

## FUND\_WEB8 - Texto de apoio

Site: [EAD Mackenzie](#)  
Tema: FUNDAMENTOS DE WEB {TURMA 01A} 2022/1  
Livro: FUND\_WEB8 - Texto de apoio

Impresso por: IRART DA SILVA SANTOS .  
Data: quinta, 19 mai 2022, 15:32

# Índice

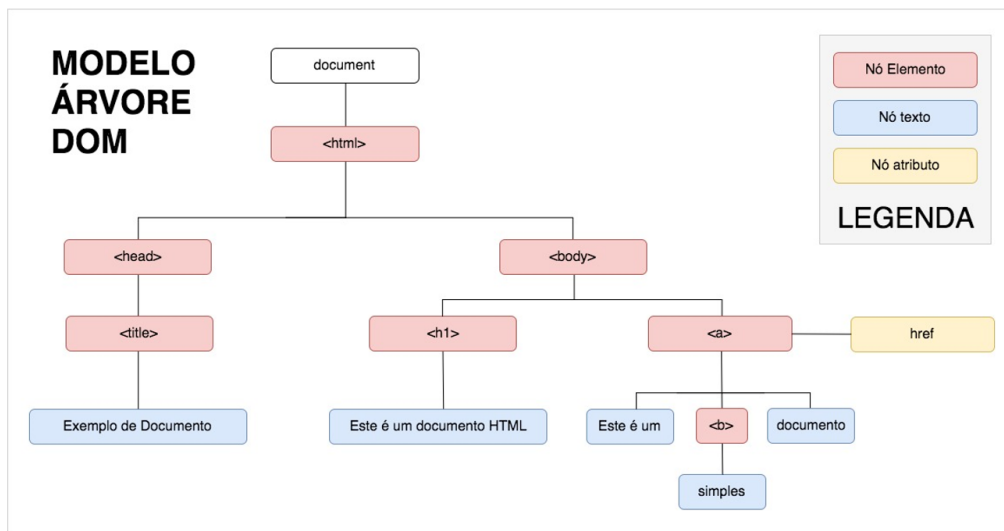
1. JAVASCRIPT – DOM E EVENTOS
2. DOM – EXEMPLOS DE ACESSO AOS ELEMENTOS HTML
3. DOM – EXEMPLOS DE ALTERAÇÃO
4. DOM – EXEMPLO DE CRIAÇÃO DE UM ELEMENTO
5. DOM – EXEMPLO DE DELEÇÃO DE UM ELEMENTO
6. EVENTO
- 6.1. Métodos usados com elementos do formulário:
7. VALIDAÇÃO
8. REFERÊNCIAS BIBLIOGRÁFICAS

# 1. JAVASCRIPT – DOM E EVENTOS

DOM (Document Object Model)

- É uma interface que representa como os documentos HTML e XML são lidos pelo seu browser.
- É chamado modelo de objeto porque o modelo (árvore DOM) é feito de objetos.
- Cada objeto representa uma parte diferente da página carregada na janela do navegador.
- DOM é criado pelo navegador quando uma página web é carregada.
- Consiste em quatro tipos de nós.

Imagem da árvore DOM



fonte: <https://cahfelix.com/wp-content/uploads/2017/08/dom-no-2.jpg>

1. Nó DOCUMENT – representa toda a página. Cada elemento, atributo e fragmento de texto na HTML é representado por um nó DOM próprio. É o ponto de partida para todas as visitas à árvore DOM.
2. Nó DE ELEMENTO – são todas as tags que estão em arquivos HTML ou XML e se transformam em elementos da árvore DOM.
3. O acesso à árvore começa procurando os elementos.
4. Nós DE ATRIBUTO – é a junção de todos os atributos para um nó específico. Há métodos e propriedades Javascript específicos para ler ou modificar os atributos desse elemento.
5. Nós DE TEXTO – é o texto que vai entre os elementos. Todo o conteúdo das tags. Nó de texto não podem ter filhos.

- As relações entre o document e todos os nós do elemento são descritas usando: pais, filhos, irmãos. Cada nó é um descendente do nó document.
- Cada nó é um objeto com métodos e propriedades.
- Scripts acessam e atualizam essa árvore DOM (não o arquivo fonte HTML).



Fonte: <https://www.todoespacoonline.com/w/wp-content/uploads/2014/05/Diagrama1.png>

Em outras palavras, O HTML DOM é um padrão para como obter, acessar, alterar, adicionar ou excluir elementos HTML.

- O Modelo de Objetos do Documento (DOM) foi criado pelo W3C;
- É uma multiplataforma que representa como as marcações em HTML, XHTML e XML são organizadas e lidas pelo navegador que você usa.
- É a interface entre a linguagem Javascript e os objetos do HTML.

Portanto, qualquer coisa criada pelo navegador Web no modelo da página Web poderá ser acessada por meio do objeto Javascript document.

**Cada elemento pode ter MÉTODOS (ações que podemos executar com eles) e PROPRIEDADES (valores que podemos buscar ou alterar os mesmos).**

**document.metodo.propriedade = "valor";**

**Como encontrar(acessar) elementos HTML:**

método	descrição
document.getElementById( <i>id</i> )	Encontra um elem por id
document.getElementsByTagName(" <i>nomeDaTag</i> ")	Encontra elem por nome tag
document.getElementsByClassName("classe")	Encontra elem por nome classe
querySelector( <i>cssSelector</i> )	Retorna apenas o primeiro elemento que corresponde aos seletores atribuídos
querySelectorAll( <i>cssSelector</i> )	Retorna todos os elementos no documento que correspondem a um seletor(es) CSS especificado

Fonte: Elaborado pela autora.

**Como alterar elementos HTML:**

Método	descrição
element.innerHTML = new html content	Altera conteúdo do innerHTML
element.attribute = new value	Muda o vlr atributo de um elem
element.setAttribute(attribute, value)	Muda o vlr atributo de um elem

element.style.property = new style	Muda o estilo de um elem
------------------------------------	--------------------------

**Fonte: Elaborado pela autora.**

#### **Adicionando e excluindo Elementos**

Método	descrição
document.createElement(element)	Cria um elem HTML
document.removeChild(element)	Remove um elem HTML
document.appendChild(element)	Adiciona um elem HTML
document.replaceChild(element)	Troca um elem HTML

**Fonte: Elaborado pela autora.**

## 2. DOM – EXEMPLOS DE ACESSO AOS ELEMENTOS HTML

### DOM\_acesso.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
<p id="demo">Use the DOM para acessar elementos.</p>
<script>
document.getElementById("demo").innerHTML = "DOM - acesso elem.";
</script>
</body></html>
```

Resultado no navegador: DOM\_acesso.html

### DOM - acesso elem.

### DOM\_seleciona.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
<p>Uso DOM para selecionar elem.</p>
<script>
document.getElementsByTagName("p")[0].innerHTML = "DOM selecionando elem";
</script>
</body></html>
```

Resultado no navegador: DOM\_seleciona.html

### DOM selecionando elem

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <p class="teste">Uso DOM para selecionar elem pela classe.</p>
  <script>
    document.getElementsByClassName("teste")[0].innerHTML = "DOM selecionando elem pela classe.";
  </script>
</body></html>
```

Resultado no navegador: DOM\_seleciona.html

### DOM\_seleciona\_classe.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <p class="teste">Uso DOM para selecionar elem pela classe.</p>
  <script>
    document.getElementsByClassName("teste")[0].innerHTML = "DOM selecionando elem pela classe.";
  </script>
</body></html>
```

Resultado no navegador: DOM\_seleciona\_classe.html

## **DOM selecionando elem pela classe.**

### **DOM\_seletor.html**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <p>Uso DOM com seletores.</p>
  <script>
    document.querySelectorAll("p")[0].innerHTML = "DOM selecionando elem com seletores.";
  </script>
</body></html>
```

Resultado no navegador: DOM\_seletor.html

## **DOM selecionando elem com seletores.**

### 3. DOM – EXEMPLOS DE ALTERAÇÃO

Como alterar o fluxo de saída HTML?

- Alterando o conteúdo HTML

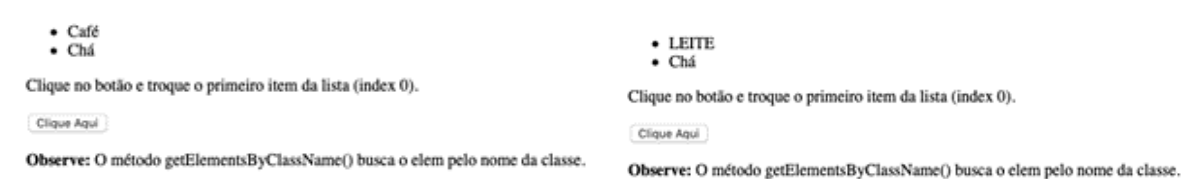
A maneira mais fácil de modificar o conteúdo de um elemento HTML é usando a propriedade `innerHTML`.

`document.getElementById(id).innerHTML = new HTML`

#### Dom\_altera\_um\_elem.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
<ul class="example">
  <li class="child">Café</li>
  <li class="child">Chá</li>
</ul>
<p>Clique no botão e troque o primeiro item da lista (index 0).</p>
<button onclick="trocaltem()">Clique Aqui</button>
<p><strong>Observe:</strong> O método getElementsByClassName() busca o elem pelo nome da classe.</p>
<script>
function trocaltem() {
  var list = document.getElementsByClassName("example")[0];
  list.getElementsByClassName("child")[0].innerHTML = "LEITE";
}
</script>
</body></html>
```

Resultado no navegador: Dom\_altera\_um\_elem.html



- Alterando o valor de um atributo

Para alterar o valor de um atributo HTML, use esta sintaxe:

`document.getElementById(id).attribute = new value`

#### DOM\_altera\_atributo.html

```
<!DOCTYPE html>
<html>
<body>

<script>
  document.getElementById("image").src = "coracao2.jpg";
</script>
</body></html>
```



Resultado no navegador: DOM\_altera\_atributo.html



- Para alterar o estilo de um elemento HTML

sintaxe:

```
document.getElementById(id).style.property = new style
```

### DOM\_altera.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <h1>DOM </h1>
  <button type="button" onclick="document.body.style.backgroundColor = 'red';>Alterar cor fundo pág</button>
</body></html>
```

Resultado no navegador: DOM\_altera.html

**DOM**

Alterar cor fundo pág

**DOM**

Alterar cor fundo pág

## 4. DOM – EXEMPLO DE CRIAÇÃO DE UM ELEMENTO

### DOM\_create.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Teste do DOM</title>
    <meta charset="utf-8">
    <style>
      .teste{background:orange}
    </style>
  </head>
  <body>
    <div id="id1" style="background:yellow; height:150px;width:450px;text-align:center">
      <div><br><br>

      <button>OI PESSOAL</button>

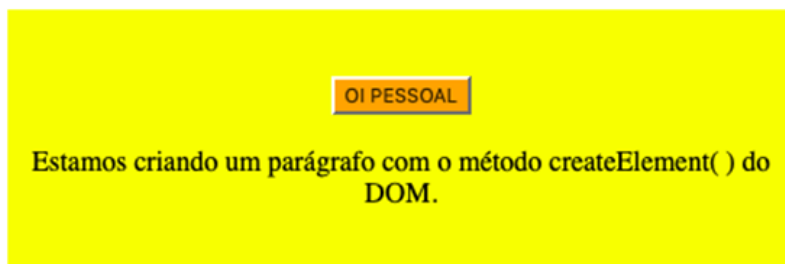
      <script>
        var criacao = document.createElement("p");
        criacao.innerHTML = "Estamos criando um parágrafo com o método createElement( ) do DOM.";

        document.getElementById("id1").appendChild(criacao);

        /*o createElement é armazenado em um var, ele só adicionado à árvore com o appendChild e o appendChild especifica qual
        elemento vai adicionar esse nó, exemplo no parágrafo da var criacao */

        var atributo = document.querySelector("button");
        atributo.setAttribute("class", "teste");
      </script>
    </div>
  </body>
</html>
```

Resultado no navegador: DOM\_create.html



## 5. DOM – EXEMPLO DE DELEÇÃO DE UM ELEMENTO

### DOM\_remove.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <ul>
    <li>Café</li>
    <li>Chá</li>
    <li>Leite</li>
    <li>Chocolate</li>
  </ul>
  <script>
    // Obtendo referências
    var lista = document.getElementsByTagName('ul')[0];
    var itens = lista.getElementsByTagName('li');
    // Removendo determinado elemento - Leite
    lista.removeChild(itens[2])
  </script>
</body></html>
```

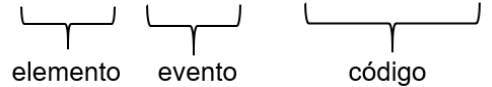
Resultado no navegador: DOM\_remove.html

- Café
- Chá
- Chocolate

## 6. EVENTO

- É a maneira como o navegador indica quando algo aconteceu.
- Quando ocorre um evento em um elemento, ele pode disparar uma função Javascript (como quando um usuário clica em um botão).

**elemento.onevento = nomeDaFunção;**



elemento      evento      código

elemento: alvo do nó de elemento DOM.

evento: precedido pela palavra on.

código: nome da função a ser chamada.

### Interface do usuário

load – dispara quando a página web foi carregada;  
unload – dispara quando a página web está sendo descarregada;  
error – dispara quando o navegador encontra um erro;  
resize – dispara quando a janela do navegador foi redimensionada;  
scroll – dispara quando o usuário rola a página para cima ou para baixo.

### FOCUS e BLUR

focus – quando um elem ganha o foco;  
blur – quando um elem perde o foco;  
focusin – o mesmo que focus;  
focusout – o mesmo que blur.

### De MOUSE

click – dispara quando o usuário clica no botão do mouse principal;  
dblclick – dispara quando o usuário clica no botão principal do mouse duas vezes em rápida sucessão;  
mousedown – dispara quando o usuário pressiona qualquer botão do mouse;  
mouseup – dispara quando o usuário solta o botão do mouse;  
mouseover – dispara quando o cursor estava fora um elem e é, então, movido para cima dele;  
mouseout – dispara quando o cursor está sobre um elem e, então, move-se para um outro elem.  
mousemove – dispara quando o cursor é movido sobre um elem.

### TECLADO

input – dispara quando o valor de um elem <input> ou <textarea> muda;  
keydown – dispara quando o usuário pressiona uma tecla no teclado;  
keypress – dispara quando o usuário pressiona uma tecla que resultaria na exibição de um caractere na tela;  
keyup – dispara quando o usuário solta a tecla.

### FORMULÁRIO

submit – quando um formulário é enviado, o submit é acionado no nó que representa o elem <form>;  
change – dispara quando o status de vários elem de um formulário muda;  
input – o evento de entrada.

Adicionando manipuladores de eventos

- Os eventos HTML DOM permitem que o Javascript registre diferentes manipuladores de eventos nos elementos de um documento HTML.
- Os eventos são, normalmente, usados em combinação com funções, e a função não será executada antes que o evento ocorra.

Método	descrição
<code>document.getElementById(id).onclick=function(){code}</code>	Adiciona um evento no código com onclick

**Fonte: Elaborado pela autora.**

## 6.1. Métodos usados com elementos do formulário:

### Exemplo: Método getElementById

Sua função simplesmente **retorna o elemento** cujo atributo ID foi especificado.

Considere o seguinte campo de um formulário:

```
<input type="text" id="nomeContato" name="nomeContato" ...
```

Para acessarmos o valor digitado nesse campo de texto em Javascript:

```
document.nomeForm.nomeContato.value
```

document.nomeForm.nomeContato.value

Ou seja, precisamos declarar document mais o nome do formulário em que o campo se localiza e o nome do campo especificado no atributo name.

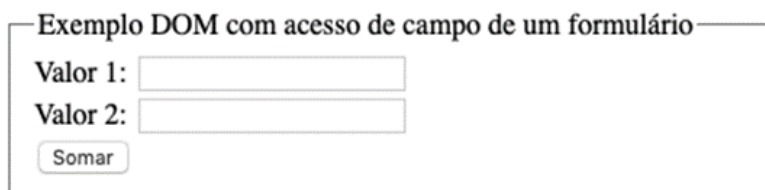
Se utilizarmos o método getElementById, podemos acessá-lo assim:

```
document.getElementById('nomeContato').value
```

### dom\_form.html

```
<html>
<head>
  <meta charset="utf-8">
  <script type="text/javascript">
    function somarValores(){
      var s1 = document.getElementById("s1").value;
      var s2 = document.getElementById("s2").value;
      var s3 = parseInt(s1) + parseInt(s2);
      alert(s3);
    }
  </script>
</head>
<body>
  <fieldset style="width:30%">
    <legend>Exemplo DOM com acesso de campo de um formulário</legend>
    <label>Valor 1:</label>
    <input id="s1" type="text"/><br>
    <label>Valor 2: </label>
    <input id="s2" type="text"/><br>
    <button id="somar" onclick="somarValores()">Somar</button>
  </fieldset>
</body></html>
```

Resultado no navegador: dom\_form.html



Exemplo DOM com acesso de campo de um formulário

Valor 1:

Valor 2:

### evento.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
```

```

</head>
<body>
  <h1 id="id1">Meu título h1!!!</h1>
  <button onclick="document.getElementById('id1').style.color = 'red'">CLique Aqui</button>
</body>

```

Resultado no navegador: evento.html

**Meu título h1!!!** **Meu título h1!!!**

CLique Aqui

CLique Aqui

#### onmouseover.html

```

<html>
<head>
<script>
  function ativa() {
    alert("Evento OnMouseOver ativado!");
  }
</script>
</head>
<body>
  <h2 OnMouseOver="ativa()"> Teste de Evento OnMouseOver </h2>
  <br/><br/><br/>
  <form>
    <input type="reset" value="Botão Reset" OnMouseOver="ativa()"><br/><br/>
    <input type="submit" value="Botão Submit" OnMouseOver="ativa()">
  </form><br/><br/>
  <a href="http://www.mackenzie.br" onclick="alert('Você está testando o evento onclick!!!');">Mackenzie</a>
</body></html>

```

Resultado no navegador: onmouseover.html



#### evento\_onclick\_form\_checkbox.html

```

<!DOCTYPE html>
<html>
<head>
<script>
  function validaCheck() {
    var checks = document.getElementsByClassName("checks");
    var str = " ";
    for (i=0; i<3; i++) {
      if (checks[i].checked === true) {
        str += checks[i].value + " ";
      }
    }
  }

```

```
    }  
    alert("A opção(s) são: " + str);  
  }  
</script>  
</head>  
<body>  
  <form name="form1" method="post">  
    <input type="checkbox" class="checks" value="banana"> A...<br>  
    <input type="checkbox" class="checks" value="maça"> B...<br>  
    <input type="checkbox" class="checks" value="melao"> C...<br><br>  
    <button onclick="validaCheck( );return false;">Enviar</button>  
  </form>  
</body></html>
```

Resultado no navegador: evento\_onclick\_form\_checkbox.html





## 7. VALIDAÇÃO

Como gerenciar propriedades de um formulário HTML usando JS?

Presença de texto

- Verificar se o usuário inseriu alguma coisa no campo de texto.

```
<html>
<head>
<script>
  function verificarTexto() {
    if (form.texto.value == "")
      {alert ("Não há texto escrito")}
  }
</script>
</head>
<body>
  <form name="form">
    <input type="text" name="texto">
    <input type="button
onClic="verificarTexto()">
  </form>
</body>
</html>
```

Ao pressionar o botão, verificar se o campo de texto está vazio ou não dando um alerta como

### Exemplo de validação - validacaoCheck.html

```
<html>
<head>
<title>JavaScript</title>
<script>
  function aviao() {
    document.form1.nome2[0].checked = true;
    document.form1.nome2[1].checked = false;
    document.form1.nome2[2].checked = false;
  }
  function carro() {
    document.form1.nome2[0].checked = false;
    document.form1.nome2[1].checked = true;
    document.form1.nome2[2].checked = false;
  }
  function helicop() {
    document.form1.nome2[0].checked = false;
    document.form1.nome2[1].checked = false;
    document.form1.nome2[2].checked = true;
  }
</script>
</head>
<body>
  <form name="form1">
    Escolha sua compra:
    <br>
    Avião:<input type="radio" name="nome" value="a"
      onClick="aviao()">
    Carro:<input type="radio" name="nome" value="b"
      onClick="carro()">
    Helicóptero:<input type="radio" name="nome" value="c"
      onClick="helicop()">
  </br>
  Cor disponível:
  </br>
```

```
Branco:<input type=checkbox name=nome2 value="a">  
Preto:<input type=checkbox name=nome2 value="b">  
Verde:<input type=checkbox name=nome2 value="c">  
</form>  
</body>  
</html>
```

Escolha sua compra:

Avião: ☒ Carro: ☐ Helicóptero: ☐

Cor disponível:

Branco: ☒ Preto: ☐ Verde: ☐

Simulação de uma compra.

- Ao clicar no radio "Avião", será acionado o checkbox "branco"
- Ao clicar no radio "carro", será acionado o checkbox "preto"
- Ao clicar no radio "helicóptero", será acionado o checkbox "verde".

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

DUCKETT, J. *Javascript & JQuery*: desenvolvimento de interfaces web interativas. Rio de Janeiro: Alta Books, 2015. ISBN 9788576089452.