

Técnico em Informática para Internet

Unidade Curricular 3: Codificar front-end de aplicações web

Olá! Sejam bem vindos



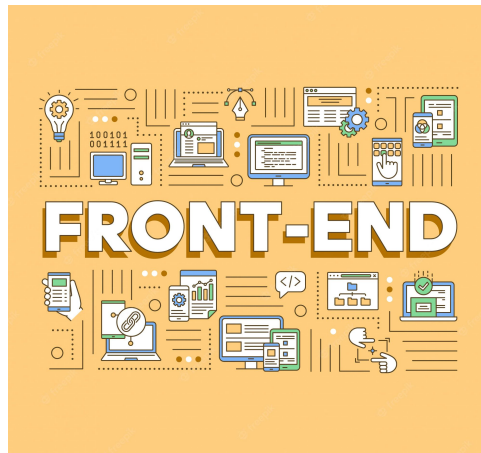
DISCIPLINA/UNIDADE CURRICULAR	UC03 - Codificar front-end de aplicações web 20h
TEMA DA AULA	JavaScript para manipular elementos estruturais (DOM)
INDICADOR	2. Manipula os elementos estruturais, de acordo com os requisitos do projeto
CRONOGRAMA	Document Object Model (DOM): objetos, propriedades e eventos; manipulação de elementos, atribuição de eventos e estilos dinâmicos
ATIVIDADE	Descrição de um exercício prático que os alunos podem fazer para aplicar o que aprenderam

Objetivos da Aula

- Estudaremos a manipulação de elementos do DOM (Document Object Model), que é a estrutura do HTML de uma página web.
- Essa manipulação será feita usando JavaScript e é uma habilidade crucial para a codificação de front-end.

Programação FrontEnd

- O Front-end está muito relacionado com a interface gráfica do projeto.
- É onde se desenvolve a aplicação com a qual o usuário irá interagir diretamente, seja em softwares, sites, aplicativos, etc.
- Portanto, é essencial que o desenvolvedor tenha uma preocupação com a experiência do usuário.



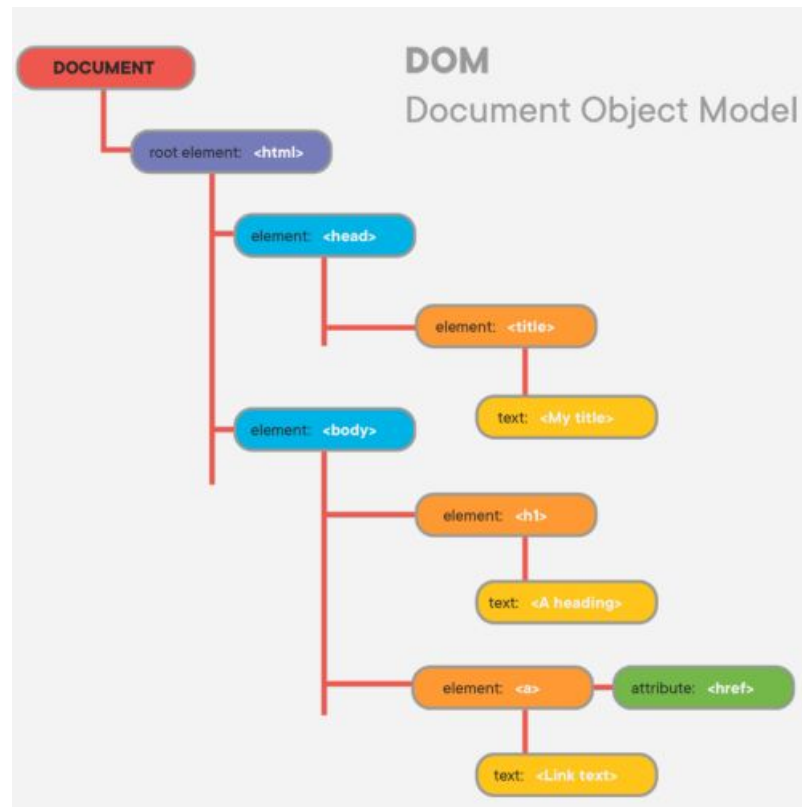
Introdução ao DOM

Introdução ao DOM

- O DOM (Document Object Model) é uma interface que permite a programadores JavaScript interagir e manipular páginas web.
- O DOM transforma toda a página web em um objeto com propriedades e métodos que podem ser usados para manipular os elementos da página.

Introdução ao DOM

- O DOM foi criado pela W3C com o objetivo de desenvolver um padrão para linguagens de script para os navegadores.



Introdução ao DOM

- O DOM é como uma árvore que representa a estrutura de uma página web.
- Cada parte da página (cada tag HTML) é um nó desta árvore. Por exemplo, considere a seguinte estrutura HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Meu site</title>
  </head>
  <body>
    <h1>Olá, mundo!</h1>
    <p>Esta é minha primeira página web.</p>
  </body>
</html>
```

A tag `<html>` é o nó raiz.
Ela tem dois nós filhos: `<head>` e `<body>`.
O nó `<head>` tem um nó filho: `<title>`,
enquanto o nó `<body>` tem dois nós filhos:
`<h1>` e `<p>`. Cada um desses nós é um objeto
que pode ser manipulado com JavaScript.

Introdução ao DOM

- O DOM permite que você acesse e modifique esses nós usando JavaScript, o que significa que você pode alterar o conteúdo, estrutura e estilo de uma página web dinamicamente.
- Por exemplo, você poderia usar JavaScript para alterar o texto dentro do elemento `<h1>` de 'Olá, mundo!' para 'Bem-vindo ao meu site!'. Ou você poderia usar JavaScript para adicionar uma nova tag `<p>` ao `<body>` da página.
- A capacidade de manipular o DOM com JavaScript é uma habilidade crucial para a codificação de front-end e é a base de muitas interações dinâmicas na web.

Identifique os NÓS do código HTML abaixo:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8">
    <title>Exemplo 01</title>
  </head>
  <body>
    <h1> Programa Olá Você! </h1>
    <p>
      Nome: <input type="text" id="nome">
      <input type="button" value="Mostrar">
    </p>
    <p id="resposta"></p>
  </body>
</html>
```

Seleção de Elementos no DOM

Seleção de Elementos DOM

- Para manipular um elemento do DOM, você precisa primeiro selecioná-lo.
- Existem vários métodos para fazer isso, como **getElementById**, **getElementsByClassName**, **getElementsByTagName** e **querySelector**.

Seleção de Elementos DOM

- **getElementById:**

- Este método retorna o elemento que tem o atributo ID com o valor especificado. Como os IDs são únicos, este método é muito útil quando você precisa encontrar um único elemento.

```
let titulo = document.getElementById('meuTitulo');
```

Seleção de Elementos DOM

- **getElementsByClassName:**
 - Este método retorna uma lista de todos os elementos com a classe especificada.

```
let itens = document.getElementsByClassName('minhaClasse');
```

Seleção de Elementos DOM

- **getElementsByTagName:**
 - Este método retorna uma lista de todos os elementos com a tag especificada.

```
let paragrafos = document.getElementsByTagName('p');
```

Seleção de Elementos DOM

- **querySelector:**

- Este método retorna o primeiro elemento que combina com o seletor CSS especificado.

```
let primeiroItem = document.querySelector('.minhaClasse');
```


Seleção de Elementos DOM

- **querySelectorAll:**
 - Este método retorna todos os elementos que combinam com o seletor CSS especificado.

```
let todosItens = document.querySelectorAll('.minhaClasse');
```

Seleção de Elementos DOM

- Lembrando que os métodos **getElementsByClassName**, **getElementsByTagName** e **querySelectorAll** retornam uma lista de elementos.
- Para acessar um elemento específico dessa lista, você precisa usar um índice, assim como faria com um array.

```
let primeiroParagrafo = document.getElementsByTagName('p')[0];
```

Atividade 1 - Selecionando e Manipulando Elementos do DOM

- Crie um arquivo HTML com a seguinte estrutura e Adicione um script JavaScript no final do seu arquivo HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exercício DOM</title>
  </head>
  <body>
    <h1 id="titulo">Olá, mundo!</h1>
    <p class="texto">Este é o primeiro parágrafo.</p>
    <p class="texto">Este é o segundo parágrafo.</p>
    <div id="novaDiv"></div>
  </body>
</html>
```

Atividade 1 - Selecionando e Manipulando Elementos do DOM

- a. Use `getElementById` para selecionar o elemento `<h1>` e altere o texto para "Bem-vindo ao meu site!".
- b. Use `getElementsByClassName` para selecionar todos os elementos com a classe 'texto'. Atribua a cor "blue" para o primeiro parágrafo e a cor "green" para o segundo parágrafo.
- c. Use `querySelector` para selecionar o primeiro parágrafo e mude o texto para "Este é um novo texto".
- d. Use `querySelectorAll` para selecionar todos os parágrafos e imprima o número de parágrafos na console.

Dica:

- Para alterar o texto de um elemento, você pode usar a propriedade `innerText`
- Para alterar a cor do texto, você pode usar a propriedade `style.color`

Atividade 2 - Seleccionando e Manipulando Elementos do DOM

a. Crie um novo elemento `<p>` com o texto "Este é um novo parágrafo." e adicione-o ao elemento `<div>` usando `getElementById` e o método `appendChild`.

Dica:

- Para criar um novo elemento, você pode usar o método `document.createElement`.

Atividade - Solução

```
// a. Selecionando o elemento <h1> e alterando o texto
let titulo = document.getElementById('titulo');
titulo.innerText = "Bem-vindo ao meu site!";

// b. Selecionando os elementos com a classe 'texto' e alterando as cores
let textos = document.getElementsByClassName('texto');
textos[0].style.color = "blue";
textos[1].style.color = "green";

// c. Selecionando o primeiro parágrafo e alterando o texto
let primeiroTexto = document.querySelector('.texto');
primeiroTexto.innerText = "Este é um novo texto";

// d. Selecionando todos os parágrafos e imprimindo o número de parágrafos
let todosParagrafos = document.querySelectorAll('p');
console.log(`Existem ${todosParagrafos.length} parágrafos na página.`);
```

```
// Criando um novo elemento <p>
let novoParagrafo = document.createElement('p');
novoParagrafo.innerText = "Este é um novo parágrafo.";

// Adicionando o novo parágrafo ao elemento <div>
let novaDiv = document.getElementById('novaDiv');
novaDiv.appendChild(novoParagrafo);
```

```
<script src="arquivo.js"></script>
```

Modificando Elementos do DOM

Modificando Elementos do DOM

- Uma vez que você selecionou um elemento DOM, é possível modificar suas propriedades e comportamento.
- Você pode alterar o conteúdo de texto do elemento, alterar atributos do elemento, como classe ou ID, ou até mesmo adicionar ou remover elementos.

Modificando Elementos do DOM

- Propriedades `textContent`, `innerHTML` e `value`

<code>textContent</code>	Consulta ou altera o texto exibido por elementos HTML como parágrafos (p), cabeçalhos (h1 , h2 ,...) ou containers (span , div).
<code>innerHTML</code>	Consulta ou altera o conteúdo de elementos HTML como parágrafos (p), cabeçalhos (h1 , h2 ,...) ou containers (span , div). Códigos HTML presentes no conteúdo são renderizados pelo navegador.
<code>value</code>	Consulta ou altera o conteúdo de campos de formulário.

Modificando Elementos do DOM

- **Alterando conteúdo de texto:** A propriedade `innerText` permite ler e modificar o conteúdo de texto de um elemento.

```
let titulo = document.getElementById('meuTitulo');  
titulo.innerText = 'Novo título para a página';
```

Modificando Elementos do DOM

- **Alterando HTML:** A propriedade `innerHTML` permite ler e modificar o HTML dentro de um elemento. Use-a com cuidado, pois ela pode fazer com que o seu site fique vulnerável a ataques de injeção de script.

```
let div = document.getElementById('minhaDiv');  
div.innerHTML = '<p>Este é um novo parágrafo.</p>';
```

Modificando Elementos do DOM

- **Alterando atributos:** O método `setAttribute` permite modificar o valor de um atributo de um elemento. O método `getAttribute` permite ler o valor de um atributo.

```
let link = document.getElementById('meuLink');  
link.setAttribute('href', 'https://www.novosite.com');  
console.log(link.getAttribute('href')); // Imprime 'https://www.novosite.com'
```

Modificando Elementos do DOM

- **Adicionando e removendo classes:** Os métodos `classList.add`, `classList.remove` e `classList.toggle` permitem adicionar, remover e alternar classes, respectivamente.

```
let paragrafo = document.getElementById('meuParagrafo');  
paragrafo.classList.add('destaque'); // Adiciona a classe 'destaque'  
paragrafo.classList.remove('destaque'); // Remove a classe 'destaque'  
paragrafo.classList.toggle('destaque'); // Alterna a classe 'destaque'
```

Modificando Elementos do DOM

- **Adicionando e removendo elementos:** Os métodos `appendChild`, `insertBefore` e `removeChild` permitem adicionar e remover elementos.

```
let novaDiv = document.createElement('div'); // Cria um novo elemento 'div'
document.body.appendChild(novaDiv); // Adiciona o novo 'div' ao final do 'body'

let novoParagrafo = document.createElement('p'); // Cria um novo elemento 'p'
document.body.insertBefore(novoParagrafo, novaDiv); // Insere o 'p' antes da nova 'div'

document.body.removeChild(novaDiv); // Remove a nova 'div'
```

Atividade 01 - Modificando Elementos do DOM

- Crie um arquivo HTML com a seguinte estrutura e adicione um script JavaScript no final do seu arquivo HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exercício DOM</title>
    <style>
      .destaque {
        color: red;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <h1 id="titulo">Olá, mundo!</h1>
    <a id="meuLink" href="https://www.google.com">Ir para Google</a>
    <p id="meuParagrafo">Este é um parágrafo.</p>
    <div id="minhaDiv"></div>
  </body>
</html>
```

Atividade 01 - Modificando Elementos do DOM

- No seu script, faça o seguinte:

- a. Use `innerText` para alterar o texto do elemento `<h1>` para "Bem-vindo ao meu site!".
- b. Use `innerHTML` para adicionar um novo parágrafo dentro da `<div>` com o texto "Este é um novo parágrafo."
- c. Use `setAttribute` para mudar o endereço do link para "https://www.senac.br/".
- d. Use `classList.add` para adicionar a classe 'destaque' ao parágrafo.
- e. Crie um novo elemento `<p>` com o texto "Este é outro parágrafo." e adicione-o ao final do `<body>` usando `appendChild`.

Dica:

- para criar um novo elemento, você pode usar o método `document.createElement`.
- Para acessar o `<body>`, você pode usar `document.body`.

Atividade 01 - Solução

```
// a. Alterando o texto do elemento <h1>
document.getElementById('titulo').innerText = "Bem-vindo ao meu site!";

// b. Adicionando um novo parágrafo dentro da <div>
document.getElementById('minhaDiv').innerHTML = "<p>Este é um novo parágrafo.</p>";

// c. Mudando o endereço do link
document.getElementById('meuLink').setAttribute('href', 'https://www.senac.br');

// d. Adicionando a classe 'destaque' ao parágrafo
document.getElementById('meuParagrafo').classList.add('destaque');

// e. Criando um novo elemento <p> e adicionando ao final do <body>
let novoParagrafo = document.createElement('p');
novoParagrafo.innerText = "Este é outro parágrafo.";
document.body.appendChild(novoParagrafo);
```

Eventos no DOM

Eventos no DOM

- Os eventos são ações ou ocorrências que acontecem no sistema que você está programando — o sistema pode ser sua página da web, o navegador, o computador ou qualquer outro sistema no qual o seu código está sendo executado.
- Por exemplo, se o usuário clica em um botão na sua página, isso é um evento. Quando a página termina de carregar, isso também é um evento.

Eventos no DOM

- Os eventos são uma parte fundamental da programação interativa no navegador.
- Eles permitem que você crie páginas que respondem às ações do usuário, como cliques, movimentos do mouse, pressionamento de teclas, e muito mais.
- Dominar o uso de eventos é um passo importante para se tornar um bom desenvolvedor front-end.

Como trabalhar com eventos

- Para responder a um evento, você precisa:
 - a. Identificar o elemento DOM que será observado. Por exemplo, você pode querer responder a um clique em um botão.
 - b. Registrar um manipulador de eventos para esse elemento. Um manipulador de eventos é uma função que é chamada quando o evento ocorre.
- Exemplos de eventos:
 - a. click: Ocorre quando o usuário clica em um elemento.
 - b. load: Disparado quando a página termina de carregar.
 - c. mousemove: Ocorre quando o cursor do mouse é movido sobre um elemento.

Atividade 01 - Eventos no DOM

- Cenário: Vamos supor que você está criando uma aplicação web simples. Você quer alterar a cor de fundo de um parágrafo quando o usuário passa o mouse sobre ele e mudar de volta quando o mouse sai.
- Instruções:
 - Crie uma página HTML básica com um parágrafo de texto.
 - Use JavaScript para adicionar ouvintes de eventos ao parágrafo para os eventos mouseover e mouseout.
 - Quando o mouse passa sobre o parágrafo (mouseover), mude a cor de fundo do parágrafo para amarelo.
 - Quando o mouse sai do parágrafo (mouseout), mude a cor de fundo de volta ao original.
- Nota: Você pode utilizar o método `addEventListener` para adicionar ouvintes de eventos e a propriedade `style.backgroundColor` para alterar a cor de fundo.

Atividade 01 - Solução

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>

<p id="myParagraph">Passe o mouse sobre mim</p>

<script src="script.js"></script>

</body>
</html>
```

```
#myParagraph {
  transition: background-color 0.5s;
}

#myParagraph:hover {
  background-color: yellow;
}
```

```
// selecionamos o parágrafo pelo seu id
let paragraph = document.getElementById("myParagraph");

// adicionamos um ouvinte de eventos para quando o mouse passa sobre o parágrafo
paragraph.addEventListener("mouseover", function() {
  paragraph.classList.add("hover");
});

// adicionamos um ouvinte de eventos para quando o mouse sai do parágrafo
paragraph.addEventListener("mouseout", function() {
  paragraph.classList.remove("hover");
});
```

Funções no JavaScript

- Funções JavaScript são declaradas a partir da palavra-chave function seguida do nome da função e dos parênteses ().
- Uma função contém um conjunto de comandos que realizam uma ação.

```
<input type="button" value="Mostrar" onclick="mostrarOla()">
```

```
<script>
  function mostrarOla() {
    var inputNome = document.getElementById("nome");
    var nome = inputNome.value;
    document.getElementById("resposta").textContent = "Olá " + nome;
  }
</script>
```

- Objeto – representa uma instância de uma classe.
- Método – representa uma instrução ou um conjunto de instruções que executam uma tarefa.
- Propriedade – representa uma característica (atributo) de um objeto

Funções no JavaScript

- Alterar a tab button.

```
<input type="button" value="Mostrar" id="mostrar">
```

- Arquivo .js

```
function mostrarOla() {  
    var nome = document.getElementById("nome").value;  
    document.getElementById("resposta").innerHTML = "Olá " + nome;  
}  
var mostrar = document.getElementById("mostrar");  
mostrar.onclick = mostrarOla;
```

Exercícios Práticos

Exercício 01

- Elaborar um programa para uma Vídeo Locadora, que leia o título e a duração de um filme em minutos. Exiba o título do filme e converta a duração para horas e minutos.



The screenshot shows a web browser window with the title "Exemplo 2.2". The address bar displays the file path "file:///C:/livrojs/cap2/ex2_2.html". The main content area has the heading "Vídeo Locadora JS". Below the heading, there are two input fields: "Título do Filme:" with the value "O Touro Ferdinando" and "Duração (min):" with the value "108". A button labeled "Converter em Horas e Minutos" is positioned below these fields. The output of the conversion is displayed below the button, showing "O Touro Ferdinando" and "1 hora(s) e 48 minuto(s)".

Exercício 02

- Elaborar um programa para uma revenda de veículos. O programa deve ler modelo e preço do veículo. Apresentar como resposta o valor da entrada (50%) e o saldo em 12x.



The screenshot shows a web browser window with the title 'Exemplo 2.3'. The address bar displays 'file:///C:/livrojs/cap2/ex2_3.html'. The main content area features a form titled 'Revenda de Veículos JS'. The form has two input fields: 'Veículo:' with the value 'Sandero 2016' and 'Preço RS:' with the value '30000.00'. Below these fields is a button labeled 'Ver Promoção'. The results of the calculation are displayed below the button: 'Promoção: Sandero 2016', 'Entrada de RS: 15000.00', and '+ 12x de RS: 1250.00'.

Revenda de Veículos JS

Veículo:

Preço RS:

Promoção: Sandero 2016

Entrada de RS: 15000.00

+ 12x de RS: 1250.00

Exercício 03

- Elaborar um programa para um restaurante que leia o preço por kg e o consumo (em gramas) de um cliente. Exiba o valor a ser pago, cfme exemplo:



The screenshot shows a web browser window titled "Exemplo 2.4" with the address bar displaying "file:///C:/livrojs/cap2/ex2_4.html". The page content is as follows:

Restaurante JS

Buffet por Quilo R\$:

Consumo do Cliente (gr):

Valor a pagar R\$: 28.00

Exercício 04

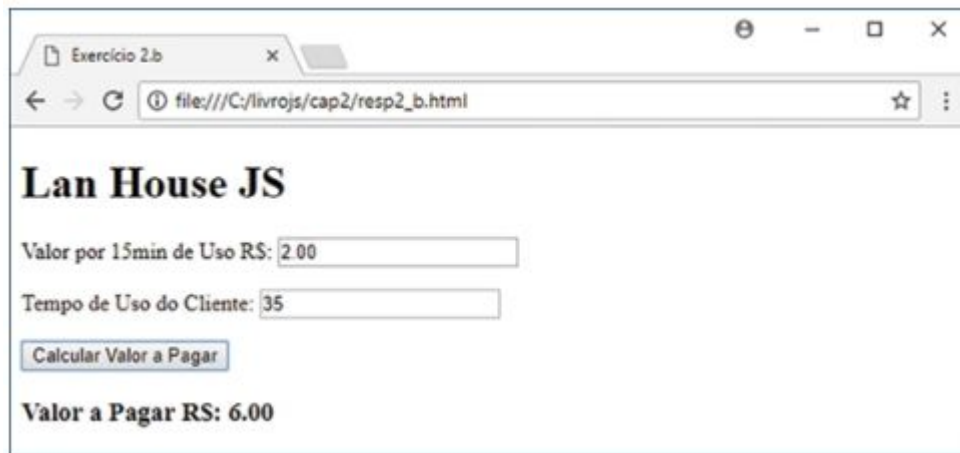
- Uma farmácia está com uma promoção – Na compra de duas unidades de um mesmo medicamento, o cliente recebe como desconto os centavos do valor total. Elaborar um programa que leia descrição e preço de um medicamento. Informe o valor do produto na promoção. A Figura apresenta a tela com um exemplo de dados de entrada e saída do programa.



The screenshot shows a web browser window titled "Exercício 2.a" with a single tab. The address bar displays the file path "file:///C:/livrojs/cap2/resp2_a.html". The main content area features the heading "Farmácia JS". Below this, there are two input fields: "Medicamento:" containing the text "Aspirina" and "Preço RS:" containing the value "7.30". A button labeled "Mostrar Promoção" is positioned below the inputs. The output section, titled "Promoção de Aspirina", displays the text "Leve 2 por apenas RS: 14.00".

Exercício 05

- Elaborar um programa para uma lan house de um aeroporto – O programa deve ler o valor de cada 15 minutos de uso de um computador e o tempo de uso por um cliente em minutos. Informe o valor a ser pago pelo cliente, sabendo que as frações extras de 15 minutos devem ser cobradas de forma integral. A Figura exibe um exemplo com dados do programa.



The screenshot shows a web browser window with a single tab titled "Exercicio 2.b". The address bar displays the file path "file:///C:/livrojs/cap2/resp2_b.html". The main content area has the title "Lan House JS" in bold. Below the title, there are two input fields: "Valor por 15min de Uso RS:" with the value "2.00" and "Tempo de Uso do Cliente:" with the value "35". A button labeled "Calcular Valor a Pagar" is positioned below these fields. At the bottom, the result is displayed as "Valor a Pagar RS: 6.00".

Field	Value
Valor por 15min de Uso RS	2.00
Tempo de Uso do Cliente	35
Valor a Pagar RS	6.00

Exercício 05

- Um supermercado está com uma promoção – Para aumentar suas vendas no setor de higiene, cada etiqueta de produto deve exibir uma mensagem anunciando 50% de desconto (para um item) na compra de três unidades do produto. Elaborar um programa que leia descrição e preço de um produto. Após, apresente as mensagens indicando a promoção – conforme o exemplo ilustrado na Figura:



The screenshot shows a web browser window titled 'Exercício 2.c' with the address bar displaying 'file:///C:/livrojs/cap2/resp2_c.html'. The page content includes the heading 'Supermercado JS', a form with two input fields labeled 'Produto:' and 'Preço RS:', and a button labeled 'Ver Promoção'. Below the form, the text 'Escova Dental - Promoção: Leve 3 por RS: 10.00' and 'O 3º produto custa apenas RS: 2.00' is displayed.

Supermercado JS

Produto:

Preço RS:

Escova Dental - Promoção: Leve 3 por RS: 10.00

O 3º produto custa apenas RS: 2.00

Dicas

O atalho do Visual Studio Code para aplicar essa formatação aos comandos do seu programa é **Alt + Shift + F**. Você pode obter um resumo com as teclas de atalho disponíveis no editor acessando o menu Help / Keyboard Shortcuts Reference.



Dicas

Math.abs(num)	Retorna o valor absoluto de um número, ou seja, se o valor for negativo, ele será convertido para positivo. Se positivo, o valor permanece o mesmo. Exemplo: Math.abs(-3) => 3 Veja uma aplicação de Math.abs() no Exemplo 8.3
Math.ceil(num)	Arredonda o valor para cima. Dessa forma, se o valor possuir decimais, retorna o próximo número inteiro do valor analisado. Exemplo: Math.ceil(4.2) => 5 Veja uma aplicação de Math.ceil() no Exemplo 9.2
Math.floor(num)	Arredonda o valor para baixo, retornando a parte inteira do número. Exemplo: Math.floor(7.9) => 7 Veja uma aplicação de Math.floor() no Exemplo 2.2
Math.pow(base, exp)	Retorna a base elevada ao expoente. Exemplo: Math.pow(3, 2) => 9 Veja uma aplicação de Math.pow() no Exemplo 3.2
Math.random()	Retorna um número aleatório entre 0 e 1, com várias casas decimais. O número aleatório possível inicia em 0 e vai até um valor inferior a 1. Exemplo: Math.random() => 0.6501314074022906 Veja uma aplicação de Math.random() no jogo do Exemplo 5.2

Math.round(num)	Arredonda o valor para o inteiro mais próximo. A partir de .5 na parte fracionária, o valor é arredondado para cima. Anterior a .5, é arredondado para baixo. Exemplo: Math.round(2.7) => 3 Veja uma aplicação de Math.round() no Exemplo 6.10
Math.sqrt(num)	Retorna a raiz quadrada do número (square root). Exemplo: Math.sqrt(16) => 4 Veja uma aplicação de Math.sqrt() no Exemplo 3.5

SE 
LIGA

Dicas

- Limpar campos de formulário

```
function limparCampos() {  
  // Limpa os conteúdos dos elementos  
  document.getElementById("inNome").value = "";  
  document.getElementById("rbMasculino").checked = false;  
  document.getElementById("rbFeminino").checked = false;  
  document.getElementById("inAltura").value = "";  
  document.getElementById("outResposta").textContent = "";  
  // posiciona (joga o foco) no elemento inNome  
  document.getElementById("inNome").focus();  
}  
var btLimpar = document.getElementById("btLimpar");  
btLimpar.addEventListener("click", limparCampos);
```

```
function limparCampos() {  
  // recarrega a página  
  location.reload();  
  // posiciona (joga o foco) no elemento inNome  
  document.getElementById("inNome").focus();  
}  
var btLimpar = document.getElementById("btLimpar");  
btLimpar.addEventListener("click", limparCampos);
```

SE 
LIGA

Referências

ALMEIDA, Flávio. Cangaceiro JavaScript: Uma aventura no sertão da programação. São Paulo: Ed. Casa do Código, 2017.

FREEMAN, Eric; Robson, Elisabeth. Use a Cabeça! Programação Javascript. Rio de Janeiro. Editora Alta Books, 2016.

IEPSEN, Edécio Fernando. Lógica de Programação e Algoritmos com JavaScript: Uma introdução à programação de computadores com exemplos e exercícios para iniciantes. São Paulo: Novatec, 2018.

Bibliografia Complementar

ALVES, William Pereira. Desenvolvimento de aplicações web com Angular. Rio de Janeiro: Ed. Alta Books, 2018.

EIS, Diego. Guia Front-End: O caminho das pedras para ser um devFront-End. São Paulo: Ed. Casa do Código, 2015.

MOLINARI, Willian. Desconstruindo a Web. As tecnologias por trás de uma requisição. São Paulo: Ed. Casa do Código, 2016.

Nieradka, Itamar Pena. Javascript + CSS + DOM: Desenvolvimento para Web. Rio de Janeiro: Ed. Novaterra, 2016.

