



Módulo 2

Javascript

< thefutureisblue.me />



Foto de João da Silva

Objetivos da aula

Introdução ao
Javascript



O que é Javascript?

JavaScript (ou simplesmente JS) é uma linguagem de programação criada em 1995 com uma extensão do HTML.

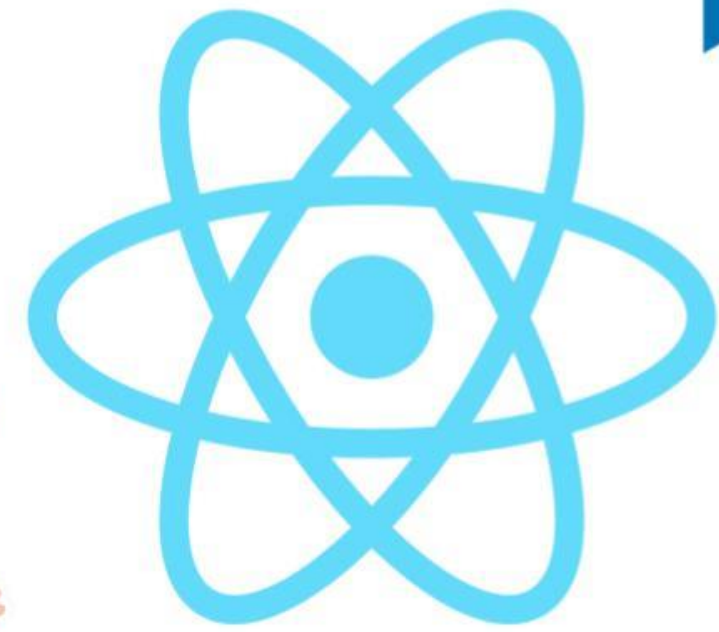
Desenvolvida por **Brendan Eich** a serviço da **Netscape**.

No início de Dezembro de 1995, o Java estava no seu ápice e a linguagem foi renomeada para JavaScript.

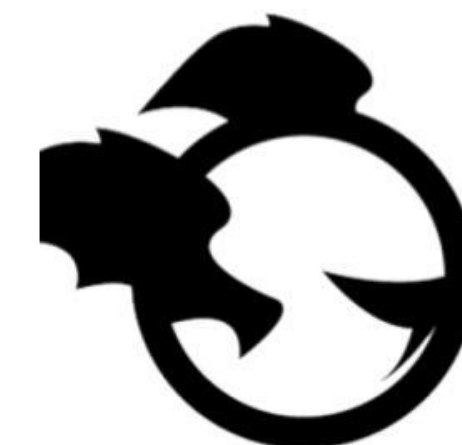
O que é Javascript?

- Em 1996, os criadores do Javascript se associaram ao ECMA (European Computer Manufacturers Association)
- JS permite a criação de páginas dinâmicas e interativas, executadas localmente.
- É uma linguagem suportada pela maioria dos browsers, como Google Chrome, Firefox, Edge, Safari, Opera, Brave.

O que é Javascript?



Omniscient



O que é Javascript?

O JavaScript, ao contrário de outras linguagens, é executada diretamente no navegador do usuário final.

Como é uma linguagem interpretada (e não compilada), todo o seu código pode ser visualizado pelo navegador.

JavaScript é a peça responsável
por responder às ações de
qualquer usuário em nossas
interfaces.

Sempre que pensamos em
ações (clique, rolar de página,
pressionar de teclas) estamos
falando de **JavaScript!**

Quais as formas de se trabalhar com JavaScript?

Elas se assemelham bastante às formas de trabalhar com CSS



Inline



Interno



Externo

Inline

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Exemplo Javascript</title>
  </head>
  <body>
    <button onclick="alert('oi!')">Clique Aqui!</button>
  </body>
</html>
```

Interno

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Exemplo Javascript</title>
  </head>
  <body>
    <button onclick="apresentarMensagem()">Clique Aqui!</button>

    <script type="text/javascript">
      function apresentarMensagem(){
        alert('oi');
      }
    </script>
  </body>
</html>
```



Externo

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Exemplo Javascript</title>
    <script type="text/javascript" src="meujavascript.js"> </script>
  </head>
  <body>
    <button onclick="apresentarMensagem()">Clique Aqui!</button>
  </body>
</html>
```

Exemplo.html

```
function apresentarMensagem(){
  alert('oi');
}
```

meujavascript.js



O que se pode fazer com o JavaScript?

- Realizar operações matemáticas.
- Interagir com o conteúdo do documento, alterando propriedades da página.
- Gerar documentos com aparência definida na hora da visualização, com base em informações do cliente.
- Interagir com o usuário por meio de eventos.

**O JavaScript nos provê várias
funções para que possamos
buscar elementos em nossa tela.**

**A seguir, vamos ver os principais
exemplos, com algumas
demonstrações.**

Localizar elementos

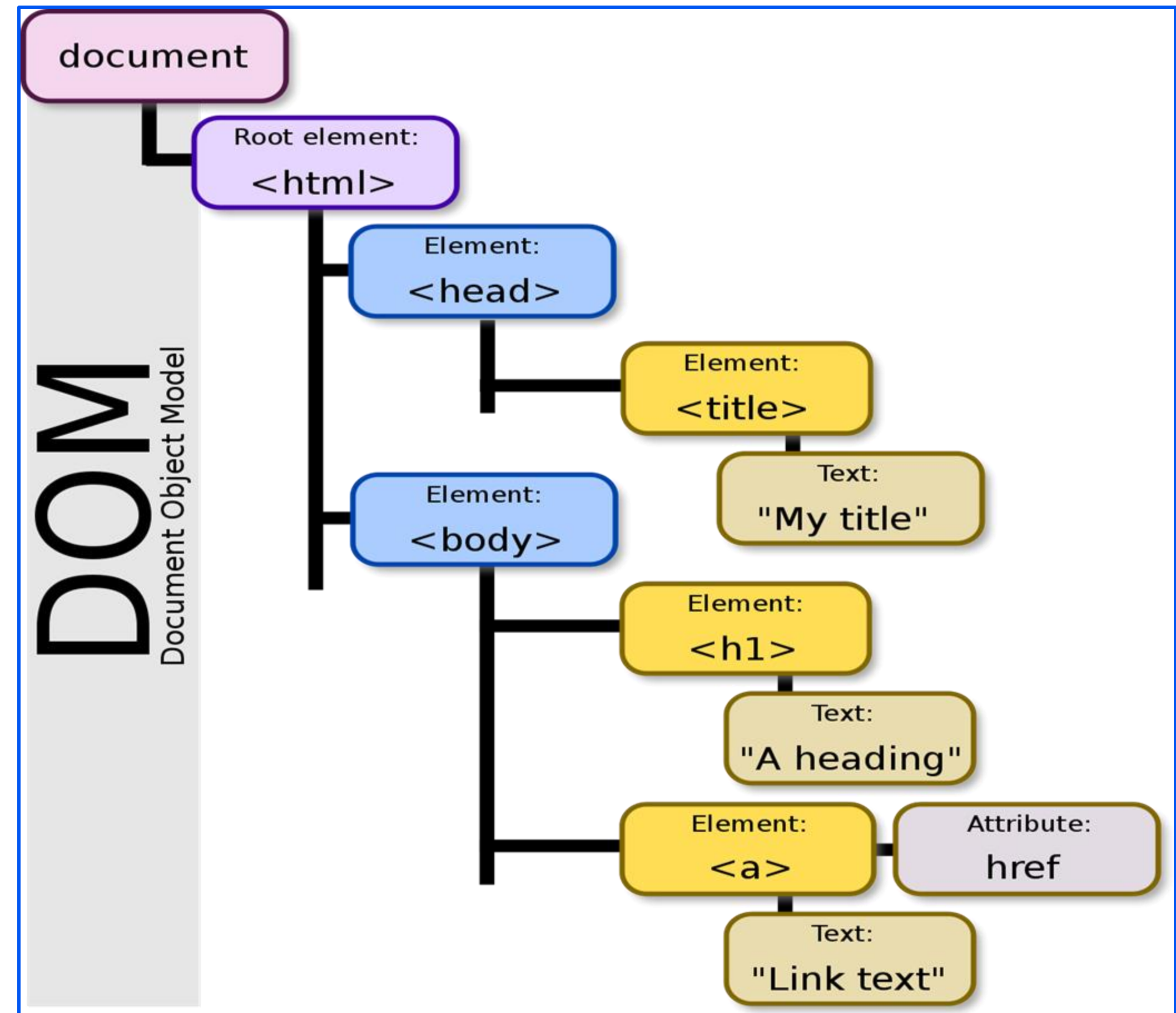
document.body	Retorna o elemento <body>
document.documentElement	Retorna o elemento <html>
document.domain	Retorna o nome de domínio do servidor do documento
document.forms	Retorna todos os elementos <form>
document.head	Retorna o elemento <head>
document.images	Retorna todas os elementos
document.links	Retorna todos os elementos que estejam com href
document.scripts	Retorna todos os elementos <script>
document.title	Retorna o elemento <title>
document.URL	Retorna a URL completa do documento



O que é DOM?

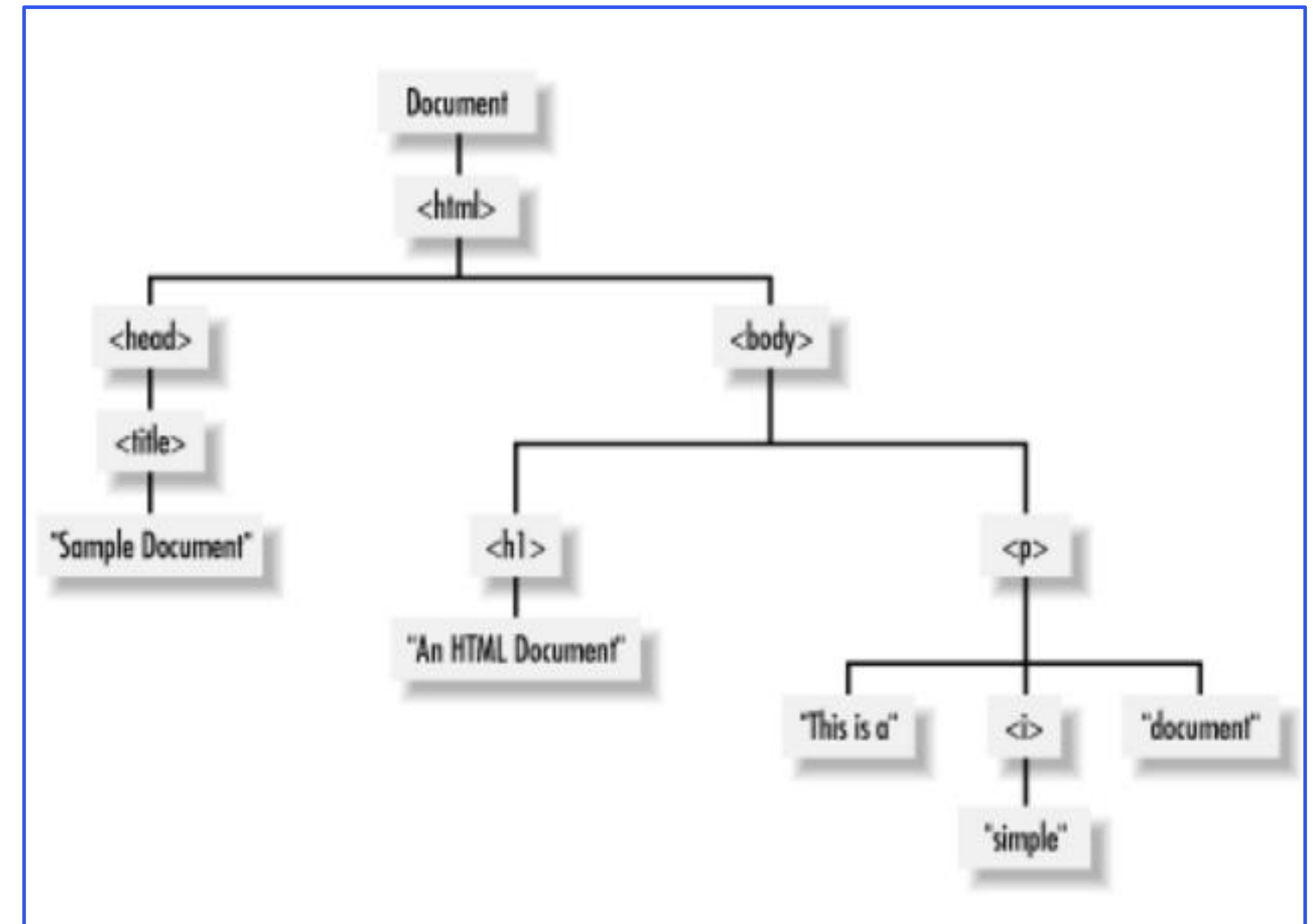
O **DOM** (*Document Object Model*) é uma interface de programação para documentos HTML, XML e SVG.

Ele fornece uma representação estruturada do documento como uma árvore e define métodos que permitem acesso para que possa ser alterada a estrutura, estilo e conteúdo do documento.



O que é DOM?

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Document</title>
  </head>
  <body>
    <h1>An HTML Docuemnt</h1>
    <p>This is a <i>simple</i> document</p>
  </body>
</html>
```



Estrutura HTML e o DOM

- Quando uma página web é requisitada, ela recebe um texto que é transformado em uma estrutura HTML;
- Por meio do JS podemos acessar cada um destes elementos do HTML através do DOM.



Localizar elementos por atributos

```
1  const exemplo1 = document.getElementById('exemplo1');
2  exemplo1.innerHTML = 'Eu represento o elemento <div id="exemplo1"></div>';
3
4  const exemplo2 = document.getElementsByClassName('classe-exemplo');
5
6  for (const elemento of exemplo2) {
7    elemento.innerHTML = 'Eu sou um elemento com a classe "classe-exemplo"';
8  }
9
```

É possível utilizar métodos para buscar elementos de acordo com seus atributos. Nesses casos podem ser retornados um ou mais elementos.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
```

✖ ▶ Uncaught TypeError: Cannot set property 'innerHTML' of null
at [Exemplo.html:7](#)

```
6 <script>
7     const principal = document.getElementById('principal');
8     principal.innerHTML = 'Hello!';
9 </script>
10 </head>
11 <body>
12     <div id="principal"></div>
13 </body>
14 </html>
```





```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Exemplo</title>
6      <script>
7          function iniciar() {
8              const principal = document.getElementById('principal');
9              principal.innerHTML = 'Hello, world!';
10         }
11     </script>
12 </head>
13 <body onload="iniciar()">
14     <div id="principal"></div>
15 </body>
16 </html>
```

Quando o navegador interpretou o JavaScript, ele tentou buscar um id "principal" que ainda não existia.

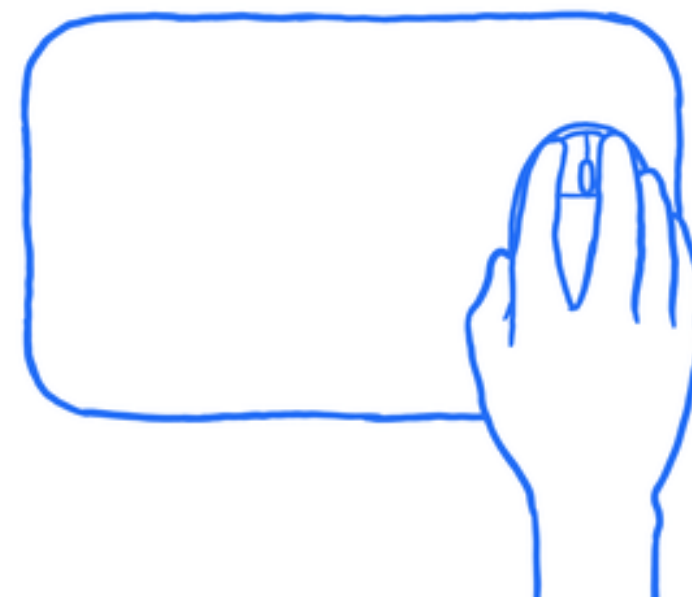
Adicionando e excluindo elementos

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Exemplo</title>
6      <script>
7        function iniciar() {
8          const novaDiv = document.createElement("div");
9          novaDiv.style = "border: 1px solid red;";
10
11          const novoTexto = document.createTextNode("Woow!");
12
13          novaDiv.appendChild(novoTexto);
14
15          const substituir = document.getElementById("substituir");
16          document.body.replaceChild(novaDiv, substituir);
17        }
18      </script>
19    </head>
20    <body onload="iniciar()">
21      <div id="substituir">Essa div será substituída</div>
22    </body>
23  </html>
```

É possível utilizar o JavaScript para adicionar, remover ou substituir um elemento HTML.

No exemplo ao lado, podemos ver a substituição de uma div por outra que foi criada pelo javascript.

Eventos, o que são, onde vivem e do que se alimentam?



Os eventos são basicamente um conjunto de ações que são realizadas em um determinado elemento da página web, seja ele um texto, uma imagem, ou uma div, por exemplo.



O navegador nos provê várias
funções para que possamos
utilizar **JavaScript** para
manipular os elementos em tela
e ***.addEventListener*** é uma delas.

Eventos

```
.addEventListener("<TIPO DE AÇÃO>", function(){}))
```

Esse método é utilizado para **registrar uma única espera de evento em um único alvo**. Mas nada impede que sejam criados vários eventos para um mesmo elemento.



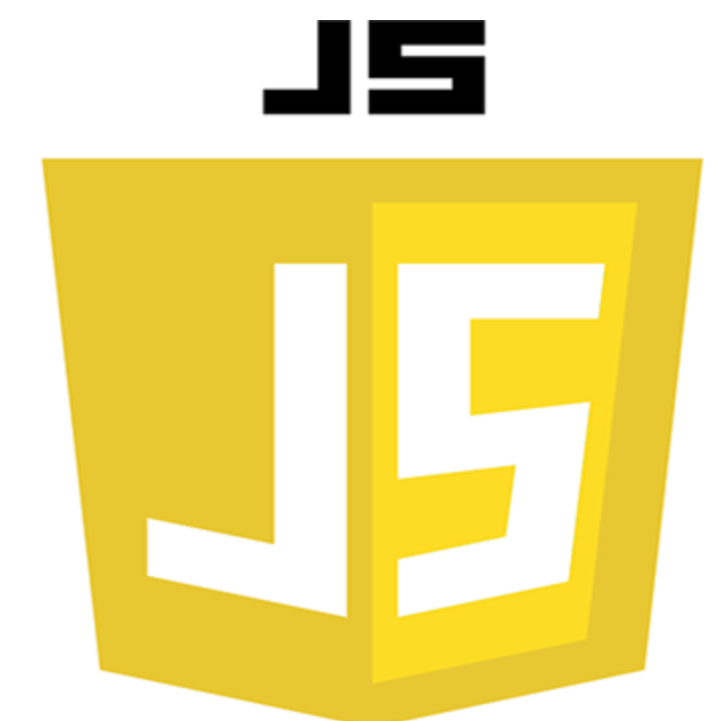
Eventos

```
1  const botao = document.querySelector("#botao");
2
3  function botaoFoiClicado() {
4      console.log("Clicou no botão.");
5  }
6
7  botao.addEventListener("click", botaoFoiClicado);
8
9  console.log("Evento adicionado ao botão. Clique para exibir uma mensagem.");
10
11  setTimeout(function () {
12      botao.removeEventListener("click", botaoFoiClicado);
13
14      console.log("5 segundos se passaram, botão não possui mais o evento.");
15  }, 5000);
```

- Da mesma forma que podemos adicionar eventos, podemos remover quando acharmos necessário;
- Para isso utilizamos o método *removeEventListener*, onde passamos o evento e a função que o evento está escutando.



onBlur	Ao remover o foco do elemento
onChange	Ao mudar o valor do elemento
onClick	Ao clicar no elemento
onFocus	Ao obter foco no elemento
onKeyPress	Quando o usuário pressiona uma tecla sobre o elemento
onLoad	Quando o elemento é carregado por completo
onMouseOver	Quando o usuário passa o mouse sobre o elemento
onMouseOut	Quando o usuário retira o mouse sobre o elemento
onSubmit	Define ação ao enviar um formulário

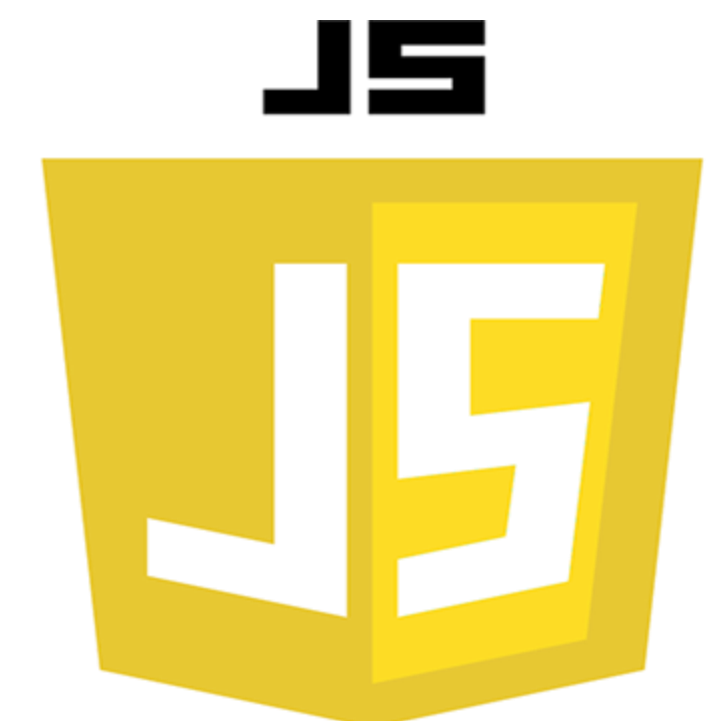


```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Exemplo</title>
6      <script>
7        let contador = 1;
8
9        function iniciar() {
10          const botao = document.body.firstChild;
11
12          botao.addEventListener("click", function () {
13            const novoElemento = document.createElement("p");
14            novoElemento.textContent = "Elemento " + contador;
15
16            document.body.appendChild(novoElemento);
17
18            contador++;
19          });
20        }
21      </script>
22    </head>
23    <body onload="iniciar()">
24      <div id="substituir">Essa div será substituída</div>
25    </body>
26  </html>

```

No exemplo ao lado, é criado um novo elemento `<p></p>` cada vez que for clicado no botão **Adicionar Elemento**.





**Botando pra
rodar**

Por hoje é só!
Obrigado! =)