

## DOM - Trabalhando com strings dinâmicas (Aula 5)

```
const nome= 'Mauro';  
const meuTemplate = `Meu nome é ${nome}`;  
console.log(meuTemplate);
```

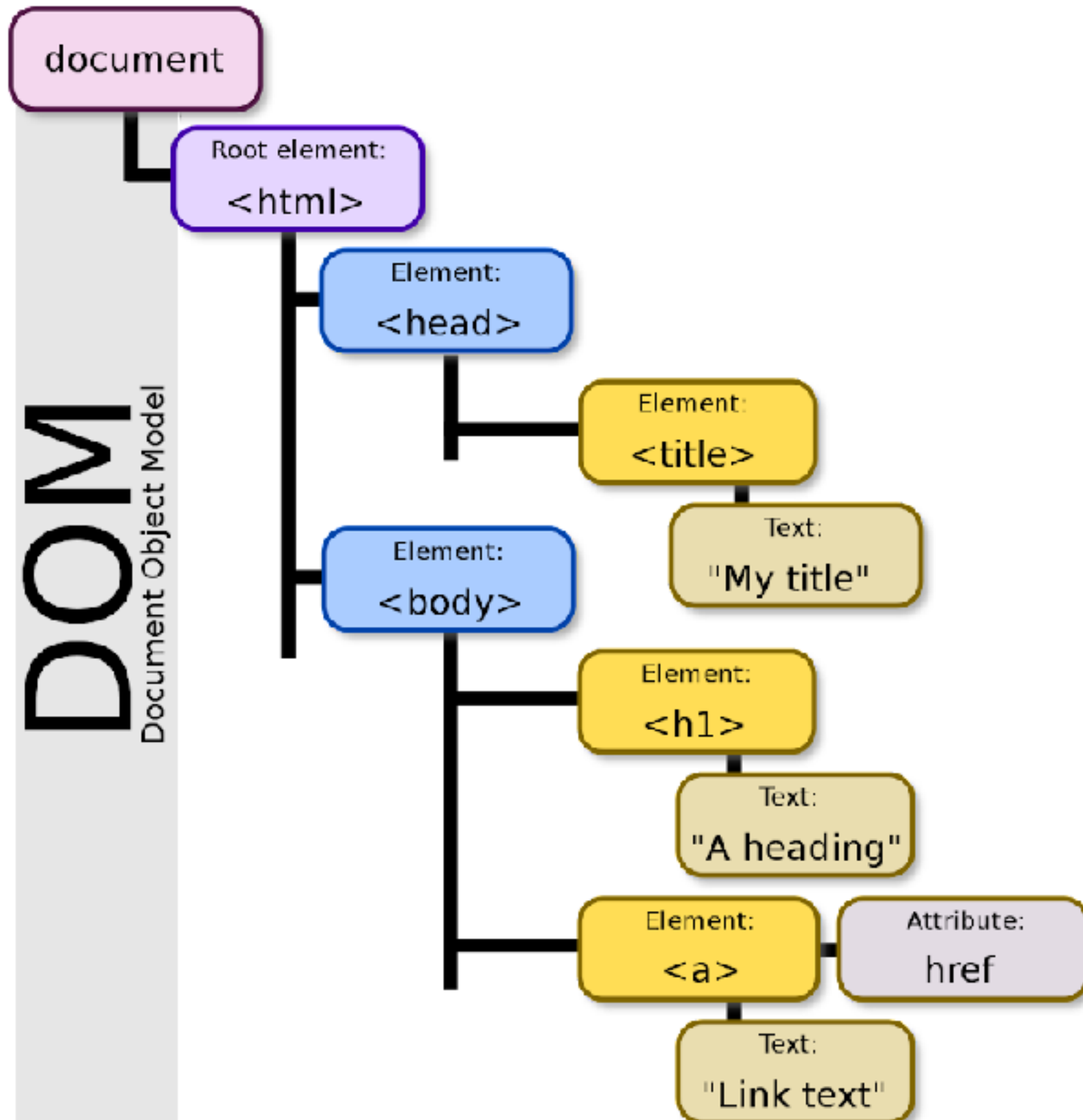
O resultado seria:  
**Meu nome é Mauro**

```
function soma(var1, var2) {  
    return var1 + var2  
}  
  
const temp = `O resultado é ${soma(2,2)}`;  
console.log(temp);
```

O resultado seria:  
**O resultado é 4**

# DOM (Aula 4)

Document Object Model



## Comparando Seletores

<code>querySelector()</code>	<code>querySelectorAll()</code>	<code>getElementById()</code>
Retorna o <b>primeiro elemento</b> do DOM que atende à condição que estamos procurando.	Retorna <b>todos os elementos</b> do DOM que atendem à condição que procuramos.	Retorna o elemento do DOM <b>que atende ao Id que estamos procurando</b> .

# DOM - Modificando elementos (Aula 5)

## .innerHTML

Se quisermos **modificar** o conteúdo de uma tag HTML, vamos usar esta propriedade:

```
{ } document.querySelector('div.nome').innerHTML = '<strong>Daniel</strong>';
```

## .innerText

Se quisermos **modificar** o texto de uma tag HTML, vamos usar esta propriedade:

```
{ } document.querySelector('div.nome').innerText = 'Maria';
```

## Propriedade Style

Nos permite ler e sobrescrever as regras CSS que são aplicadas a um elemento que selecionamos.

```
{ } let titulo = document.querySelector('.title');
    titulo.style.color = 'cyan';
    titulo.style.textAlign = 'center';
    titulo.style.fontSize = '12px';
    titulo.style.backgroundColor = '#dddddd';
```

## DOM - Manipulando 'nós' (Aula 7)

Método	
1. createElement()	Cria um nó de tipo de elemento de acordo com o nome da tag html que indicamos.
2. createTextNode()	Cria um nó de texto explícito entre aspas. Ele não é exibido até que você o atribua a um elemento existente no DOM.
3. appendChild()	Adere dentro do DOM um elemento filho a um elemento pai. Se o elemento pai já existia no documento, ele muda sua posição em relação ao outro elemento pai indicado. Se não existir, nós o criamos com o método 1.

## Sintaxe para criar um nó de elemento

### Objeto

Aqui especificamos o objeto do documento como um prefixo, sem esses dados o nó não será criado, por isso deve ser usado sempre.

### Elemento

Aqui, colocaremos o nome do nó que queremos criar no DOM. Por exemplo: "Botão", "div", "seção" etc.

js

```
document.createElement("input");
```

### Método

Aqui vamos definir o método que usamos para criar o nó de acordo com o tipo de que precisamos. Nesse caso, createElement para criar uma tag de entrada.

js

```
document.createTextNode("Olá Mundo");
```

js

```
document.body.appendChild(titulo);
```