são frequentemente usados para desativar código durante o desenvolvimento e teste.

```
// console.log('Este log está desativado');

/*
console.log('Este bloco de código está desativado');
console.log('Este também');
*/
```

Comentários

```javascript
/**
 * Calcula a área de um círculo dado o raio.
 * @param {number} raio - O raio do círculo.
 * @return {number} A área do círculo.
 */
function calcularAreaCirculo(raio) {
  // Verifica se o raio é um número positivo
  if (raio <= 0) {
    return 0;
  }

  // Constante para Pi
  const PI = 3.14159;

  // Calcula a área usando a fórmula A = πr²
  let area = PI * raio * raio;
  return area;
}

// Exemplo de uso
let raio = 5;
let area = calcularAreaCirculo(raio);
console.log('A área do círculo é:', area); // A área do círculo é: 78.53975
```

Tipos de dados e conversões de tipos

- Em JavaScript, existem vários tipos de dados fundamentais que você pode usar para armazenar e manipular valores.
- A linguagem é dinamicamente tipada, o que significa que as variáveis não têm um tipo fixo e podem armazenar valores de qualquer tipo.
- Além disso, JavaScript oferece várias maneiras de converter tipos de dados de forma explícita e implícita.

Tipos de Dados

- Números
  - Representa tanto inteiros quanto números de ponto flutuante.
  - Exemplos: 42, 3.14, -7.

```
let inteiro = 42;
let pontoFlutuante = 3.14;
```

Tipos de Dados

- Strings
  - Representa texto. Pode ser delimitada por aspas simples, aspas duplas ou crases (para template literals).
  - Exemplos: 'Olá, mundo!', "JavaScript", `Template literal`.

```
let simples = 'Olá';
let duplas = "Mundo";
let template = `Olá, ${duplas}!`;
```

Tipos de Dados

- Booleanos
  - Representa valores de verdade. Pode ser true ou false.
  - Exemplos: true, false.

```
let verdadeiro = true;
let falso = false;
```

Tipos de Dados

- Undefined
  - Representa uma variável que foi declarada, mas não inicializada.
  - Exemplo: let x; // undefined.
- Null
  - Representa a ausência intencional de qualquer valor. Deve ser atribuído explicitamente.
  - Exemplo: let y = null;.

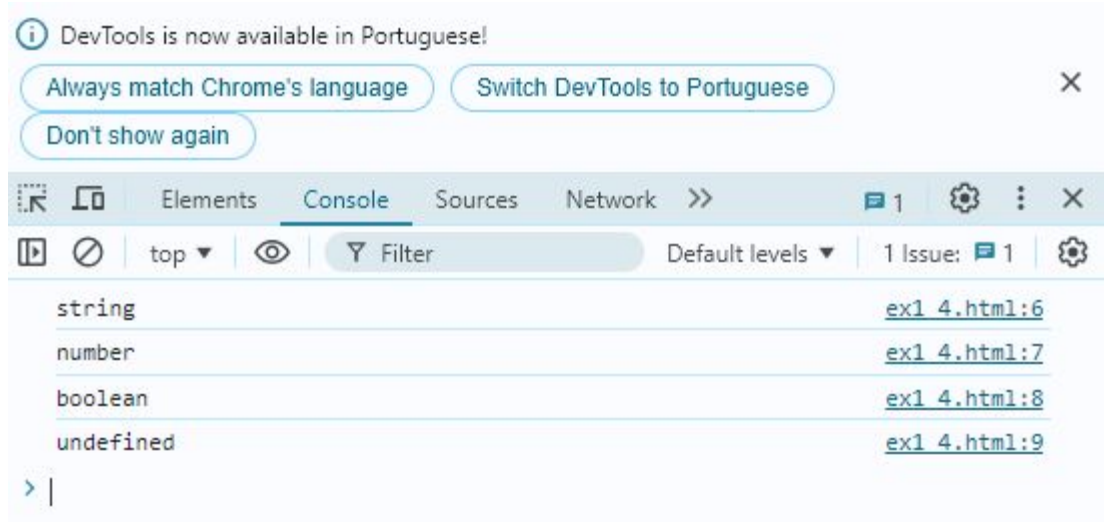
Conversões de Tipo

```
1  <script>
2    /*
3     Operações envolvendo strings e números
4    */
5    const a = "20"
6    const b = a * 2          // b = 40
7    const c = a / 2          // c = 10
8    const d = a - 2          // d = 18
9    const e = a + 2          // e = 202 ???
10   alert("e: " + e)         // exibe o valor de uma variável
11 </script>
```

const e = Number(a) + 2 // =22

Tipos de Variáveis

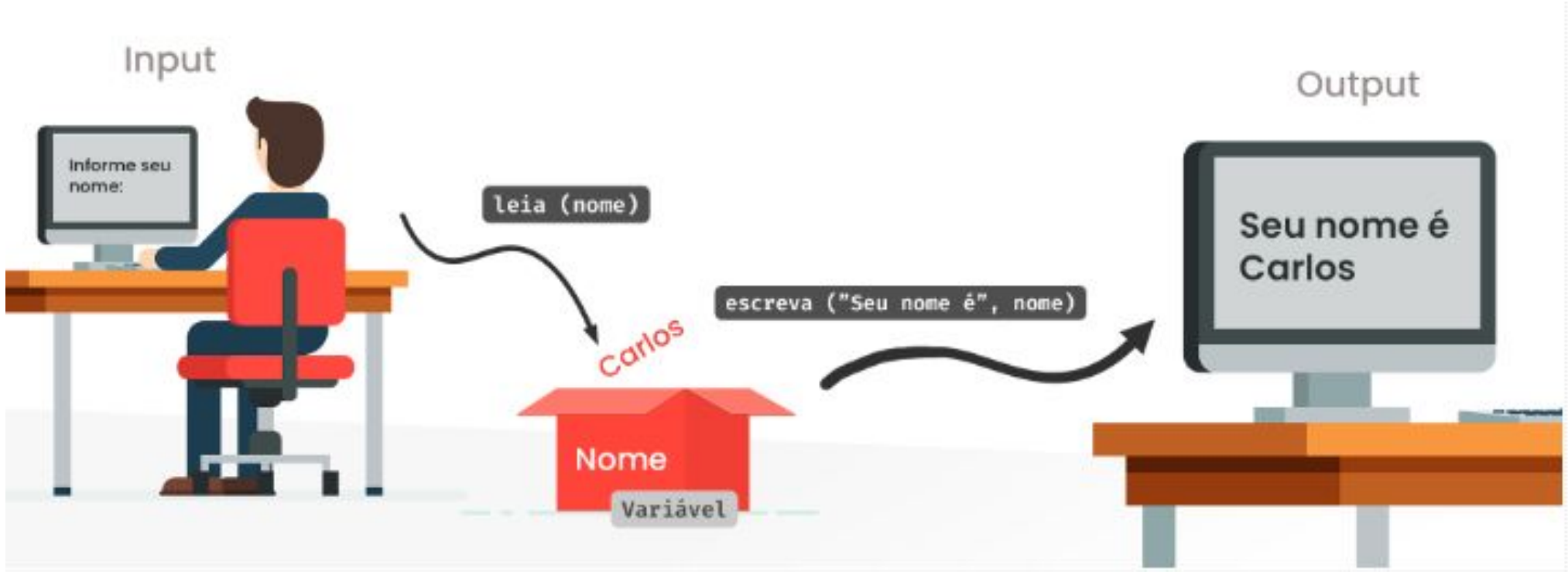
```
1  <script>
2    const fruta = "Banana"
3    const preco = 3.50
4    const levar = true
5    let novoValor
6    console.log(typeof fruta)          // string
7    console.log(typeof preco)          // number
8    console.log(typeof levar)          // boolean
9    console.log(typeof novoValor)      // undefined
10 </script>
11
```



Tipos de Variáveis

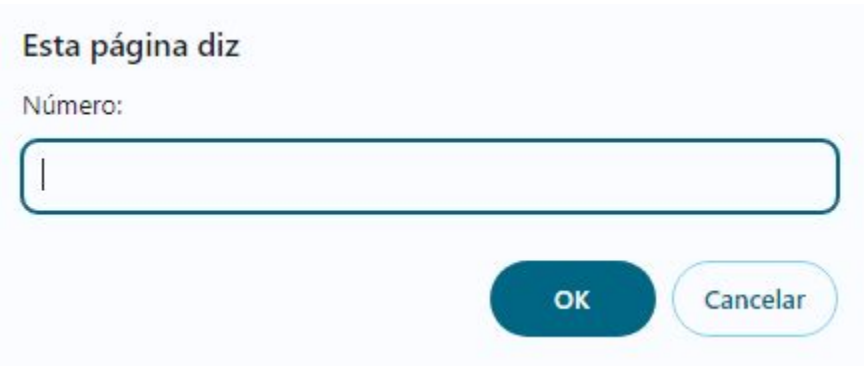
- A declaração de uma variavél/constante com **const** exige uma atribuição de valor - que se materá fixo durante o programa ou bloco de código onde foi declarada.

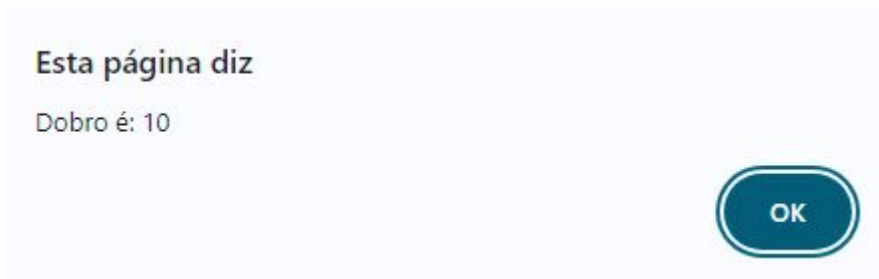
Entrada - Processamento - Saída



Exemplo 1 - Recebe um número e apresenta o dobro

```
<meta charset="utf-8">
<script>
  const num = prompt("Número: ")              // lê um dado de entrada
  const dobro = num * 2                       // calcula o dobro
  alert("Dobro é: " + dobro)                  // exibe a resposta
</script>
```

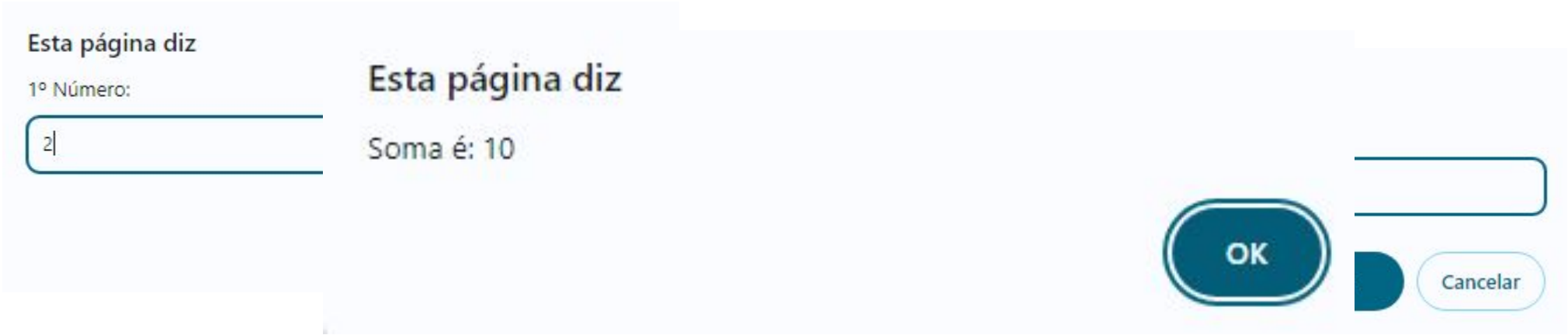




Exemplo 2 - Soma de dois Números

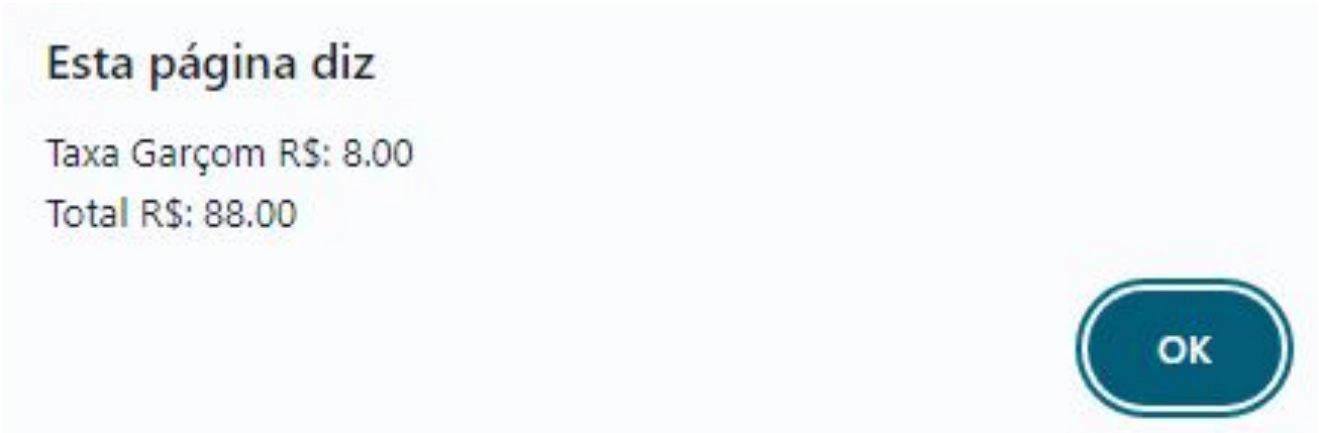
```
<meta charset="utf-8">
<script>
  const num1 = Number(prompt("1° Número: "))        // lê os números
  const num2 = Number(prompt("2° Número: "))
  const soma = num1 + num2                          // calcula a soma
  alert(`Soma é: ${soma}`)                          // exibe o resultado
</script>

```

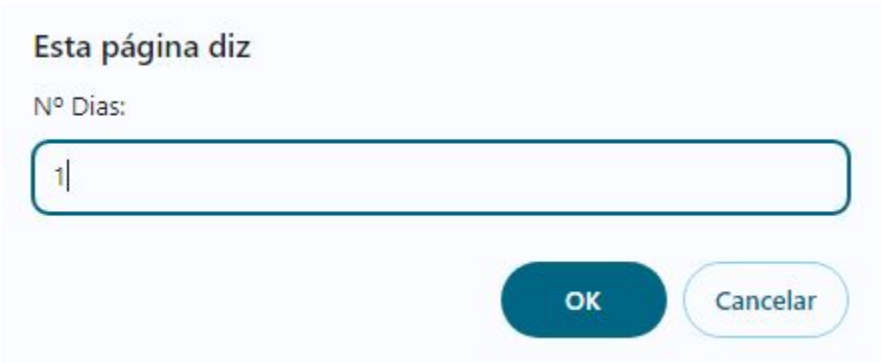


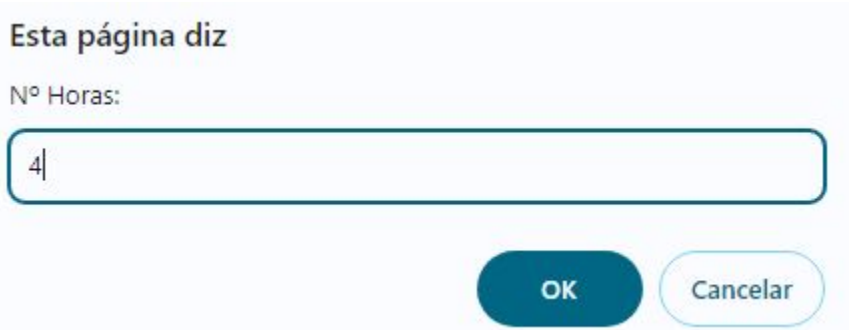
Exemplo 3 - Cálculo do valor do Jantar

```
1  <meta charset="utf-8">
2  <script>
3    const jantar = Number(prompt("Valor do Jantar R$: ")) // lê o valor do jantar
4    const garcom = jantar * 0.10                          // calcula variáveis de saída
5    const total = jantar + garcom
6    alert(`Taxa Garçom R$: ${garcom.toFixed(2)}\nTotal R$: ${total.toFixed(2)}`)
7  </script>
```



Exemplo 4 - Cálculo da duração de horas de uma viagem





Exercícios

1. Elaborar um programa que leia um número. Calcule e informe os seus vizinhos, ou seja, o número anterior e posterior.
2. Elaborar um programa para uma pizzaria, o qual leia o valor total de uma contae quantos clientes vão paga-la. Calcule e informe o valor a ser pago por cliente.
3. Elaborar um programa para uma loja, o qual leia o preço de um produto e informe as opções de pagamento da loja. Calcule e informe o valor para pagamento à vista com 10% de desconto e o valor em 3x.
4. Elaborar um programa que leia 2 notas de um aluno em uma disciplina. Calcule e informe a média das notas.