



JAVASCRIPT.



MATERIAL DE APOIO

Disciplina: JavaScript

Nome da aula: Primeiros Passos.

Professor(a): Marcelo Estruc

Introdução

Entre as linguagens de programação mais populares e amplamente utilizadas, o JavaScript é um dos principais candidatos. O JavaScript é usado como linguagem de programação do lado do cliente por 95,2% de todos os sites do mundo. É uma linguagem de programação interpretada leve e de alto nível. É uma escolha frequentemente recomendada pelos desenvolvedores para a construção de aplicativos para várias necessidades, desde sites simples até jogos interativos. Nos resultados da pesquisa anual do desenvolvedor de 2018 do Stack Overflow, foi apontado que 71,5% dos desenvolvedores profissionais marcaram o JS como sua escolha de idioma preferido.

Objetivos da aula

- Discutir a construção de soluções Javascript na prática.

Resumo

Nesta aula, você aprendeu duas maneiras diferentes de “pegar” elementos do seu site para que você possa manipular. Você aprendeu o `getElementById` e aprendeu a `getElementsByTagName`. Esses são geralmente os melhores métodos para “pegar” os elementos, mas eles têm limitações. O primeiro, `getElementById`, dá acesso apenas aos elementos que foram atribuídos a um id. O segundo, `getElementsByTagName`, é bom para mudanças de vários elementos ao mesmo tempo, mas é um pouco complicado para um bom trabalho mais detalhado.

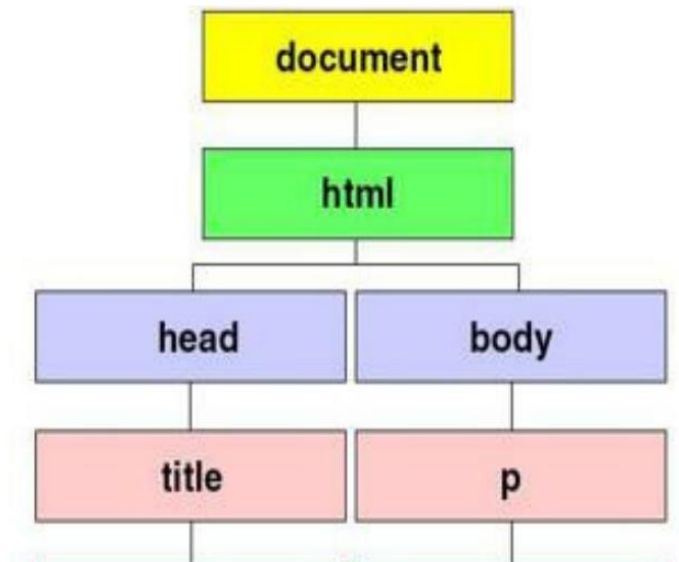
Felizmente, essas duas opções são apenas dois dos muitos métodos para trabalhar com o Modelo de Objeto de Documento, o DOM. O DOM é um organograma criado automaticamente pelo navegador quando sua página web é carregada em um navegador. Todos os elementos do seu site - as tags, os blocos de texto, as imagens, os links, as tabelas, o estilo, atributos e muito mais - estão neste organograma. Isso significa que seu código JavaScript pode pegar qualquer elemento do seu site. Além disso, o JavaScript pode adicionar, mover ou excluir elementos. Se você quisesse, você poderia quase criar um site do zero usando os métodos DOM do JavaScript.

Aqui está um documento html simplificado. Os três níveis superiores do DOM são sempre os mesmos para uma página web padrão. O documento é o primeiro nível. Abaixo do document está o segundo nível, o html. E abaixo do html está o terceiro nível, que é composto pela head e body. Abaixo de cada um deles há mais níveis. Veja o código abaixo:

```
<html>
  <head>
    <title>
      Meu título
    </title>
  </head>
  <body>
    <p>
      Texto na página
    </p>
  </body>
</html>
```

Fonte: Autoral

Aqui está a mesma coisa, mostrada como um organograma.



Fonte: Autoral

Encontrando Elementos

Como você aprendeu nessa aula, o Document Object Model (DOM) é uma hierarquia de pais e filhos. Tudo no site, excluindo o próprio document, é filho de algo. Então, agora, vamos falar sobre como seu código JavaScript pode usar essa hierarquia de pais e filhos para ler ou alterar praticamente qualquer coisa em um site. Podemos “pegar” qualquer elemento do nosso site e manipular. Abaixo um exemplo de como podemos pegar um elemento pelo id.

```
var email = document.getElementById("email");
```

Você também aprendeu como fazer uma coleção de array de todos os elementos de um determinado tipo dentro do documento...

```
var elementoP = document.getElementsByTagName("p");
```

Tendo feito um array de parágrafos, você pode separar qualquer parágrafo dentro do array para que você possa, por exemplo, pegar seu conteúdo.

```
var conteudo = elementoP[2].innerHTML;
```

A instrução acima atribui a variável conteúdo o texto do terceiro parágrafo do documento.

Um outro exemplo é, se quisermos restringir a pegar o segundo elemento de uma lista, poderíamos fazer assim:

```
var variasLi = document.getElementsByTagName("li")
var segundoltem = variasLi[0]
```

Dessa forma, conseguimos pegar o segundo elemento da lista para poder manipular. Depois que temos o elemento, podemos fazer qualquer coisa com ele. Por exemplo, trocar o valor dele ou trocar a cor dele.

Referência Bibliográfica

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6ª Ed. Porto Alegre: Bookman, 2013.

FREEMAN, Eric. **Use a cabeça!**: programação JavaScript. 1ª Ed. São Paulo: Alta Books, 2016.

Exercícios

1) Se quisermos pegar um elemento pelo id para depois podermos manipular, qual instrução em Javascript você usaria?

- a) document.querySelector()
- b) document.getElementById()
- c) document.getElementsByName()
- d) document.getElementsByClassName()
- e) document.getElementsByTagName()

2) Se quisermos pegar uma coleção de elementos pelo name para depois podermos manipular, qual instrução em Javascript você usaria?

- a) `document.querySelector()`
- b) `document.getElementById()`
- c) `document.getElementsByName()`
- d) `document.getElementsByClassName()`
- e) `document.getElementsByTagName()`

3) Se quisermos pegar uma coleção de elementos pela classe para depois podermos manipular, qual instrução em Javascript você usaria?

- a) `document.querySelector()`
- b) `document.getElementById()`
- c) `document.getElementsByName()`
- d) `document.getElementsByClassName()`
- e) `document.getElementsByTagName()`

4) Se quisermos pegar um elemento que tenha o `id="novo"` e `class="secao"` para depois podermos manipular, qual instrução em Javascript você usaria?

- a) `document.querySelector()`
- b) `document.getElementById()`
- c) `document.getElementsByName()`
- d) `document.getElementsByClassName()`
- e) `document.getElementsByTagName()`

Gabarito

1. A **Letra B** é a alternativa correta, pois o JavaScript utiliza da instrução `getElementById` para pegar um elemento pelo ID.
2. A **Letra C** é a alternativa correta, pois o JavaScript utiliza da instrução `getElementsByName()` para pegar uma coleção de elementos pelo name.
3. A **Letra D** é a alternativa correta, pois o JavaScript utiliza da instrução `getElementsByClassName()` para pegar uma coleção de elementos pela classe.
4. A **letra A** é a alternativa correta, pois o JavaScript utiliza da instrução `querySelector()` para pegar um elemento que tenha o `id="novo"` e `class="secao"`.