**Para saber mais: manipulando classes com ClassList**

O JavaScript é uma linguagem de programação amplamente utilizada para adicionar interatividade e dinamismo às páginas da web. Uma das tarefas comuns na manipulação da interface do usuário é a alteração de classes CSS de elementos HTML. Anteriormente, a forma tradicional de fazer isso era manipular a propriedade className, que envolvia trabalhar com strings e pode ser propenso a erros. No entanto, com a introdução do método **classList**, a partir do ECMAScript 5, essa tarefa se tornou mais fácil e segura.

O **[classList](https://www.w3schools.com/jsref/prop_element_classlist.asp" \t "_blank)** é uma propriedade do JavaScript que representa uma lista de classes CSS. Ele fornece métodos que facilitam a adição, remoção e verificação de classes, tornando a manipulação de classes CSS mais eficiente e menos suscetível a erros de programação.

Além do que apresentei para você em vídeo, vamos conhecer mais formas de usar o **classList**?

**Adicionando uma classe**

Para adicionar uma classe a um elemento HTML, podemos usar o método **add()** do **classList**. Este método aceita o nome da classe como argumento e adiciona a classe ao elemento, se ela ainda não estiver presente. Conforme exemplo a seguir:

**const** element = document.getElementById('meuElemento');

element.classList.**add**('minhaClasse');

**Removendo uma classe**

Para remover uma classe de um elemento HTML, podemos utilizar o método **remove()** do **classList**. Este método aceita o nome da classe como argumento e remove a classe do elemento, se ela estiver presente. Veja o exemplo abaixo:

**const** element = document.getElementById('meuElemento');

element.classList.**remove**('minhaClasse');

**Alternando uma classe**

O método **toggle()** do **classList**permite alternar uma classe em um elemento. Se a classe já estiver presente no elemento, o método a remove; caso contrário, ele a adiciona, conforme exemplo a seguir:

**const** element = document.**getElementById**('meuElemento');

element.classList.**toggle**('minhaClasse');

**Verificando se uma classe está presente**

Para verificar se uma classe específica está associada a um elemento, podemos usar o método **contains()** do **classList**, como no exemplo:

**const** element = document.**getElementById**('meuElemento');

**if** (element.classList.**contains**('minhaClasse')) {

// A classe 'minhaClasse' está presente no elemento

// Faça algo aqui...

}

**Substituindo classes**

Como feito em aula, podemos substituir uma classe por outra usando os métodos **add()** e **remove()** em sequência.

**const** element = document.getElementById('meuElemento');

element.classList.**remove**('classeAntiga');

element.classList.**add**('classeNova');

**Manipulando várias classes simultaneamente**

É possível adicionar ou remover várias classes de uma vez usando o método **add()** ou **remove()** passando vários argumentos separados por vírgula. Veja o exemplo:

**const** element = document.getElementById('meuElemento');

element.classList.**add**('classe1', 'classe2', 'classe3');

element.classList.**remove**('classe2', 'classe3');

O **classList** em JavaScript é uma ferramenta poderosa e eficiente para manipular classes CSS em elementos HTML. Com os métodos **add(),** **remove(), toggle(), contains()**e outras funcionalidades, podemos realizar tarefas de manipulação de classes de forma mais segura e simplificada, tornando nossas aplicações web mais interativas e dinâmicas.

Nota: Certifique-se de que o JavaScript seja carregado após a renderização do DOM, para que os elementos estejam acessíveis antes de tentar manipular suas classes.

**Para saber mais: utilizando áudio em páginas da web**

O JavaScript é uma linguagem de programação amplamente utilizada para criar interatividade em páginas da web. Uma das funcionalidades mais interessantes que ele oferece é a capacidade de manipular áudio, permitindo que os desenvolvedores adicionem e controlem elementos de áudio em seus sites de forma dinâmica. Uma das formas mais comuns de fazer isso é usando o objeto ‘Audio’.

O objeto ‘Audio’ é uma parte do HTML5 e fornece uma interface fácil e poderosa para reproduzir e controlar arquivos de áudio em tempo real. Que tal explorar o uso, entendendo como criar, controlar, reproduzir e estilizar áudio em uma página da web?

### ****Criando um Objeto ‘Audio’****

### Para criar um novo objeto ‘Audio’, podemos simplesmente usar a seguinte sintaxe:

**const** audioElement = **new** Audio('caminho/do/arquivo-de-audio.mp3');

### Substitua 'caminho/do/arquivo-de-audio.mp3' pelo caminho real do arquivo de áudio que você deseja reproduzir.

### ****Controles básicos de áudio****

Após criar um objeto ‘Audio’, podemos controlar sua reprodução e outras propriedades usando métodos e propriedades disponíveis.

* **play():** inicia a reprodução do áudio;
* **pause():** pausa a reprodução do áudio;
* **currentTime:** propriedade que retorna ou define a posição atual de reprodução do áudio, em segundos;
* **volume:** propriedade que retorna ou define o nível de volume do áudio, variando de 0 a 1.

### ****Exemplo de utilização dos métodos do objeto Audio:****

**const** audioElement = **new** Audio('caminho/do/arquivo-de-audio.mp3');

audioElement.play(); // Inicia a reprodução

audioElement.pause(); // Pausa a reprodução

audioElement.currentTime = 10; // Move para 10 segundos no áudio

audioElement.volume = 0.5; // Define o volume para metade (50%)

### Lembre-se de que a reprodução de áudio em páginas da web pode ser intrusiva para alguns usuários, portanto, use-o com moderação e sempre forneça controles de volume e opções de pausa aos visitantes do seu site.