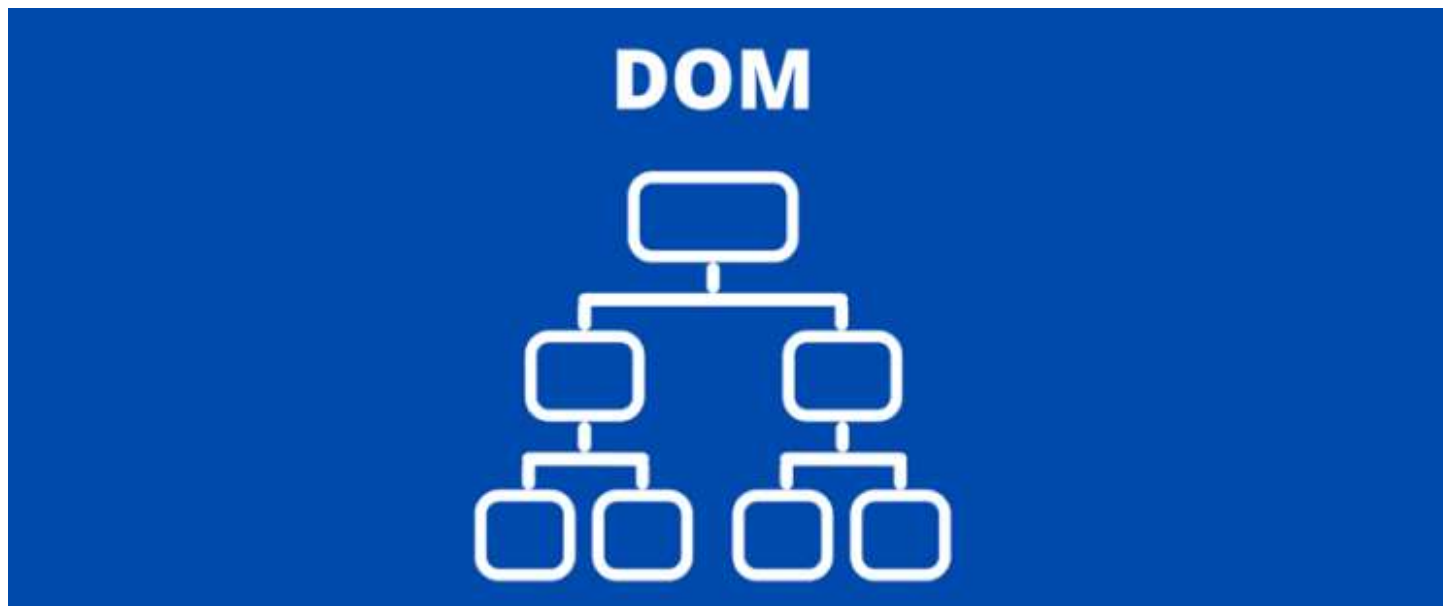


Artigos > **Front-end**

O que é o DOM?

**Ellen Pimentel**

12/08/2022

COMPARTILHE

A tradução do acrônimo inglês [DOM](#) (Document Object Model) significa Modelo de Documento do Objeto; trata-se da modelagem de todo o HTML.

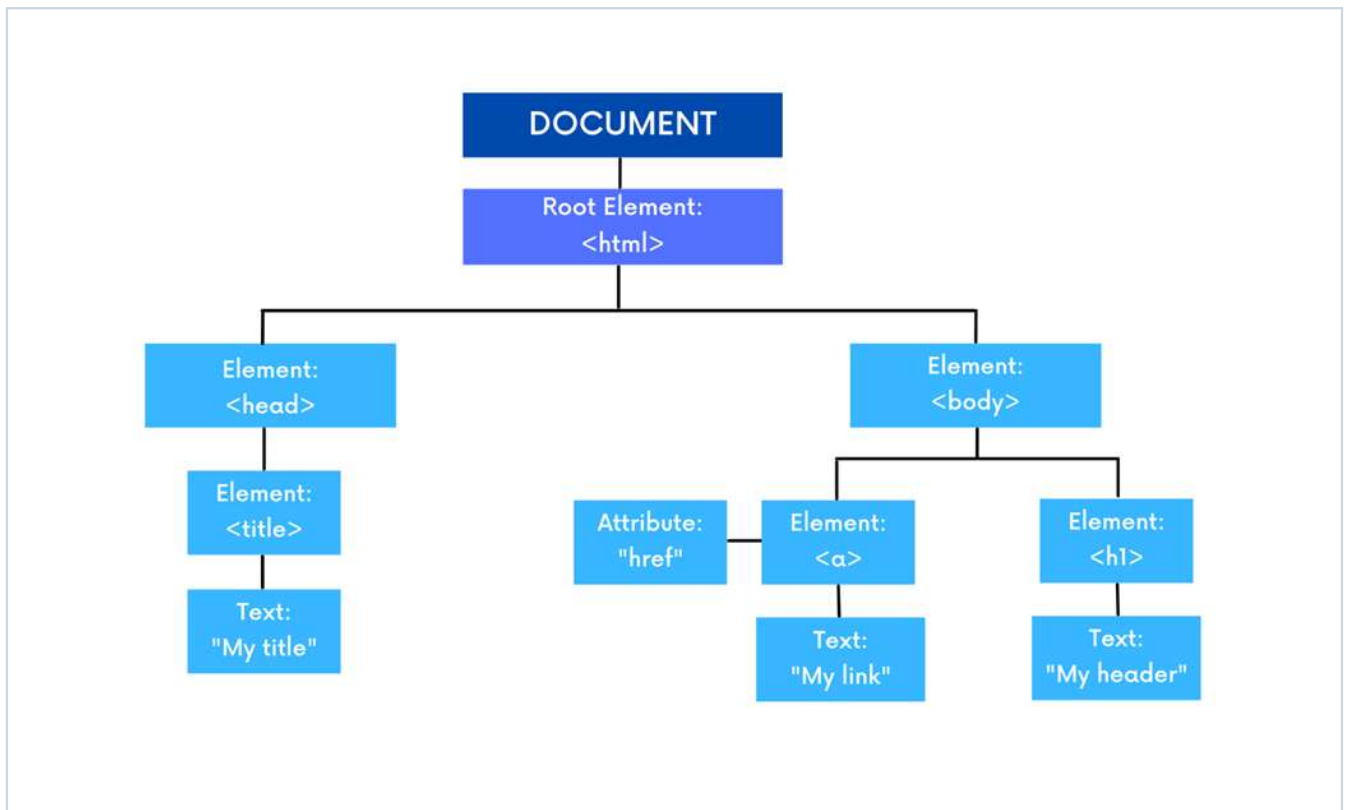
A partir da sua quinta versão, o HTML até então estático, além das melhorias na estrutura e separação da semântica da estilização, ganhou interatividade com a criação da linguagem de programação JavaScript, e é nesse momento que surge o DOM.

Uma das melhores formas para entender um conceito, e é algo recorrente no universo da tecnologia, é fazer abstrações e analogias.

Estrutura

O DOM é como uma árvore genealógica, porém, de forma invertida. O elemento que antecede o *document* é o *window*, que nada mais é que a janela do navegador. Em sua estrutura, o *document* está no topo como objeto global e tem como elemento raiz a *tag html* e todas as outras descendem dela através das suas ramificações (*branches*).

A *tag html*, objeto pai, apresenta dois objetos filhos: o **head** e o **body** (o cabeçalho e o corpo). Os objetos que seguem nas ramificações de baixo são denominados como *child*, e os de cima, *parent*. A *tag head* é *parent* da *tag title*, e a *body* é *parent* das *tags a* e *h1*, e assim sucessivamente, de acordo com a hierarquia. Das *tags*, derivam os atributos, e destes, seus valores.



Fonte: [Representação adaptada do W3Schools](#)

No código, ele ficaria traduzido dessa maneira:

```
<!DOCTYPE html>
<html>
<head>
  <title>My title</title>
</head>
<body>
  <a href="My link"></a>
  <h1>My header</h1>
</body>
</html>
```

Onde ele está inserido?

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.

Para realizar a comunicação entre eles é necessário inserir a *tag script* no arquivo HTML, e como boa prática, ela deve estar antes do fechamento da *tag body* para que os *scripts* sejam carregados após o código base.

É possível realizar de duas formas: escrevendo o código em JavaScript dentro da própria *tag script*, ou inserindo o caminho relativo do arquivo externo. Também como boa prática, a segunda opção é a mais recomendada para a separação de responsabilidades e melhor manutenção do código.

```
<script>
    alert("Olá, Mundo!")
</script>
```

```
<script src="script.js"></script>
```

Maneiras de manipulá-lo

No DevTools do navegador, na aba Console, ao digitar **console.dir** (diretório) e passar o parâmetro *document*, é exibida uma lista com propriedades e funcionalidades do documento.

```
> console.dir(document)
```

```
▼ #document ⓘ
  ▶ location: Location {ancestorOrigins: DOMStringList, href: 'chrome://new-tab-page/'
    URL: "chrome://new-tab-page/"
  ▶ activeElement: body
  ▶ adoptedStyleSheets: []
  ...
```

São várias as formas de navegação dentro do DOM, no JavaScript utilizamos o objeto *document* e através do ponto acessamos as propriedades e métodos, possibilitando selecionar, alterar, deletar, e criar elementos aos componentes do website, de acordo com a padronização criada pela [W3Schools](https://www.w3schools.com).

Para realizar essas ações temos alguns métodos, tais como:

- `document.getElementById()` ;
- `document.getElementsByClassName()` ;
- `document.getElementsByTagName()` ;
- `document.querySelector()` ;
- `document.querySelectorAll()` ;
- `document.createElement()` ;
- `element.addEventListener()` ;

Com o `document.querySelector` , por exemplo, dentre as opções que ele oferece, podemos alterar o texto no documento HTML:

```
<body>
  <h1> Olá! </h1>
  <script src="script.js"></script>
</body>
```

```
document.querySelector("h1").innerText = "Olá, Mundo!"
```



Lembra que a janela do navegador é representada pelo objeto *window*? Utilizando outros exemplos, quando escrevemos a função `alert` ou o método `write` , é como se escrevêssemos dessa forma:

```
<script>
  window.alert("Olá, Mundo!")
  window.document.write("Olá, Mundo!")
</script>
```

O *window* pode ser omitido, ele não é obrigatório no JS, assim como acontece com o ponto-e-vírgula (;) ao final de um comando.

```
<script>
  alert("Olá, Mundo!")
  document.write("Olá, Mundo!")
</script>
```

Conclusão

O DOM é um conjunto de objetos e sua estrutura de dados é representada através de um diagrama, ou de forma figurativa, uma árvore de objetos.

Quando se fala do desenvolvimento *web* básico, temos a tríade inseparável: HTML, CSS e JavaScript, cada um com o seu [papel](#). Conhecer a [base](#) da programação e o que eles representam é essencial para poder avançar com *frameworks* e bibliotecas como Angular, Vue.js e React.

A tecnologia é uma área dinâmica, e assim como em diversos aspectos da nossa vida, não é necessário ter um dom para realizar algo, mas sim, estudo contínuo e muita prática.

Confira neste artigo:

- [Estrutura](#)
- [Onde ele está inserido?](#)
- [Maneiras de manipulá-lo](#)
- [Conclusão](#)