

MULTIMÍDIA

JavaScript – Básico - Parte II

Agenda

- Funções
- DOM
- Exemplos

Funções

- Funções são blocos de construção fundamentais em JavaScript.
- Uma função é um procedimento de JavaScript - um conjunto de instruções que executa uma tarefa ou calcula um valor.
- Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.

exemplo1.html

```
<body>
  <h1>Exemplo 1</h1>
  <input type="text" name="txtNum1" id="num1" placeholder="Digite o Número">
  <input type="text" name="txtNum2" id="num2" placeholder="Digite o Número">

  <select name="txtop" id="op">
    <option value="+">Adição</option>
    <option value="-">Subtração</option>
    <option value="*">Multiplicação</option>
    <option value="/">Divisão</option>
    <option value="%">Resto</option>
  </select>

  <button onclick="somar((receber()))">SOMAR</button>

  <script src="js/exemplo1.js"></script>
</body>
```

exemplo1.js

```
1
2 //eval() função nativa = executa um cálculo representado como uma string
3 //template strings são strings que permitem expressões embutidas `${}`
4 function somar(x1, x2, op) {
5     if (op == "/" || op == "%") {
6         if (x2 == 0) {
7             document.write("<h2>Divisão por 0</h2>");
8         }
9     }
10    return eval(`${x1} ${op} ${x2}`);
11 }
12
13 function receber() {
14     var n1 = parseFloat(document.getElementById('num1').value);
15     var n2 = parseFloat(document.getElementById('num2').value);
16     var op = document.getElementById('op').value;
17
18     document.write("<h1>Resultado:</h1>");
19     document.write("<h2>" + somar(n1, n2, op) + "</h2>");
20
21 }
```

exemplo2.html

exemplo2.html

04-códigos > exemplo2.html > ...

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>JavaScript Básico - Parte II</title>
8
9  </head>
10
11 <body>
12     <h1>Exemplo 2</h1>
13
14     <input type="text" name="txtNum1" id="num1" placeholder="Digite o Número">
15
16     <button onclick="funcao1()">FUNÇÃO 1</button>
17     <button onclick="receber()">FUNÇÃO 2</button>
18
19     <script src="js/exemplo2.js"></script>
20 </body>
21
22 </html>
```

exemplo2.js

JS exemplo2.js ●

04-códigos > js > JS exemplo2.js > funcao2

```
1  function funcao1() {  
2  
3      var info = {  
4          nome: 'Rodrigo',  
5          sobrenome: 'Martins'  
6      };  
7  
8      document.write("<h1>Resultado</h1>");  
9      document.write("<h2>" + info.nome + "</h2>");  
10     document.write("<h2>" + info.sobrenome + "</h2>");  
11  
12 }  
13
```

exemplo2.js

```
14  function receber() {
15      var n1 = funcao2(parseInt(document.getElementById('num1').value));
16
17      document.write("<h1>Resultado</h1>");
18      document.write("<h2>" + n1(3) + "</h2>");
19      document.write("<h2>" + funcao2(10)(20) + "</h2>");
20
21  }
22
23  function funcao2(x) {
24      return function (y) {
25          return x + y;
26      };
27  }
```


Função anônima

- Uma função anônima é uma definição de função que não está vinculada a um identificador.
- As funções anônimas geralmente são argumentos passados para funções de ordem superior ou usadas para construir o resultado de uma função de ordem superior que precisa retornar uma função.

Arrow Functions

- O JavaScript ES6, trouxe novas features e dentre elas uma nova forma de criar funções usando o operador `=>`.
- Esta nova forma de se trabalhar com funções são chamadas **Arrow Functions**.

exemplo3.html

```
<body>
  <h1>Exemplo 3 - Função Anônima e Arrow Functions</h1>
  <input type="text" name="txtNum1" id="num1" placeholder="Digite o Número">

  <button onclick="receber()">CALCULAR</button>

  <script src="js/exemplo3.js"></script>
</body>

</html>
```

exemplo3.js

```
1  /*
2  função anônima
3  var quadrado = function (numero) {
4  |   return numero * numero;
5  | }
6  */
7
8  // arrow functions
9  var quadrado = (numero) => {
10 |   return numero * numero;
11 | }
12
13 function receber() {
14 |   var n1 = quadrado(parseInt(document.getElementById('num1').value));
15 |   document.write("<h1>Resultado:</h1>" + "<h2>" + n1 + "</h2>");
16 | }
17
```

DOM

- O **Document Object Model** ou simplesmente **DOM** é utilizado pelo navegador Web para representar a sua página Web.
- Quando altera-se esse modelo com o uso do Javascript altera-se também a página Web.
- Um dos grandes responsáveis por isso tudo é o objeto **document** que é responsável por conceder ao código **Javascript** todo o acesso a árvore **DOM** do navegador Web.

exemplo4.html

A propriedade **innerHTML** é útil para obter ou substituir o conteúdo dos elementos HTML.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>JavaScript Básico - Parte II</title>
8
9  </head>
10
11 <body>
12
13     <p id="teste"></p>
14
15     <script>
16         document.getElementById("teste").innerHTML = "<h1>Tem que ter o DOM</h1>";
17     </script>
18
19 </body>
20
21 </html>
```

exemplo5.html

Localizando elemento HTML por **ID**.

```
9    <body>
10      <h1>Exemplo 5</h1>
11
12      <p id="intro">Localizando elemento HTML por ID</p>
13
14      <p id="demo"></p>
15
16      <script>
17          var elemento = document.getElementById("intro");
18          document.getElementById("demo").innerHTML =
19              "O texto do parágrafo de introdução é " + elemento.innerHTML;
20      </script>
21  </body>
22  </html>
```

exemplo6.html

Localizando elementos HTML pelo **nome da tag**.

```
9  <body>
10    <h1>Exemplo 6</h1>
11
12    <p>Localizando elementos HTML pelo nome da tag</p>
13    <p>Tem que ter o DOM</p>
14    <p>Aula de Prog. Scripts</p>
15
16    <p id="demo"></p>
17    <p id="demo1"></p>
18    <p id="demo2"></p>
19
20    <script>
21      var x = document.getElementsByTagName("p");
22      document.getElementById("demo").innerHTML =
23        'O texto do primeiro parágrafo (índice 0) é: ' + x[0].innerHTML;
24
25      x = document.getElementsByTagName("p");
26      document.getElementById("demo1").innerHTML =
27        'O texto do primeiro parágrafo (índice 1) é: ' + x[1].innerHTML;
28
29      x = document.getElementsByTagName("p");
30      document.getElementById("demo2").innerHTML =
31        'O texto do primeiro parágrafo (índice 1) é: ' + x[2].innerHTML;
32    </script>
33  </body>
```


exemplo7.html

Localizando elementos HTML pelo nome da classe.

```
9    <body>
10      <h1>Exemplo 7</h1>
11
12      <p class="intro">Localizando elementos HTML pelo nome da classe</p>
13      <p class="intro">Tem que ter o DOM</p>
14
15      <p id="demo"></p>
16      <p id="demo1"></p>
17
18      <script>
19          var x = document.getElementsByClassName("intro");
20          document.getElementById("demo").innerHTML =
21              '0 texto do primeiro parágrafo (índice 0) é: ' + x[0].innerHTML;
22
23          x = document.getElementsByClassName("intro");
24          document.getElementById("demo1").innerHTML =
25              '0 texto do primeiro parágrafo (índice 1) é: ' + x[1].innerHTML;
26      </script>
27    </body>
28  </html>
```

exemplo8.html

Localizando elementos HTML por **seletores CSS**.

```
<body>
  <h1>Exemplo 8</h1>

  <p>Localizando elementos HTML por seletores CSS - querySelector e querySelectorAll</p>

  <h2 class="teste">h2 class="teste"</h2>
  <p class="teste">Parágrafo class="teste"</p>

  <button onclick="altera()">querySelector</button>
  <button onclick="alteraTudo()">querySelectorAll</button>

  <script>
    function altera() {
      document.querySelector(".teste").style.backgroundColor = "blue";
    }

    function alteraTudo() {
      var x, i;
      x = document.querySelectorAll(".teste");
      for (i = 0; i < x.length; i++) {
        x[i].style.backgroundColor = "yellow";
      }
    }
  </script>
</body>
```

exemplo9.html

Localizando elementos HTML por coleções de objetos HTML

```
11  <body>
12    <h1>Exemplo 9</h1>
13    <h2>Localizando elementos HTML por coleções de objetos HTML</h2>
14
15    <form id="frm1">
16      <p>Nome: <input type="text" name="nome"></p>
17      <p>Sobrenome:<input type="text" name="sobrenome"></p>
18    </form>
19
20    <p>Clique em "Experimente" para exibir o valor de cada elemento no formulário.</p>
21
22    <button onclick="experimente()">Experimente</button>
23
24    <p id="demo"></p>
25
```

exemplo9.html ----- continuação

```
26     <script>
27         function experimete() {
28             var x = document.forms["frm1"];
29             var text = "";
30             var i;
31             for (i = 0; i < x.length; i++) {
32                 text += x.elements[i].value + "<br>";
33             }
34             document.getElementById("demo").innerHTML = text;
35         }
36     </script>
37 </body>
38
39 </html>
```

exemplo10.html

Alterando o conteúdo HTML

```
10  <body>
11    <h1>Exemplo 10</h1>
12
13    <h2 id="demo">Alterando o conteúdo HTML</h2>
14
15    <p id="demo">Tem que ter o DOM</p>
16
17    <script>
18      |   var x = document.getElementById("demo").innerHTML = "Texto Alterado por DOM";
19    </script>
20  </body>
21  </html>
```

exemplo11.html

DOM - Alterando o CSS

```
10  <body>
11      <h1>Exemplo 11</h1>
12
13      <p id="p1">Alterando o CSS</p>
14      <p id="p2">Tem que ter o DOM</p>
15
16      <script>
17          document.getElementById("p2").style.color = "blue";
18          document.getElementById("p2").style.fontFamily = "Arial";
19          document.getElementById("p2").style.fontSize = "larger";
20      </script>
21  </body>
22  </html>
```

exemplo12.html

DOM – Usando Eventos (onclick)

```
11 <body>
12     <h1>Exemplo 12</h1>
13
14     <h1 id="id1">Tem que ter o DOM</h1>
15
16     <button type="button" onclick="document.getElementById('id1').style.color = 'red'">
17         Experimente</button>
18 </body>
19
20 </html>
```

exemplo13.html

DOM – Usando Eventos (onload)

```
<body onload="checaCookies()">

  <h1>Exemplo 13</h1>

  <p id="demo"></p>

  <script>
    function checaCookies() {
      var text = "";
      if (navigator.cookieEnabled == true) {
        text = "Cookies Ativados";
      } else {
        text = "Cookies Desativados";
      }
      document.getElementById("demo").innerHTML = text;
    }
  </script>
</body>
```


exemplo14.html

DOM – Usando Eventos (onchange)

```
11  <body>
12
13      <h1>Exemplo 14</h1>
14
15      Digite seu nome: <input type="text" id="nome" onchange="upper()">
16
17      <p>Saia do campo de entrada, seu nome será transformado em maiúsculo.</p>
18
19      <script>
20          function upper() {
21              var x = document.getElementById("nome");
22              x.value = x.value.toUpperCase();
23          }
24      </script>
25
26  </body>
27  </html>
```

exemplo15.html

DOM – Usando Eventos (onmouseover e onmouseout)

```
11 <body>
12
13 <h1>Exemplo 15</h1>
14
15 <div onmouseover="mOver(this)" onmouseout="mOut(this)"
16 | style="background-color: ■ #D94A38;width:120px;height:40px;padding:40px;">
17 | PASSE O MOUSE AQUI
18 </div>
19
20 <script>
21 | function mOver(obj) {
22 | | obj.innerHTML = "OBRIGADO";
23 | }
24
25 | function mOut(obj) {
26 | | obj.innerHTML = "PASSE O MOUSE AQUI";
27 | }
28 </script>
29
30 </body>
31 </html>
```

exemplo16.html

DOM – Usando Eventos (onmousedown e onmouseup)

```
11 <body>
12
13 <h1>Exemplo 16</h1>
14
15 <div onmousedown="mDown(this)" onmouseup="mUp(this)"
16 style="background-color: #D94A38;width:90px;height:20px;padding:40px;">
17 CLIQUE AQUI</div>
18
19 <script>
20 function mDown(obj) {
21     obj.style.backgroundColor = "#1ec5e5";
22     obj.innerHTML = "SOLTA AGORA";
23 }
24
25 function mUp(obj) {
26     obj.style.backgroundColor = "#D94A38";
27     obj.innerHTML = "CLIQUE AQUI";
28 }
29 </script>
30
31 </body>
32 </html>
```

exemplo17.html

DOM – Usando Eventos (onfocus)

```
11  <body>
12
13      <h1>Exemplo 17</h1>
14
15      Digite seu nome: <input type="text" onfocus="foco(this)">
16
17      <script>
18          function foco(x) {
19              x.style.background = "yellow";
20          }
21      </script>
22  </body>
23
24  </html>
```

O método `addEventListener()` manipulando/ouvindo eventos

- ❑ O método **`addEventListener()`** anexa um manipulador de eventos ao elemento especificado.
- ❑ Você pode adicionar muitos manipuladores de eventos a um elemento.
- ❑ Você pode remover facilmente um ouvinte de eventos usando o método **`removeEventListener()`**.

Sintaxe:

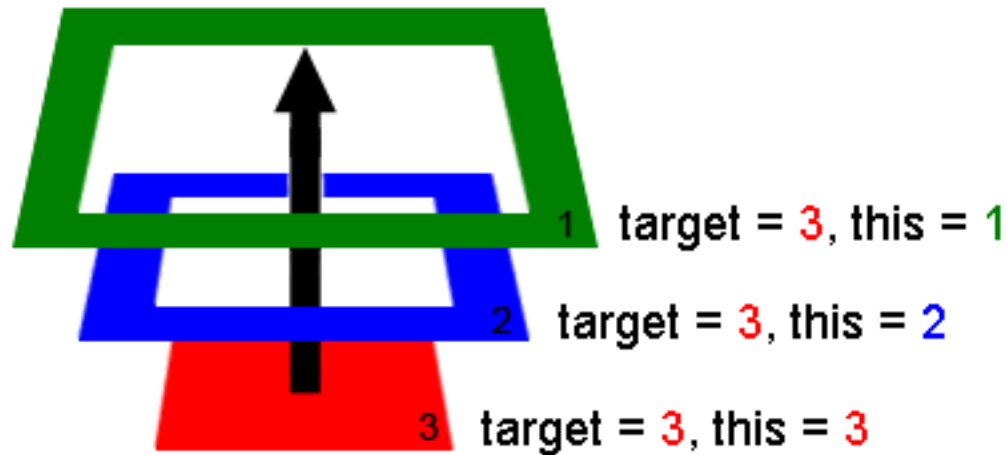
`element.addEventListener(event, function, useCapture);`



o padrão é false (bubbling)

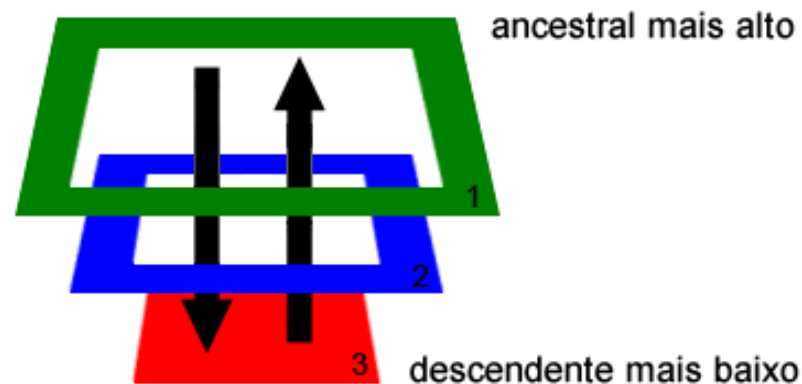
Bubbling

- O princípio fundamental do efeito **bubbling** diz o seguinte: depois que um evento é disparado no elemento mais distante de uma cadeia aninhada do **DOM** ele é disparado em seus elementos ancestrais na ordem crescente de aninhamento.



Capturing

- Em todos os navegadores, exceto nos IE os eventos são processados em dois estágios.
- No primeiro estágio o evento percorre a cadeia aninhada de cima para baixo (dos elementos ancestrais para os elementos descendentes) – este estágio é chamado **capturing**.



- No outro estágio ocorre o efeito bubbles. Este comportamento é padronizado pelas especificações do W3C.

Capturing

- Todos os métodos de manipulação de eventos simplesmente ignoram a fase **capturing**.
- Para que o evento ocorra na fase **capturing** declaramos o último argumento método **addEventListener** como sendo **true**.

exemplo18.html

```
11  <body>
12
13    <h1>Exemplo 18</h1>
14
15    <button id="btn1">Experimente</button>
16
17    <p id="demo"></p>
18
19    <script>
20      document.getElementById("btn1").addEventListener("click", mostraData, false);
21
22      function mostraData() {
23        document.getElementById("demo").innerHTML = Date();
24      }
25    </script>
26
27  </html>
```

exemplo19.html

```
<body>  
  <h1>Exemplo 19</h1>  
  
  <button id="meuBotao">Experimente</button>  
  
  <script src="js/exemplo19.js"></script>  
</html>
```

exemplo19.js

```
function minhaFuncao() {  
  alert("CLICOU!");  
}  
  
document.getElementById("meuBotao").addEventListener('click', minhaFuncao);
```

exemplo20.html

```
11 <body>
12
13 <h1>Exemplo 20</h1>
14
15 <button id="btn1">Experimente</button>
16
17 <script>
18     var x = document.getElementById("btn1");
19     x.addEventListener("click", funcao1);
20     x.addEventListener("click", funcao2);
21
22     function funcao1() {
23         alert("Executando a Função 1");
24     }
25
26     function funcao2() {
27         alert("Executando a Função 2");
28     }
29
30 </script>
31
32 </html>
```

Lista de eventos:

<https://developer.mozilla.org/pt-BR/docs/Web/Events>

exemplo21.html

```
11  <body>
12
13      <h1>Exemplo 21</h1>
14
15      <button id="btn1">Experimente</button>
16
17      <p id="demo"></p>
18
19      <script>
20          var x = document.getElementById("btn1");
21          x.addEventListener("mouseover", funcao1);
22          x.addEventListener("click", funcao2);
23          x.addEventListener("mouseout", funcao3);
24
25          function funcao1() {
26              document.getElementById("demo").innerHTML
27              += "Função 1 executada escutando o evento mouseover!<br>";
28          }
```

exemplo21.html

```
29
30     function funcao2() {
31         document.getElementById("demo").innerHTML
32         += "Função 2 executada escutando o evento click!<br>";
33     }
34
35     function funcao3() {
36         document.getElementById("demo").innerHTML
37         += "Função 3 executada escutando o evento mouseout!<br>";
38     }
39     </script>
40
41 </html>
```

exemplo22.html

```
11  <body>
12
13      <h1>Exemplo 22</h1>
14      <h2>0 evento será disparado ao redimensionar a janela</h2>
15
16      <p id="demo"></p>
17
18      <script>
19          window.addEventListener("resize", function () {
20              document.getElementById("demo").innerHTML = Math.random();
21          });
22      </script>
23
24  </html>
```

exemplo23.html

```
11  <body>
12
13    <h1>Exemplo 23</h1>
14    <h2>Passando valores por parâmetros com o addEventListener</h2>
15
16    <button id="btn1">SOMAR/MULTIPLICAR</button>
17
18    <p id="demo"></p>
19
20    <script>
21      var n1 = 10;
22      var n2 = 10;
23
24      document.getElementById("btn1").addEventListener("click", function () {
25        |    somar(n1, n2);
26      });
```


exemplo23.html

```
27
28     document.getElementById("btn1").addEventListener("mouseover", function () {
29         multiplicar(n1, n2);
30     });
31
32     function somar(a, b) {
33         var resultado = a + b;
34         document.getElementById("demo").innerHTML += "A soma é " + resultado + "<br>";
35     }
36
37     function multiplicar(a, b) {
38         var resultado = a * b;
39         document.getElementById("demo").innerHTML += "A multiplicação é: " + resultado + "<br>";
40     }
41 </script>
42
43 </html>
```

exemplo24.html

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>JavaScript Básico - Parte II</title>
8
9      <style>
10         #Div1,
11         #Div2 {
12             background-color: coral;
13             padding: 50px;
14         }
15
16         #P1,
17         #P2 {
18             background-color: white;
19             font-size: 20px;
20             border: 1px solid;
21             padding: 20px;
22         }
23     </style>
24
25 </head>
```

exemplo24.html

```
26
27 <body>
28
29     <h1>Exemplo 24</h1>
30     <h2>Bubbling X Capturing</h2>
31
32     <div id="Div1">
33         <h2>Bubbling:</h2>
34         <p id="P1">CLIQUE!</p>
35     </div><br>
36
37     <div id="Div2">
38         <h2>Capturing:</h2>
39         <p id="P2">CLIQUE!</p>
40     </div>
41
42     <script>
43         document.getElementById("P1").addEventListener("click", function () {
44             alert("Você clicou no elemento de dentro - Branco");
45             }, false);
46
```



exemplo24.html

```
47     document.getElementById("Div1").addEventListener("click", function () {
48         alert("Agora executou o elemento de Fora - Laranja");
49     }, false);
50
51     document.getElementById("P2").addEventListener("click", function () {
52         alert("Agora executou o elemento de dentro - Branco");
53     }, true);
54
55     document.getElementById("Div2").addEventListener("click", function () {
56         alert("Primeiro executou o elemento de fora - Laranja");
57     }, true);
58     </script>
59
60 </html>
61
```

O método `removeEventListener()`

- O método **`removeEventListener()`** remove os manipuladores de eventos que foram anexados ao método **`addEventListener()`**.

exemplo25.html

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>JavaScript Básico - Parte II</title>
8
9      <style>
10         #div1 {
11             background-color:  red;
12             border: 1px solid;
13             padding: 50px;
14             color:  white;
15             font-size: 20px;
16         }
17     </style>
18
19 </head>
20
```

exemplo25.html

```
21 <body>
22   <div id="div1">
23     <h1>Exemplo 25</h1>
24     <h2>O método removeEventListener() remove os manipuladores de
25       eventos que foram anexados ao método addEventListener()</h2>
26   </div>
27   <button onclick="removeEvento()" id="bt1">REMOVER</button>
28 </div>
29
30 <p id="demo"></p>
31
32 <script>
33   document.getElementById("div1").addEventListener("mousemove", adicEvento);
34
35   function adicEvento() {
36     document.getElementById("demo").innerHTML = Math.random();
37   }
38
39   function removeEvento() {
40     document.getElementById("div1").removeEventListener("mousemove", adicEvento);
41   }
42 </script>
43 </body>
44 </html>
```

Referências Bibliográficas

- <https://developer.mozilla.org/pt-BR/docs/Web/API/Document>
- <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Fun%C3%A7%C3%B5es>
- <https://www.devmedia.com.br/trabalhando-com-dom-em-javascript/29039>
- https://www.w3schools.com/js/js_htmlDOM_elements.asp
- <https://imasters.com.br/front-end/javascript-bubbling-e-capturing>

FIM

Obrigado

Rodrigo