

IOS – Instituto de
Oportunidade Social

Aula JS 11 - JavaScript HTML DOM – Parte 01



- > DOM
- > JavaScript HTML DOM
- > Objeto Window
- > Métodos do DOM
- > Para aprender mais

IOS – Instituto de
Oportunidade Social

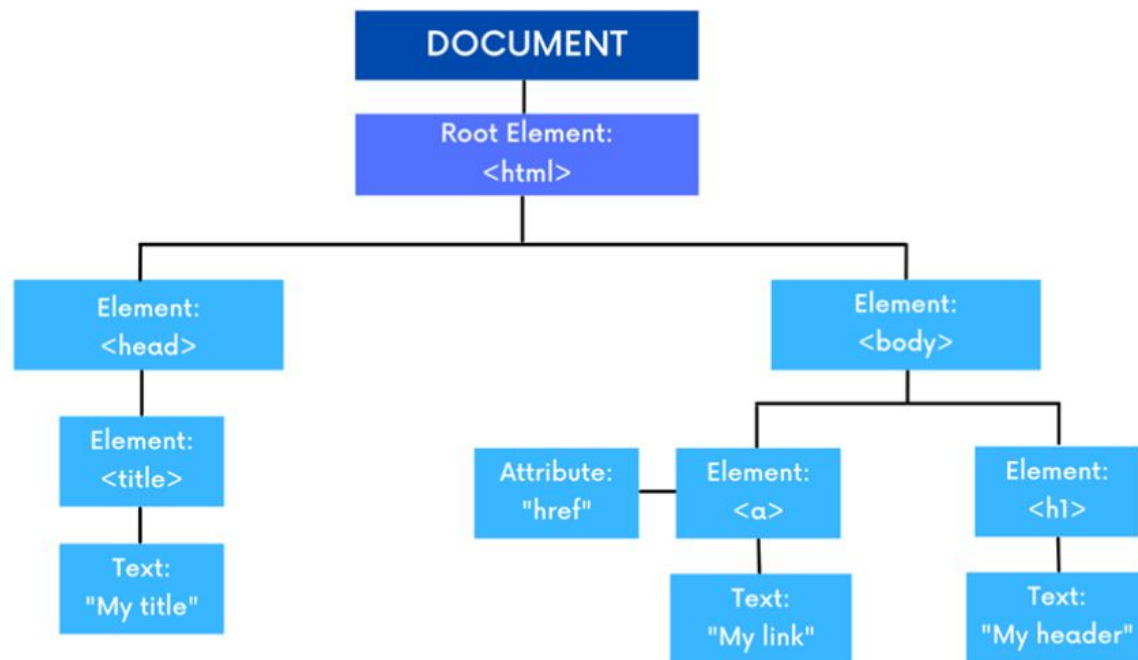
DOM



O **DOM** (**Document Object Model**) é uma interface de programação multiplataforma e independente de linguagem, que é usada em documentos **HTML**, **XHTML**, **XML** e **SVG**. Essa interface foi criada e é mantida pela entidade **World Wide Web Consortium** (**W3C**).

Basicamente, o **DOM** fornece uma representação estruturada do documento como uma árvore, onde os elementos do documento são como nós das ramificações da árvore. Para um documento HTML, a imagem abaixo mostra a árvore **DOM** com alguns elementos (**tag**). É importante você ver essa árvore na **língua inglesa mesmo**, pois o **HTML** e o **DOM** são baseados **no inglês** e você vai usar essas palavras na **língua original desses recursos**.

DOM - Document Object Model



> Vamos Praticar

E a pergunta que surge é: mas o **DOM** faz parte do **HTML** ou do **JavaScript**? **Na verdade, de nenhum**, ele é gerado pelo **browser**. Ao carregar a **página**, o navegador cria o **documento**, a **interface**, e o **Javascript** usa o **DOM** para se conectar ao **HTML**.

IOS – Instituto de
Oportunidade Social

JavaScript HTML DOM



Até o momento, **utilizamos o console para exibir as mensagens com o resultado do processamento de alguma informação no JavaScript**. O console é importante para você fazer o **debug** de um código, aprender lógica de programação de uma determinada linguagem e ser um recurso de visualização do estado de uma aplicação, mas o Javascript é muito mais do isso.

A principal aplicação do **JavaScript** é realizar a manipulação dos elementos do **DOM**, pois o **JavaScript** foi **criado para incorporar uma lógica computacional dentro de um ambiente de script**, que é o ambiente encontrado em um navegador web. Pois o navegador web consegue executar um script, mas não consegue executar um programa.

Com o DOM, Javascript tem o poder para criar conteúdo dinâmico em um documento HTML, permitindo o JS:

- alterar todos os elementos HTML na página,
- alterar todos os atributos HTML na página,
- mudar todos os estilos CSS na página,
- remover elementos e atributos HTML existentes,
- adicionar novos elementos e atributos HTML,
- reagir a todos os eventos HTML existentes na página e
- pode criar novos eventos HTML na página.

IOS – Instituto de
Oportunidade Social

Objeto Window



O **objeto window** representa a **janela do navegador** que contém o elemento **DOM**. Foi esse objeto que permitiu utilizarmos o método **alert()** para exibir mensagens no navegador. Vamos fazer um pequeno projeto para ver esse objeto funcionando. Vamos criar o script no arquivo HTML mesmo, pois é somente uma demonstração didática.

Apostila página 141

> Vamos Praticar

Alguns dos métodos do objeto window são mostrados abaixo:

Método	Descrição
alert()	Mostra uma caixa de alerta contendo uma mensagem e um botão de OK.
confirm()	Mostra uma caixa de diálogo contendo uma mensagem com um botão de OK e um de CANCEL.
prompt()	Mostra uma caixa de diálogo para o usuário entrar com algum dado.
open()	Abre uma nova janela no navegador.
close()	Fecha a janela atual do navegador.
setTimeout()	Executa uma ação após um determinado tempo. Essa ação pode ser uma chamada de função, uma expressão de avaliação, etc.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Métodos do DOM



Como foi dito toda página possui um objeto **document** e ele é a **porta de acesso para todos os elementos da página web**. Desse modo, é possível pelo JavaScript utilizar os diversos métodos do DOM.

> Vamos Praticar

O método **createElement()** cria um novo nó/elemento com um nome específico. Esse nome pode ser com letras maiúsculas ou minúsculas.

Criar um elemento botão

Criar um elemento parágrafo

Criar um elemento imagem

```
document.createElement('BUTTON');
```

```
document.createElement('p');
```

```
document.createElement('img');
```

> Vamos Praticar

A propriedade **innerHTML** define ou retorna o conteúdo de um elemento **HTML**. E o método **appendChild()** adiciona o nó/elemento criado em um elemento pai específico.

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

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Para Aprender mais



Para Aprender mais



Procure sempre aprender e estudar mais. Seguem alguns links para você estudar e aprender mais: DOM:

https://pt.wikipedia.org/wiki/Modelo_de_Objeto_de_Documentos

<https://developer.mozilla.org/pt-BR/docs/Glossary/DOM>

https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model

https://www.w3schools.com/js/js_htmlDOM.asp

Objeto window:

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

Objeto document:

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

Método createElement():

<https://developer.mozilla.org/pt-BR/docs/Web/API/Document/createElement>

eElement Propriedade innerHTML:

<https://developer.mozilla.org/pt-BR/docs/Web/API/Element/innerHTML>

Método appendChild():

<https://developer.mozilla.org/pt-BR/docs/Web/API/Node/appendChild>

IOS – Instituto de
Oportunidade Social

Exercício



Exercício 1

Crie um Array unidimensional com 8 posições contendo os seguintes valores "17, 43, 8, 4, 97, 56, 29" em cada posição irá conter um valor do tipo number.

- utilizando o método `forEach` crie uma função que receba cada posição do array e verifique se o número é par ou ímpar e exiba o resultado no `console.log`
- Com o método `filter` crie uma verificação para retornar somente os valores menores ou iguais a 18.