JavaScript é uma linguagem client-side, ela roda no front-end.

O ideal do J é manipular o DOM (Modelo de Objeto de Documento) e validação de formulários (no sentido de evitar requisições desnecessárias ao servidor) -> desafoga o servidor

Pode inserir o codigo JS de várias formas, igual o CSS.

- tag script -> forma mais comum, geralmente fica no final da página ou na head

- tag script com a tag src -> indica o caminho do arquivo JS

- Eventos HTML e JS, como on-click.. -> é basicamente um conjunto de ações em um determinado elemento da página.

var nomeCompleto = "";

var professor =

{

nome: "Fabricio",

sobrenome: "Londero"

};

for (p in professor) {

nomeCompleto += professor[p] + " ";

}

alert(nomeCompleto);

(Esse trecho de código cria um objeto professor com duas propriedades: nome e sobrenome.

Em seguida, um loop for..in (foreach) é usado para percorrer as propriedades do objeto professor e concatenar seus valores na variável nomeCompleto.

A cada iteração do loop, a variável p recebe o nome da propriedade atual, que é usada para acessar o valor correspondente no objeto professor. O valor retornado é então concatenado com um espaço em branco e adicionado à variável nomeCompleto.

O valor de nomeCompleto é exibido em um alerta. O resultado final será uma string contendo o nome completo do professor)

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

<https://www.w3schools.com/jsref/dom_obj_event.asp>.

O DOM "Event" é um objeto do DOM (Document Object Model) que representa um evento, como um clique do mouse, uma tecla pressionada, um carregamento de página concluído, entre outros.

O "Event" é criado automaticamente quando um evento ocorre e pode ser acessado através de um parâmetro de retorno de uma função de manipulador de eventos. Esse objeto contém informações sobre o evento que ocorreu, como o tipo de evento, a posição do mouse no momento do evento, a tecla pressionada, entre outros.

O "Event" possui várias propriedades e métodos úteis que podem ser usados para manipular o evento e realizar diferentes ações. Por exemplo, a propriedade "type" contém o tipo de evento que ocorreu, a propriedade "target" contém o elemento HTML em que o evento ocorreu e o método "preventDefault()" pode ser usado para cancelar o comportamento padrão do evento.

<!DOCTYPE *html*>

<html>

<head>

    <script>

EXERCÍCIO 3: Adaptar esse exemplo para ele mostrar quantas vezes o botão foi clicado. quando atingir 5 cliques, não mostra mais mensagens.

//<button onclick="alert('Botão clicado!');">Ok!</button>

var clicks = 0; // continua no body

    </script>

</head>

<body>

<p>clique no botão</p>

<button *onclick*="clicks++; (clicks <= 5)?alert(clicks):console.log(clicks);">Ok!</button>

</body>

</html>

Nesse código, primeiro é definida uma variável **clicks** fora do **onclick** para que ela possa ser acessada e atualizada a cada clique no botão.

O **onclick** é uma função que é executada quando o botão é clicado. Quando o botão é clicado, a função é executada e o valor de **clicks** é incrementado em 1. Em seguida, é verificado se o número de cliques é menor ou igual a 5. Se for, é exibida uma caixa de mensagem com o número de cliques. Caso contrário, o número de cliques é exibido no console.

Basicamente, essa solução usa a expressão condicional ternária **?** para verificar se o número de cliques é menor ou igual a 5. Se for, exibe o número de cliques usando o **alert**. Caso contrário, exibe o número de cliques no console usando **console.log**.