

The background of the slide is a scenic landscape. It features a calm river in the foreground, reflecting the light from the sun. The middle ground is filled with a dense forest of evergreen trees. In the background, there are rugged mountains. The sky is a deep teal color, and a bright sunburst is visible behind the mountains, with a faint rainbow arching across the sky on the right side.

# JS

## Aula 04

Cristiano Gonçalves

# Manipulando Strings

- Split
- Search
- ToUpperCase
- ToLowerCase
- Concat
- ValueOf
- IndexOf
- Trim

# HTML



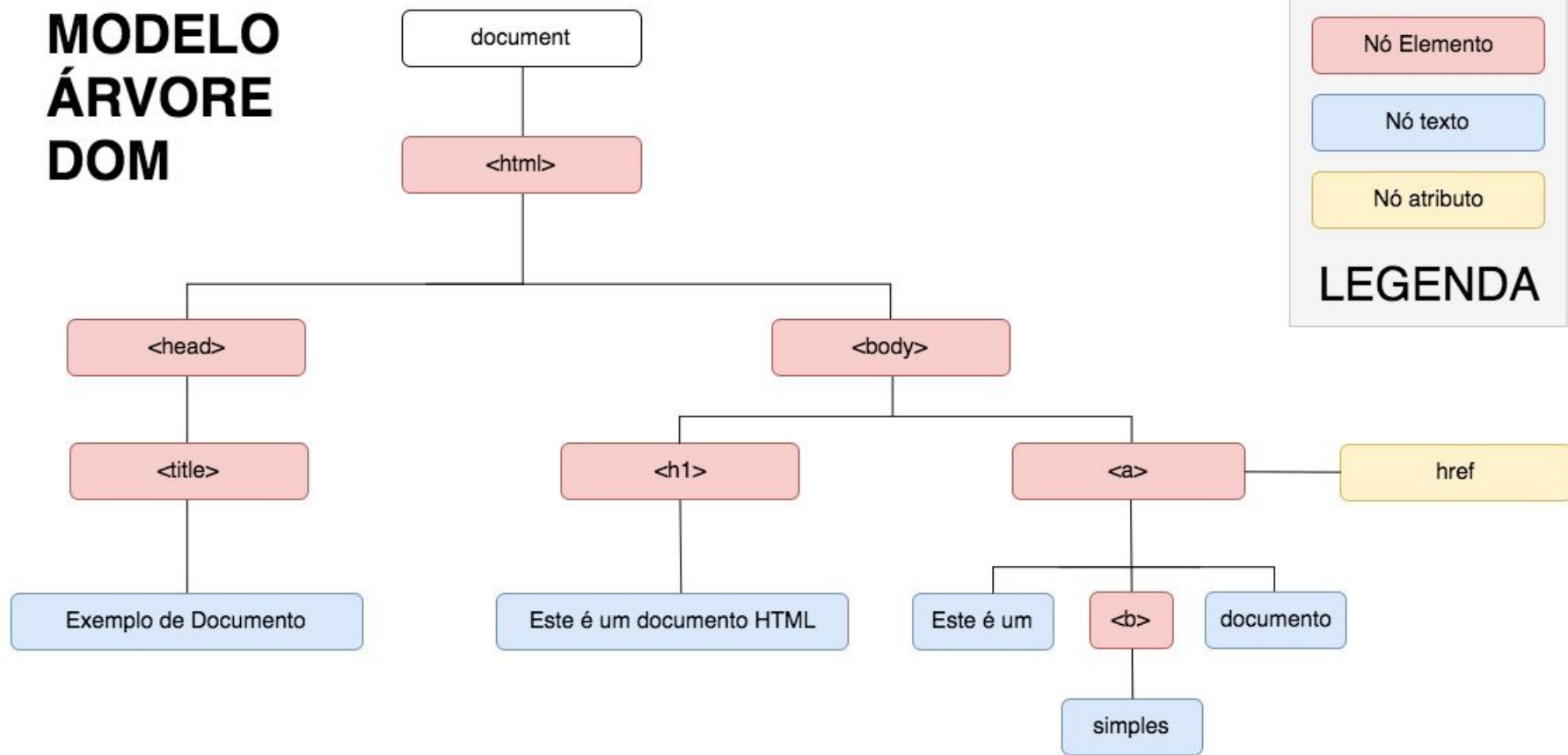
# CSS



# DOM - Document Object Model

O **Modelo de Objeto de Documento (DOM)** é uma interface de programação para documentos HTML, XML e SVG . Ele fornece uma representação estruturada do documento como uma árvore. O DOM define métodos que permitem acesso à árvore, para que eles possam alterar a estrutura, estilo e conteúdo do documento. O DOM fornece uma representação do documento como um grupo estruturado de nós e objetos, possuindo várias propriedades e métodos. Os nós também podem ter manipuladores de eventos que lhe são inerentes, e uma vez que um evento é acionado, os manipuladores de eventos são executados. Essencialmente, ele conecta páginas web a scripts ou linguagens de programação.

# MODELO ÁRVORE DOM



**Embora o DOM seja frequentemente acessado usando JavaScript, não é uma parte da linguagem JavaScript. Ele também pode ser acessado por outras linguagens.**

# Buscando elementos

```
document.getElementById('#id');  
document.getElementsByClassName('.classe');  
document.getElementsByTagName('p');
```

## Buscando elementos

```
document.querySelector('#id');  
document.querySelectorAll('.classe');
```



## Alterando elementos

```
elemento.innerHTML = 'Conteúdo';  
element.textContent = 'Outro conteúdo';
```

# Criando Elementos

```
let titulo = document.createElement('h1');  
titulo.innerText = 'Cristiano';
```

## Adicionando elementos

```
· pai.appendChild(filho);
```

# Adicionando elementos

```
element.insertAdjacentHTML(posição, texto);  
// -----  
// Antes do elemento.  
element.insertAdjacentHTML('beforebegin', '<p>Texto</p>');  
// Dentro do elemento, antes de seu primeiro filho (childNodes).  
element.insertAdjacentHTML('afterbegin', '<p>Texto</p>');  
// Dentro do elemento, após seu último filho (childNodes)  
element.insertAdjacentHTML('beforeend', '<p>Texto</p>');  
// Após o elemento.  
element.insertAdjacentHTML('afterend', '<p>Texto</p>');
```

## Removendo elementos

```
elemento.remove();
```

Hands on



# Referências

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

<https://github.com/cezaraugusto/You-Dont-Know-JS/>

<https://www.youtube.com/user/rodrigobranas>

<https://github.com/da2k/curso-javascript-ninja>