Interface gráfica do usuário, Site

Descrição gerada automaticamente

**DESAFIO 3**

No desafio anterior, você aplicou a linguagem de folha de estilos do *bootstrap* e do CSS na estrutura do HTML5 já desenvolvido. Nesta etapa, desenvolverá aplicações e funcionalidades nas páginas, para deixá-las mais dinâmicas, a fim de mostrar mensagens e informações que permitam uma melhor interação com os usuários.

Para isso, no desafio 3, você deverá:

* **Aplicar a linguagem de programação JavaScript para implementar eventos no código em HTML que constituirão o site**.

Para este desafio, você estudará os seguintes conteúdos:

* **Linguagem de programação JavaScript (JS)**
  + **Sintaxe básica**
  + **Variáveis**
  + **Constantes**
  + **Operadores**
  + **Laços de repetição**
  + **Laços condicionais**
  + **Classes**
  + **Função**
  + **Bibliotecas**
  + **Manipulação de arquivos**
  + **Conversão de arquivos**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**JAVASCRIPT**

Quando desenvolvemos um site, preparamos o Front-End aplicando a linguagem de marcação HTML e estilizamos suas páginas, por meio do CSS3 e do *bootstrap*, para que elas apresentem uma estrutura adequada e um visual agradável ao usuário. No entanto, utilizando apenas o HTML e o CSS, obtemos apenas páginas estáticas, sem interação com o usuário.

Para adicionarmos interatividade às páginas, tornando-as funcionais, precisamos aplicar uma linguagem de programação que permita ao usuário executar ações, como exibir informações em menus, janelas modais, animações, infográficos interativos, mapas, preencher e enviar informações em um formulário, entre outras possibilidades.

O JavaScript é uma linguagem de programação utilizada para criar páginas de sites que sejam dinâmicas e executadas em navegadores. Trata-se de uma linguagem de alto nível, ou seja, sua sintaxe é mais intuitiva e amigável ao programador.

O JavaScript não se restringe apenas às páginas e aos navegadores. Hoje, existem diversos *frameworks*, com estruturas de códigos prontos a serem seguidos, e APIs (*Application Programming Interface*, em português, Interface de Programação de Aplicativo), com rotinas e padrões de programação, além de bibliotecas com centenas de funções, que possibilitam utilizar o JavaScript em aplicativos mobile, softwares para desktop e até mesmo no desenvolvimento de aplicações Back-End.

Siga em frente e conheça como aplicar a linguagem JavaScript para implementar eventos no código em HTML, deixando as páginas web dinâmicas e com boa performance.

**IMPORTANTE**

Não confunda JavaScript e Java, pois são duas linguagens de programação diferentes.

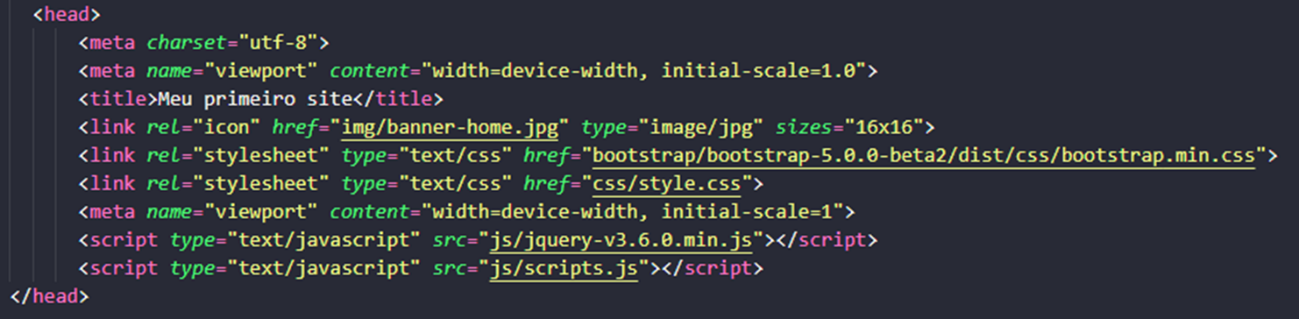
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

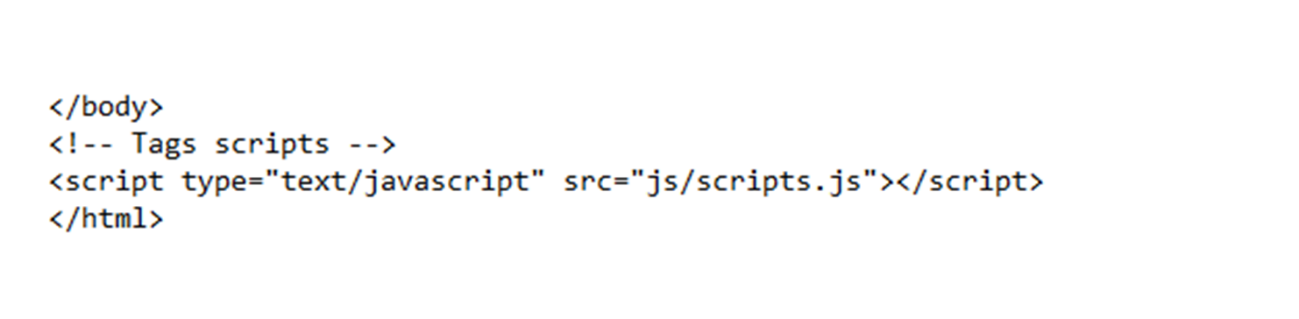
**SINTAXE BÁSICA**

No JavaScript, as instruções (ou comandos) são chamadas de declaração. Uma instrução pode conter várias linhas, que serão lidas (linha por linha) e interpretadas pelo navegador, sequencialmente. O final de cada instrução deve ser indicado por ponto e vírgula ( ; ).

O código JavaScript pode ser escrito na tag **<script>**, dentro da tag **<head>**, mas é recomendado inseri-lo após a tag **<body>** do HTML para otimizar o carregamento da página.

[[Desenho de um círculo

Descrição gerada automaticamente com confiança média](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/script.png)](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/script.png)

[[Desenho de um círculo

Descrição gerada automaticamente com confiança média](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/script-body.png)](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/script-body.png)

Assim como o CSS, o JavaScript pode ficar em um arquivo externo e ser linkado ao HTML. Dessa forma, cada tipo de código terá seu próprio arquivo.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**UM OLHAR PRÁTICO**

Faça, agora, um teste prático, criando uma pasta para o arquivo JavaScript. Dentro dele, insira um comando simples. No HTML, chame o arquivo .js, renderize o HTML no navegador e confira o código JavaScript.

**DICA**

Para comentar no JavaScript, use:

// para comentários de uma linha  
/\* comentários de várias linhas \*/

**VOCÊ SABIA?**

No Javascript, as aspas usadas no texto exibido em tela podem ser aspas simples ( ' ) ou aspas duplas ( " ).

Embora as duas formas sejam válidas, o ideal é seguir o mesmo padrão pelo projeto todo.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**VARIÁVEIS**

Variáveis são elementos que armazenam dados de diferentes tipos (que serão utilizados em um algoritmo) na memória do computador.

Para criar uma variável em JavaScript, basta declará-la, digitando:

* a palavra "**var**" com o sinal de igual ( **=** );
* o identificador;
* o valor que queremos atribuir a ela;
* finalizar com ponto e vírgula ( **;** ).

var = 7;

Exemplo:

Os valores das variáveis podem ser um nome, um número, um caractere, entre outros. O **var** é acompanhado por um identificador escolhido pelo usuário. Esse identificador é único e armazena os valores passados para a variável. No exemplo, as aspas serão utilizadas para valores de texto, como uma **string** ou um caractere (V ou F, por exemplo). Para números (inteiros ou decimais), não é necessário utilizar aspas.

Interface gráfica do usuário

Descrição gerada automaticamente

Veja a seguir um exemplo do uso de variáveis, fazendo um texto aparecer ou sumir ao clicar em um botão. Leia com atenção os comentários do código.

<!doctype html>

<html>

<head>

</head>

<body>

<button onclick="myFunction()">Clique aqui</button>

<!-- botão que apaga o texto ao ser clicado -->

<div id="contextText"><p>Texto que vai sumir</p></div>

<!-- div com id -->

</body>

<script>

function myFunction(){

var x = document.getElementById('contextText');

/\* a variável x é o texto dentro da div com id="contextText" \*/

if (x.style.display ==='none'){

/\* se o sytle da div for none, ou seja, estiver invisível... \*/

x.style.display = 'block';

/\* ... então faça o style ser block, ou seja, ficar visível... \*/

} else {

x.style.display = 'none';

/\* ... caso contrário, (se já estiver visível) faça sumir.\*/

}

}

</script>

<!-- teste em seu arquivo scripts.js. Coloque o script no scripts.js e faça o link no HTML. -->

</html>

Outros valores para variáveis são:

* **Boolean**: é um valor lógico, verdadeiro ou falso.
* **Null**: palavra-chave que indica um valor nulo.
* **Undefined**: propriedade que não possui valor definido.
* **Number**: número, inteiro ou decimal.
* **String**: texto. Tudo que estiver dentro de aspas será considerado: palavras, caracteres, números, símbolos e espaços.
* **Caractere**: uma letra.
* **Symbol**: tipo de dado, no qual as instâncias são únicas.

Outro tipo de variável são as ***arrays***. *Array* é uma lista de zero ou mais expressões inseridas dentro de colchetes ( **[ ]** ). Você pode passar mais de um valor por *array*, por exemplo:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

]

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Constante é um tipo de variável cujo valor é fixo, ou seja, não terá seu valor alterado durante uma atribuição ou declaração, por exemplo, o valor de PI ou a data de nascimento do usuário.

As regras para definir o nome de uma constante são as mesmas utilizadas para uma variável. A diferença é que a palavra **var** será substituída por **const**. Observe:

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**LAÇOS DE REPETIÇÃO**

Laços ou estruturas de repetição (também chamados de loops) são blocos de código que, de acordo com uma condição (ou contador), repetem uma ação. Os comandos de repetição mais comuns são **while**, **do while** e **for**:

**WHILE**

Esse comando é utilizado, normalmente, quando não se sabe o número exato de interações. O contador (ou condição) é modificado ao final do loop.

Uma imagem contendo Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente

Veja um exemplo de estrutura condicional e leia os comentários do código com atenção.

<!doctype html>

<html>

<head>

<script src="https://code.jquery.com/jquery-latest.min.js"></script>

</head>

<body>

<form action="javascript:void(0);">

<label>Escolha seu sexo</label>

<input type="radio" name="sexo" id="feminino"><label>Feminino</label>

<input type="radio" name="sexo" id="masculino"><label>Masculino</label>

<input type="submit" id="button">

</form>

</body>

<script>

$('#button').click(function(){

if($('#feminino').is(':checked')){

alert("Você escolheu feminino");

}else if($('#masculino').is(':checked')){

alert('Você escolheu masculino');

}else{

alert('Você não escolheu');

}

});

</script>

</html>

Também é possível utilizar os operadores lógicos nas estruturas condicionais.  
Clique nas abas a seguir para ver exemplos.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Texto

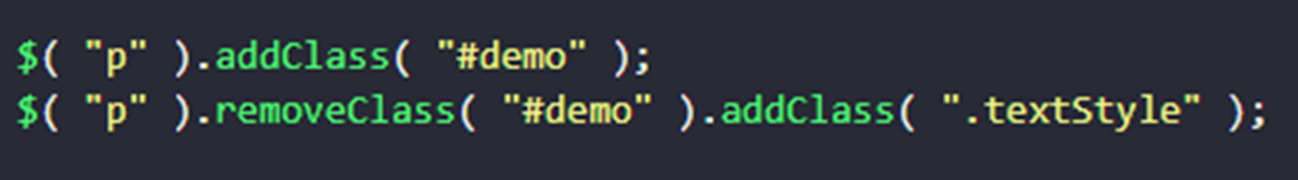
Descrição gerada automaticamente

**CLASSES**

As classes são métodos nos quais ocorre uma mudança de **class** atribuídas no HTML. Temos três elementos em que você pode utilizá-las: **.addClass()**, **.hasClass()**, **.removeClass()**.

**.ADDCLASS( ) E .REMOVECLASS( )**

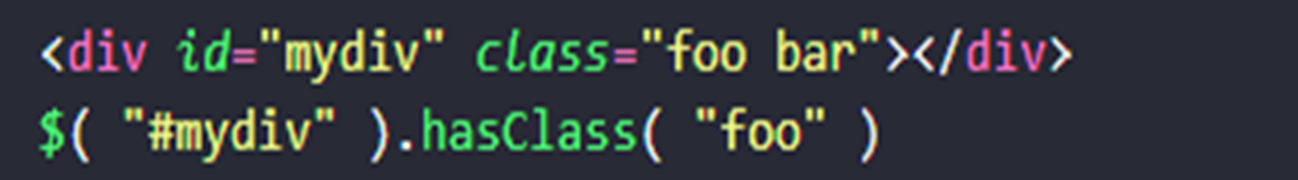
**.addClass** e **.removeClass** adicionam e removem, respectivamente, a **class** dos elementos do HTML, alterando seu estado, ou seja, sua estilização.

[[Desenho de um círculo

Descrição gerada automaticamente com confiança média](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/remove-class.png)](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/remove-class.png)

**.HASCLASS( )**

**.hasClass** é um método que retorna **true** ou **false**, pois ele irá comparar a **class** atribuída ao elemento do HTML.

[[](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/hasclass.png)](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/hasclass.png)

Um exemplo prático para o uso das classes ocorre quando você tem uma **div** pai que contém diversas tags **p** e você deseja estilizar apenas uma. Nessa situação, você pode adicionar uma **class** para essa tag específica e, no JavaScript, você irá chamá-la utilizando esse método e passando como condição a estilização.

## FUNÇÕES

Em algumas situações, é necessário executar a mesma tarefa várias vezes. Para evitar a **redundância desse bloco de códigos, uma função é utilizada**.

A função é iniciada pela palavra **function** e seguida pelo nome da função. Para criar o nome de uma função, também são aplicadas as mesmas regras para a criação de um nome de uma variável.

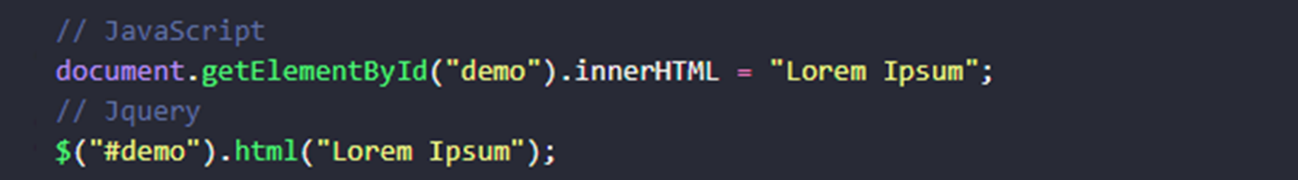
Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**BIBLIOTECAS**

Bibliotecas são compostas de funções pré-escritas, nas quais é possível utilizar o código sem implementar algo novo. Dessa forma, é possível chamar a função e utilizá-la em diversos locais do código para realizar alguma ação em sua aplicação.

Neste curso, você conhecerá a biblioteca de funções do JavaScript Jquery, que possibilita simplificar o código de funções que incrementarão a aplicação web, fazendo manipulação do [DOM](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/aula-3.html) (HTML final), o que permite deixar o código interativo e funcional. A imagem a seguir contém um exemplo:

[[Desenho de um círculo

Descrição gerada automaticamente com confiança média](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/lib-javascript.png)](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/lib-javascript.png)

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

### MANIPULAÇÃO DO DOM

Manipulação do DOM é toda alteração ou interação ocorrida em um ou mais elementos do HTML, ocasionando uma mudança de estado. Por exemplo, um clique em um botão, no qual será aplicado um evento, pode ativar uma função que altera o tamanho de uma imagem, cor de fundo do site, fonte de um texto ou qualquer elemento na página HTML.

Interface gráfica do usuário, Texto, Aplicativo, Email

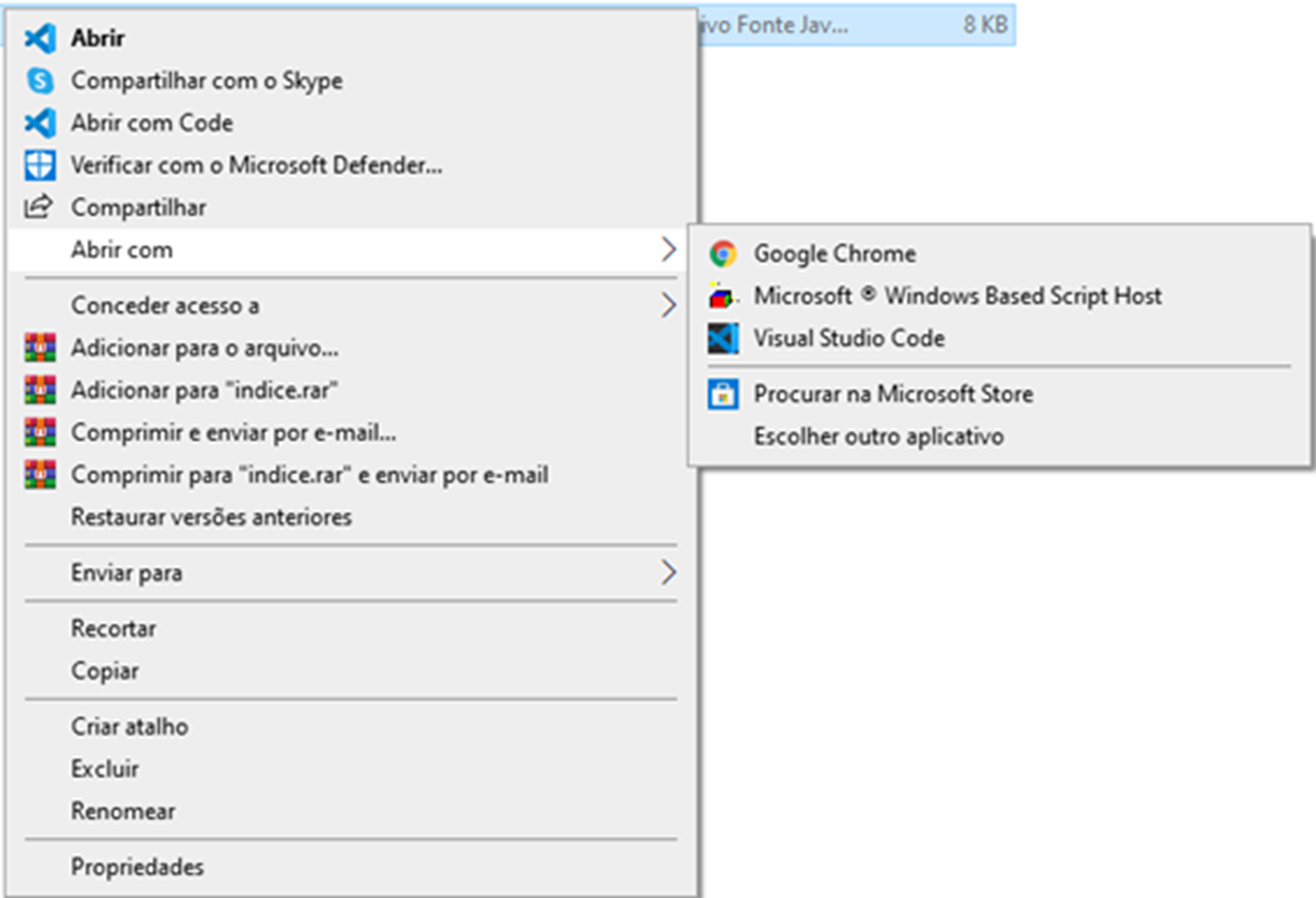
Descrição gerada automaticamente

**EXTENSÃO DE ARQUIVO JAVASCRIPT**

O JS é uma linguagem de programação interpretada, sendo implementada em browsers. Quando for preciso utilizar código em JS fora do arquivo HTML, podemos salvá-lo de forma separada utilizando extensão .js e chamar o arquivo criado usando a tag <script>.

Caso deseje abrir o arquivo e não tenha um programa apropriado para ler o código (IDE), o Windows retornará com uma mensagem de erro de leitura. Caso isso aconteça, você pode alterar a maneira como o arquivo será aberto. **Clique com o botão direito do mouse sobre o arquivo e selecione a “Abrir com”, conforme mostra a figura ao lado**.

Você pode clicar em “Escolher outro aplicativo” e alterar sua versão de leitura.

[[](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/windows-file.png)](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/img/aula-03/windows-file.png)

**DICA**

Você pode testar todos os códigos em seu scripts.js, para isso, verifique a inspeção do navegador quando renderizar o HTML.

Para isso, pressione CTRL + Shift + J. Na janela que abrirá, localize a aba Console. Então, cole o código Javascript que deseja testar e aperte Enter. Pronto, o algoritmo será executado pelo navegador.

Também é possível acessar o DevTools (Ferramenta de Desenvolvedor) com a tecla F12 e então fazer o mesmo procedimento de copiar e colar o algoritmo Javascript.

**VAMOS PRATICAR**

**1.** Considere a seguinte situação: você deverá acrescentar o conteúdo de parágrafo do seu código HTML em um arquivo JavaScript externo. Identifique a alternativa que apresenta a sintaxe adequada para a realização dessa tarefa.

Parte superior do formulário

1. <!-- HTML -->
2. <p id ="txtrect"></p>;
3. //JS

document.querySelector("p id").innerHTML ="Teste";

1. <!-- HTML -->
2. <p id ="txtrect"></p>;
3. //JS

document.querySelector(".txtrect").innerHTML ="Teste";

1. <!-- HTML -->
2. <p id ="txtrect"></p>;
3. //JS

document.querySelector(#txtrect).addText ="Teste";

1. <!-- HTML -->
2. <p id ="txtrect"></p>;
3. //JS

document.querySelector("#txtrect").innerHTML ="Teste";

1. <!-- HTML -->
2. <p id ="txtrect"></p>;
3. //JS

document.querySelector(txtrect).addText ="Teste";

Parte inferior do formulário

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**2.** Analise a sintaxe a seguir e escolha a alternativa correta:

function multiplica(num1,num2) {

var resultado = num1 \* num2;

return resultado;

}

multiplica(10,88)

Parte superior do formulário

1. 88
2. 880
3. 10,88
4. 1088

Parte inferior do formulário

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

**3.** Dado o laço de condição a seguir, identifique a alternativa que apresenta qual será a saída correta.

x = null;

y = undefined;

if (x==y){

console.log('os valores são iguais')}

else {console.log('os valores são diferentes')}

Parte superior do formulário

1. Uncaught Syntax Error.
2. Os valores são iguais.
3. Os valores são diferentes.
4. Null.
5. Undefined

Parte inferior do formulário