

JavaScript – Document Object Model e Eventos

1) Manipulando o DOM com getElementById

- Crie um documento HTML básico com uma estrutura <html>, <head>, <body>.
- Adicione um elemento <p id="texto">Texto original</p> no corpo da página.
- Em um script JavaScript externo ou no final do corpo, use o método getElementById para alterar o texto desse parágrafo para "Texto atualizado via DOM".

Objetivo: Usar getElementById para manipular o conteúdo de um elemento específico no DOM.

Dica: https://www.w3schools.com/jsref/met_document_getelementbyid.asp

2) Usando o Objeto window

- Adicione um botão ao documento com o texto "Clique para receber uma mensagem".
- Escreva uma função que, ao clicar no botão, exiba um alert com uma mensagem personalizada, como "Bem-vindo(a) ao nosso site!".
- Adicione também um console.log para exibir a mesma mensagem no console do navegador.

Objetivo: Usar método alert para interagir com o usuário.

Dica: https://www.w3schools.com/jsref/met_win_alert.asp

3) Alterando o Estilo do Documento com o Objeto document

- Crie um botão com o texto "Mudar Cor de Fundo".
- Adicione uma função que, ao clicar no botão, altere a cor de fundo (backgroundColor) do documento para azul.
- Adicione lógica condicional para que, se o fundo já estiver azul, ele volte à cor original (para praticar a alternância de estilos).

Objetivo: Alterar o estilo do documento dinamicamente.

Dica: https://www.w3schools.com/jsref/prop_style_backgroundcolor.asp

4) Exibindo a URL Atual com o Objeto location

- Adicione um botão com o texto "Mostrar URL".
- Crie uma função que, ao clicar no botão, exiba a URL atual do documento dentro de um parágrafo ou div criado especialmente para isso.

Objetivo: Utilizar a propriedade href para acessar informações da URL.

Dica: https://www.w3schools.com/jsref/obj_location.asp

5) Navegando no Histórico com o Objeto history

- Adicione um botão com o texto "Voltar à página anterior".
- Atribua uma função ao evento click do botão para utilizar o método back do objeto history.
- Para evitar erros, verifique antes se há um histórico anterior e exiba uma mensagem caso não haja.

Objetivo: Praticar a manipulação do histórico de navegação.

Dica: https://www.w3schools.com/jsref/obj_history.asp

6) Verificando Cookies com o Objeto navigator

1. Adicione um quinto botão chamado "Verificar Cookies".
2. Crie uma função que verifica se os cookies estão habilitados (navigator.cookieEnabled) e exibe uma mensagem informando o resultado.
3. Associe a função ao botão usando addEventListener.

Objetivo: Explorar o uso do navigator para obter informações sobre o navegador do usuário.

Dica: https://www.w3schools.com/jsref/obj_navigator.asp

7) Alterando a Cor com o Evento onmouseover

- Adicione uma caixa (<div>) com uma cor de fundo padrão e algum texto.
- Defina uma função para alterar a cor de fundo da caixa quando o mouse passar sobre ela (onmouseover) e uma função para restaurar a cor original (onmouseout).

Objetivo: Implementar uma interação visual com eventos de mouse.

Dica: https://www.w3schools.com/jsref/event_onmouseover.asp

8) Detectando Teclas Pressionadas com o Objeto event

- Adicione um campo de entrada de texto (<input type="text">).
- Crie uma função que responda ao evento onkeydown exibindo uma mensagem no console, indicando qual tecla foi pressionada. Use a propriedade key do objeto event para capturar o valor da tecla.
- Extra: Faça com que o campo de entrada limite a inserção de certas teclas (por exemplo, impedir números), usando lógica condicional.

Objetivo: Usar o evento onkeydown para capturar a entrada do usuário em tempo real.

Dica 1: https://www.w3schools.com/jsref/event_onkeydown.asp

Dica 2: https://www.w3schools.com/jsref/event_key_key.asp

Boa prática e bom estudo do DOM e dos eventos no JavaScript!