



## JavaScript – Manipulando o DOM

### Exercícios de Fixação

01. Com a página HTML abaixo, implemente-a em sua IDE e execute-a no seu navegador.

```
<html >
<head>
  <script language="javascript">
    function testandoArvoreDOM() {
      var element = document.documentElement.lastChild;
      alert("Eu sou o nodo" + element.nodeName);

      var outroElemento = document.getElementsByTagName('h1')[0];
      alert("Eu sou o nodo " + outroElemento.nodeValue);

      var filho = outroElemento.firstChild;
      alert("Eu sou o nodo " + filho.nodeValue);
    }
  </script>
  <title>
    Teste de Árvore DOM
  </title>
</head>
<body>
  <h1>Teste de Título</h1>
  <p>
    Você conhece sobre árvores DOM? Se não conhece aprenda em
    <a href="http://www.javascript.com"> www.javascript.com </a>
    e seja o mais novo entendido sobre o assunto.
  </p>
  <span onclick="testandoArvoreDOM();">Clique aqui para Testar o Acesso a Árvore DOM</span>
</body>
</html>
```

Ao clicar no botão da tela, você verá o resultado da execução, portanto responda:

- A. Qual porção do HTML 'element' está apontando?
- B. Porque há um resultado 'null' aparecendo? Onde está ocorrendo isso e porquê?
- C. A variável "filho" está referenciando qual elemento?

**02.** Usando a página HTML abaixo, faça códigos javascript para:

1. Exiba apenas o texto da tag com id=paragrafo (sem tags).
2. Exiba em tela o conteúdo dentro das tags <span>.
3. obtenha o elemento li com o texto 666 dentro dele e exiba esse valor
4. Crie uma função chamada exibirInnerHTML que receberá 1 parametro (um elemento) e dentro da função escreverá o innerHTML no console. Repita essa ação para as tags <p>, <strong>, e <li> com id=item.

Use botões para sua execução.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <p id="paragrafo">123 <span>teste</span> 123 <span>Lorem <strong>negrilo</strong></span> 123</p>
    <ul>
      <li>111</li>
      <li>222</li>
      <li id="item">333</li>
      <li>444</li>
      <li>555</li>
      <li>666</li>
      <li>777</li>
      <li>888</li>
    </ul>
  </body>
</html>
```

**03.** Crie uma aplicação WEB na qual o visitante insere faz um login, ele deve então inserir um texto para autenticar o usuário e outro para senha. Se o usuário digitado for igual “*sergio*” e a senha for igual a “*lenda*”, mostre uma mensagem em um parágrafo com o texto “*Seja Bem vindo*” na cor verde, caso contrário, mostra e mensagem “*Login incorreto*” na cor vermelha. **Dica:** Você pode usar operadores de conjunção &&.

**04.** Crie uma página para uma loja virtual. A página deve conter a descrição do produto “*Boneco do Rodrigo*” que custa R\$ 75,00. O usuário deve então informar em quantas parcelas ele deseja pagar e a página web deve calcular o valor de cada parcela.

**05.** Crie uma aplicação web para um boletim escolar aonde o professor deve digitar o nome do aluno e digitar 3 notas de 0 a 10 para ele. A aplicação web deve tirar a média das menções e exibir uma mensagem de acordo com os seguintes critérios;

- a) Média igual ou maior que 7; exibir a mensagem em verde de “APROVADO”
- b) Média entre 5 e 6,9; exibir a mensagem em amarelo de “RECUPERAÇÃO”
- c) Média menor que 5; exibir a mensagem em amarelo de “REPROVADO PELO RODRIGO”

**06.** Seja o seguinte código da página abaixo:

```
<body>
  <p>
    No corpo desta página há um parágrafo
    e uma lista não-ordenada.
  </p>
  <ul>
    <li> Primeiro item da lista</li>
    <li> Segundo item da lista</li>
    <li> Terceiro item da lista</li>
  </ul>

  <script type="text/javascript" src="ex2.js">
  </script>
</body>
```

Coloque um botão abaixo da lista chamado a função Listar() que será implementada no arquivo ex2.js. Esta função deverá mostrar, após o botão, a quantidade de itens que possui a lista e escrever os textos de cada item, um embaixo do outro. Use a função do Javascript `getElementsByTagName()` que retorna um vetor com todos os elementos de uma determinada tag, no caso `<li>`. Basta depois manipular esse vetor como um array javascript normal, lembrando que cada um de seus elementos será um `<li>`.

**07.** Inclua uma outra lista no exemplo anterior e faça um script para obter a referência aos itens apenas dessa segunda lista que você adicionou.

*Dica: Para obter os elementos da segunda lista, você deve ler o “segundo” elemento da lista de itens retornados por `document.getElementsByTagName()`.*

**08.** No exercício anterior, coloque um outro botão e um campo de input de texto. Faça uma função que será relacionada ao click do botão quem quando o usuário digitar o texto, ele incluirá esse texto como um novo item da lista. A função deve verificar se de fato houve digitação de texto, caso contrário, mostrará uma mensagem de texto informando do erro ao usuário, logo abaixo do campo de entrada. Quando houver a inclusão, a função garantirá que o cursor estará novamente no campo de entrada e qualquer mensagem seja apagada.

**09.** Modifique os estilos dos campos acima. Quando o usuário entrar no campo de digitação, este deve mudar a cor de fundo. Use uma cor amarelo ou azul bem clara. Quando sair do campo, volta ao fundo branco padrão.

**10.** Usando a mesma página do exercício 04, coloque outro campo de entrada onde o usuário digitará qual posição ele deseja colocar um item na lista. A partir dessa posição, modifique a função no exercício anterior para ele inserir naquela posição indicada. Caso o usuário não tenha digitado nenhuma posição, insira ao final. Caso a posição que ele digitar não exista, deve ser mostrada uma mensagem a ele informando da impossibilidade e pedindo se ele deseja incluir ao final (use o método `confirm()` para isso). Em caso positivo, faça a inclusão.

**11.** Tabela “zebrada” é uma técnica que permite que as linhas de uma tabela apareçam em cores diferentes, alterando pelas linhas pares e ímpares. Exemplo:

100	William	Gietz	515.123.8181
101	Sтивен	King	515.123.4567
102	Neena	Kochhar	515.123.4569
103	Lex	De Haan	515.123.4569
104	Alexander	Hunold	590.423.4567
105	Bruce	Ernst	590.423.4568
106	David	Austin	590.423.4569
107	Valli	Pataballa	590.423.4569
108	Diana	Lorenz	590.423.5567
109	Nancy	Greenberg	515.124.4569
110	Daniel	Faviet	515.124.4169
111	John	Chen	515.124.4269
112	Ismail	Sclama	515.124.4369
113	Jose Manuel	Urman	515.124.4469
114	Luís	Popp	515.124.4567
115	Den	Raphaely	515.127.4561
116	Alexander	Khao	515.127.4562
117	Shelley	Baida	515.127.4563
118	Sigal	Tobias	515.127.4564
119	Guy	Himuro	515.127.4565
120	Karen	Colmenares	515.127.4566
121	Matthew	Weiss	650.123.1234

Usando a página HTML abaixo, onde contém uma tabela, faça um código Javascript para deixá-la zebraada, ao apertar de um botão de execução. Seu código deve selecionar todos os `<tr>` que estão dentro de `<tbody>`, percorre-los e incluir a classe `‘zebraon’` naqueles que ocupam posições pares (ou ímpares a seu gosto). Utilize o método `querySelectorAll()`. PESQUISE SOBRE ELE!

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8" />
<title>Zebra</title>
<style>
    .zebraon{background: silver}
</style>
</head>

<body>
```

```
<table class="zebra">
  <thead><tr>
    <th>Vendedor</th> <th>Total</th>
  </tr></thead>
  <tbody><tr>
    <td>Manoel</td> <td>12. 300, 00</td>
  </tr><tr>
    <td>Joaquim</td> <td>21. 300, 00</td>
  </tr><tr>
    <td>Maria</td> <td>13. 200, 00</td>
  </tr><tr>
    <td>Marta</td> <td>32. 100, 00</td>
  </tr><tr>
    <td>Antonio</td> <td>23. 100, 00</td>
  </tr><tr>
    <td>Pedro</td> <td>31. 200, 00</td>
  </tr></tbody>
</table>
</body>
</html>
```