Professores:

Celso Giusti

Daniel Manoel Filho

Isadora Dias Afonso

Marlon Palata Fanger Rodrigues

INTRODUÇÃO AO JAVASCRIPT EM FRONT-END

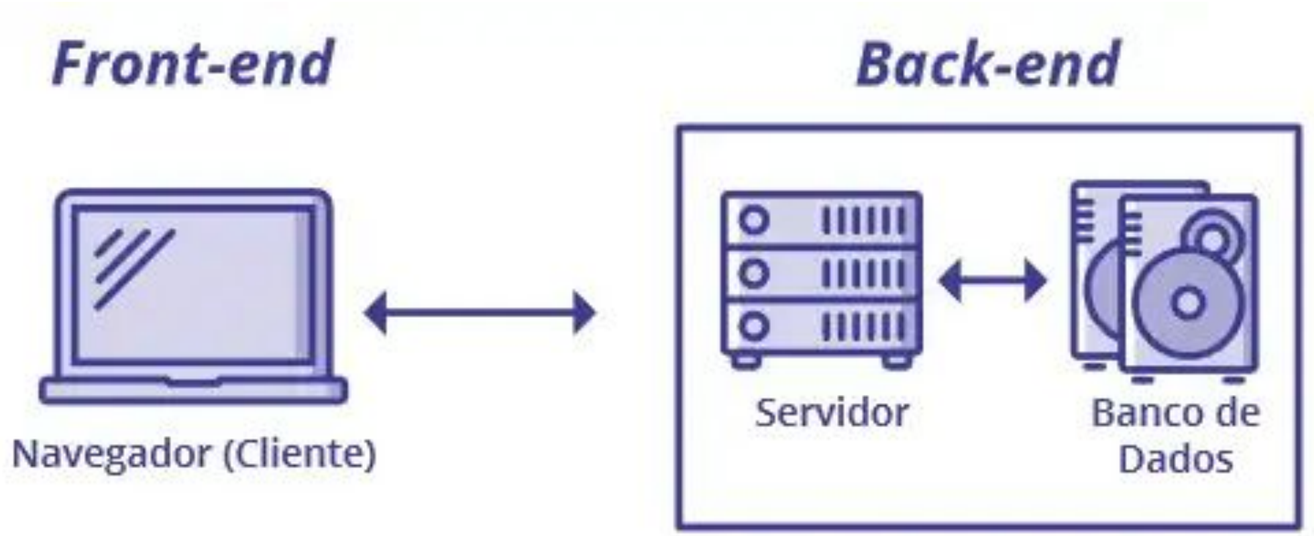
SENAI

Javascript em Front-End

Do Back-end para o Front-end

Como vocês já viram JavaScript no semestre passado voltado para o Back-end, a lógica (variáveis, loops, condicionais) continua a mesma. A grande diferença agora é **ONDE** o código roda.

- **Back-end:** O código roda no servidor (Node.js) para processar dados e banco de dados.
- **Front-end:** O código roda no **Navegador** do usuário para manipular a tela e reagir a cliques.



Javascript em Front-End

Onde inserir o código?

Tag <script>

No Front-end, o navegador precisa saber que aquele arquivo é um script. Usamos a tag **<script>**. Existem duas formas:

1. **Interna:** Código escrito direto no HTML (usado para testes rápidos).
2. **Externa:** Código em um arquivo .js separado (padrão profissional).

Javascript em Front-End

Uso Externo vs Interno

Característica	Script Interno	Script Externo (Recomendado)
Organização	Polui o HTML.	Mantém o projeto limpo.
Reutilização	Só funciona naquela página.	Pode ser usado em várias páginas.
Performance	Carrega sempre.	O navegador faz cache (fica mais rápido).
Manutenção	Difícil de achar o erro.	Fácil de organizar pastas.

Javascript em Front-End

O Método Padrão

Para a maioria dos casos, o ideal é colocar a tag **<script>** logo **antes de fechar a tag </body>**. Isso garante que o navegador carregue todo o conteúdo visual (HTML/CSS) antes de tentar executar o script.

```
<!doctype html>
<html lang="pt-br">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
 </head>
 <body>
  <h1>Olá, Mundo!</h1>
    <script src="src/scripts/script.js"></script>
 </body>
</html>
```

Javascript em Front-End

Usando o Atributo defer

Se você prefere manter o link dentro do **<head>**, use o atributo **defer**. Ele diz ao navegador para baixar o script em segundo plano e só executá-lo depois que o HTML estiver totalmente carregado.

```
<head>
    <script src="src/scripts/script.js" defer></script>
</head>
```

Javascript em Front-End

Operadores Aritméticos

São usados para cálculos matemáticos. No Front-end, usamos muito para calcular totais de carrinhos, fretes ou porcentagens de barras de progresso.

Operador	Operação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto (Módulo)

Javascript em Front-End

Operadores Relacionais

Servem para comparar valores. O resultado é sempre Booleano (**true** ou **false**).
Essencial para validações (ex: "O usuário preencheu o campo?").

Operador	Significado
==	Igual (valor)
===	Estritamente igual (valor e tipo)
!=	Diferente
> / <	Maior / Menor
>= / <=	Maior ou igual / Menor ou igual

Javascript em Front-End

Operadores Lógicos

Usados para criar condições complexas.

- **&& (E):** Todas as partes devem ser verdadeiras.
- **|| (OU):** Apenas uma parte precisa ser verdadeira.
- **! (NÃO):** Inverte o estado (o que é true vira false).

Javascript em Front-End

Funções de Data e Hora

No Front-end, o objeto **Date()** é usado para mostrar horários, verificar se o restaurante está aberto ou exibir prazos de entrega em tempo real.

O JavaScript utiliza o objeto **Date** para trabalhar com tempo. No Front-end, ele captura as informações do relógio interno do dispositivo do usuário (celular ou computador).

Principais comandos para "extrair" informações:

- **getHours()**: Retorna a hora atual (0 a 23).
- **getMinutes()**: Retorna os minutos (0 a 59).
- **getDate()**: Retorna o dia do mês (1 a 31).
- **getMonth()**: Retorna o mês (**Atenção:** Janeiro é 0, Fevereiro é 1...).
- **getFullYear()**: Retorna o ano com 4 dígitos (ex: 2026).
- **toLocaleDateString()**: Formata a data inteira ((pt-br) padrão brasileiro → DD/MM/AAAA).
- **toLocaleTimeString():** Formata a hora completa ((pt-br) padrão brasileiro → hh:mm:ss - Horas de 0 a 24).

DOC: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Date

Javascript em Front-End

Funções de Data e Hora

```
// Criando o objeto com a data/hora exata do agora
const tempoAgora = new Date();

// 1. Pegando a hora para saudação
const hora = tempoAgora.getHours();

// 2. Formatando data e hora para o recibo do cliente
const dataFormatada = tempoAgora.toLocaleDateString('pt-BR');
const horaFormatada = tempoAgora.toLocaleTimeString('pt-BR');

console.log("Pedido realizado em: " + dataFormatada);
console.log("Horário: " + horaFormatada);

if (hora >= 18 && hora <= 23) {
    alert("Estamos abertos! Boa janta.");
}
```

Javascript em Front-End

Funções Matemáticas: Objeto Math

Diferente de cálculos simples, o **Math** resolve arredondamentos de preços ou gera números aleatórios (ex: sorteio de brindes).

Exemplo:

```
let precoQuebrado = 45.89;
console.log(Math.round(precoQuebrado)); // 46 (Arredonda para o mais próximo)
console.log(Math.floor(precoQuebrado)); // 45 (Arredonda para baixo)
console.log(Math.random()); // Gera número entre 0 e 1
```

Javascript em Front-End

Funções de String

Trabalhar com textos no Front-end é constante: formatar nomes de clientes, validar e-mails ou esconder partes de um cartão de crédito.

Exemplo:

```
let nome = " TechFood ";
console.log(nome.trim()); // Remove espaços vazios nas pontas
console.log(nome.toUpperCase()); // TECHFOOD
console.log(nome.length); // Conta os caracteres
```

Javascript em Front-End

PREPARAÇÃO EXEMPLOS TECHFOOD

Nesta aula, o foco é a lógica pura e a interação inicial. O código ainda não "mexe" no visual do HTML de forma automática, ele usa as janelas do navegador (**alert/prompt**).

1. Preparação do Ambiente

- Crie uma pasta chamada **src/scripts/.**
- Crie o arquivo **aula4.js** dentro dela.
- No final do seu **index.html,** antes de fechar a tag **</body>**, adicione: **<script src="src/scripts/aula4.js"></script>**

Javascript em Front-End

Orientação a Objetos: Classes e Atributos

A OO no Front-end ajuda a organizar os dados que vêm do banco de dados para serem mostrados na tela. A **Classe** é o molde (ex: Produto) e os **Atributos** são os dados (ex: nome, preço).

Exemplo:

```
class Prato {
    constructor(nome, preco) {
        this.nome = nome;
        this.preco = preco;
    }
}
```

Javascript em Front-End

OO: Métodos/Funções Internas

Métodos são comportamentos do objeto. Por exemplo, um produto necessitar de uma adaptação de exibição como moeda.

Exemplo:

```
class Prato{
   constructor(nome, preco) { ... }

   exibirComMoeda() {
      return "R$ " + this.preco.toFixed(2);
   }
}
```

Javascript em Front-End

Comando: alert()

O comando **alert()** é uma função nativa do navegador que exibe uma caixa de mensagem simples. No Front-end, ele é muito usado para avisos urgentes ou confirmações de segurança. Enquanto o alerta estiver na tela, a navegação do usuário fica pausada.

Exemplo: Vamos criar um aviso automático que aparece assim que o cliente entra no portal da loja.

```
// Exibe uma mensagem de boas-vindas ao carregar o TechFood
alert("Seja bem-vindo ao TechFood! Aproveite nossos cupons de hoje.");
```

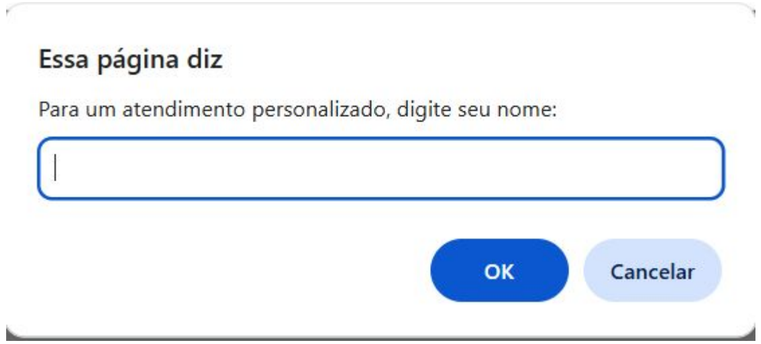
Javascript em Front-End

Comando: prompt()

Diferente do alerta que só mostra dados, o **prompt()** solicita um dado ao usuário. Ele abre uma caixa de texto e o que o usuário digitar pode ser guardado em uma variável.

Exemplo: Vamos capturar o nome do cliente para o atendimento no início da navegação.

```
// Capturando o nome para personalizar a interface
const cliente = prompt("Para um atendimento personalizado, digite seu nome:");
```



Nota: Todo dado vindo do prompt chega ao JavaScript como uma String (texto).

Javascript em Front-End

Funções de String (Exemplo)

Vamos "limpar" o nome que o usuário digitou, removendo espaços acidentais e padronizando para letras maiúsculas.

```
let cliente = "  joao silva  ";
let nomeFormatado = cliente.trim().toUpperCase();


alert("BEM-VINDO, " + nomeFormatado); // Saída: "BEM-VINDO, JOAO SILVA"
```

Javascript em Front-End

Objeto Date() (Exemplo)

Vamos verificar a hora atual do computador do cliente para decidir se exibimos uma saudação de "Bom dia" ou "Boa tarde".

```
const tempo = new Date();
    const hora = tempo.getHours();

    if (hora < 12) {
      alert("Bom dia! O TechFood já está aceitando pedidos.");
    } else {
      alert("Olá! Confira nosso cardápio de almoço e jantar.");
    }
```

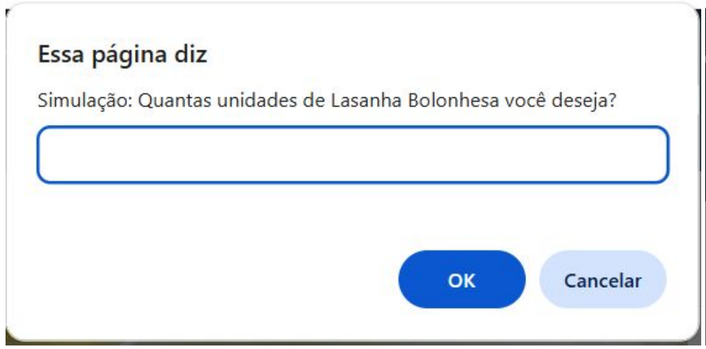
Javascript em Front-End

Operadores Aritméticos (Exemplo)

Vamos calcular o valor total de uma compra multiplicando o preço fixo de um prato pela quantidade informada pelo cliente.

```
const lasanha = new Prato("Lasanha Bolonhesa", 45.00);

let qtd = prompt("Simulação: Quantas unidades de " + lasanha.nome + " você deseja?");
let total = lasanha.preco * qtd;
// Exibindo o cálculo final
alert("Resumo da Simulação:\nPrato: " + lasanha.nome + "\nTotal: " +
lasanha.exibirComMoeda());
```



Javascript em Front-End

Resumo

SENAI

✅ JavaScript no navegador manipula a interface.

✅ Script Externo é a melhor prática para organização.

✅ Operadores processam dados vindos do usuário.

✅ Classes organizam os componentes do site de forma profissional.

✅ **alert** e **prompt** são as primeiras ferramentas de interação.

✅ O objeto **Date** lê o relógio do cliente.

Referências

SENAI-SP. **Desenvolvimento de Sistemas: Front-end**. São Paulo: Editora SENAI, 2023. MDN WEB DOCS.. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/.

W3C. **JAVASCRIPT**. Disponível em: https://www.w3.org/.

MARCOTTE, Ethan. **Responsive Web Design**. A Book Apart, 2011.

FOTO DA ESCOLA

ESCOLA SENAI ÍTALO BOLOGNA

Av. Goiás, 139

Telefone

(11) 2396-1999

Instagram

@senai.itu

Facebok

/senaisp.itu

Site

https://sp.senai.br/unidade/itu/