



DOM COM JAVASCRIPT

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HERCULANO DE BIASI

herculano.debiasi@unoesc.edu.br

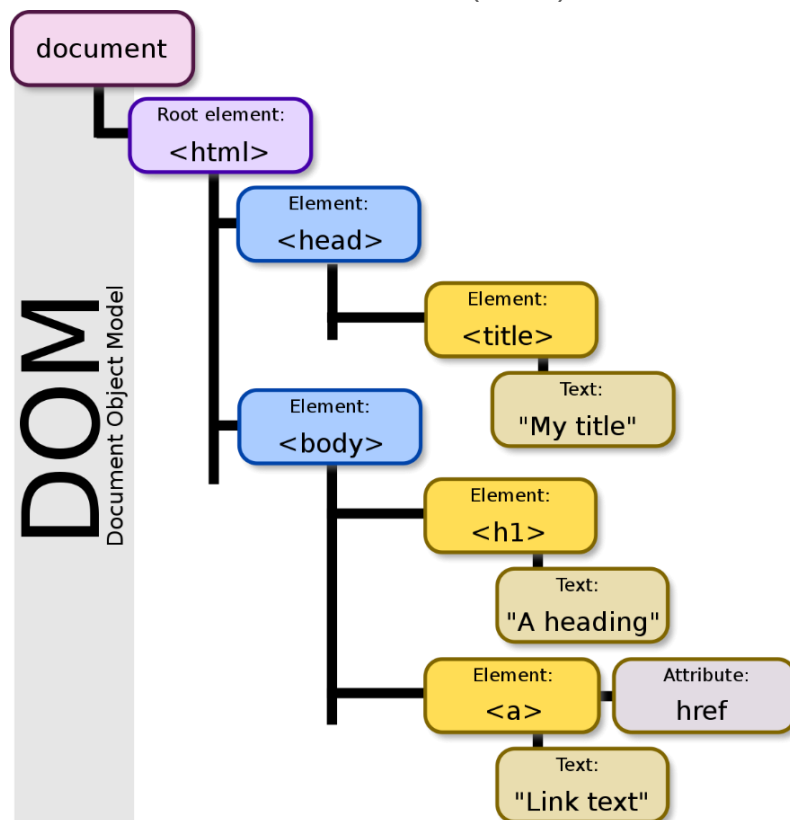


TÓPICOS

- DOM
- Selecionando e manipulando elementos
- Manipulando estilos e classes
- **BOM**
- Eventos
- Formulários

DOM

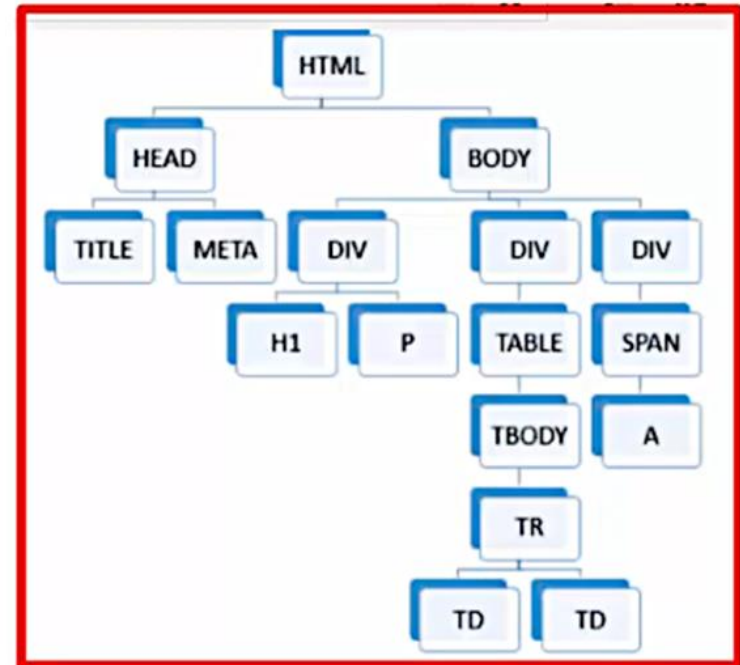
- A representação interna da página é chamada DOM, que significa *Document Object Model* (Modelo de Objetos do Documento) e é criada automaticamente pelo navegador toda vez que um arquivo HTML ou XML válido for carregado
- Cada elemento é chamado de nó ou nodo (*node*)



DOM

■ Exemplo

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>JavaScript</title>
6   </head>
7
8   <body>
9
10    <div>
11      <h1>Exemplo de árvore DOM</h1>
12      <p>Esta página é um modelo para estudo do DOM</p>
13    </div>
14
15    <div>
16      <table>
17        <tbody>
18          <tr>
19            <td>Primeira coluna da linha</td>
20            <td>Segunda coluna da linha</td>
21          </tr>
22        </tbody>
23      </table>
24    </div>
25
26    <div>
27      <span>
28        <a href="#">Link para outra página</a>
29      </span>
30    </div>
31
32  </body>
33 </html>
```

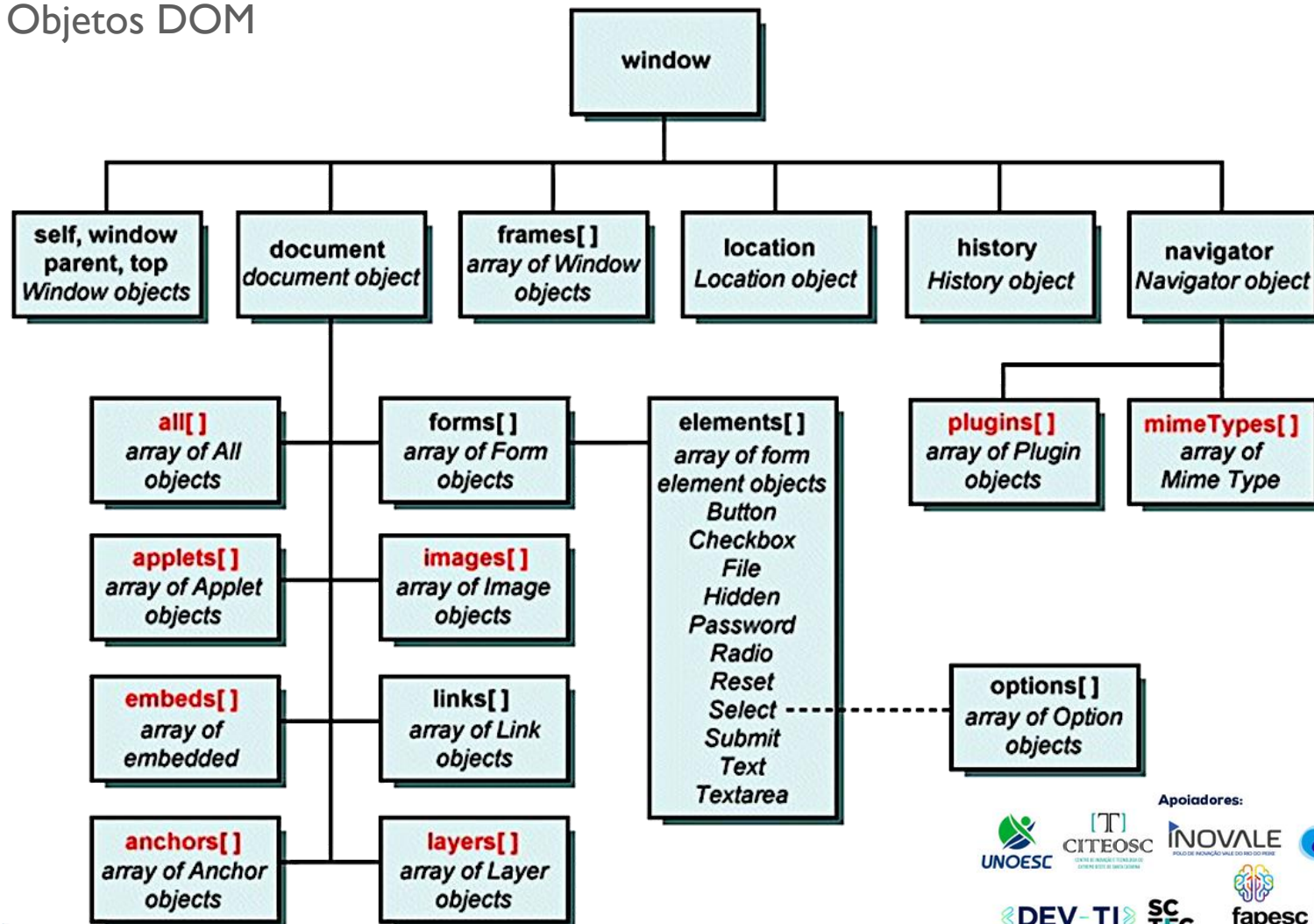


DOM

- DOM é uma *interface* de representação de como os documentos HTML e XML são lidos pelos navegadores
 - Após a leitura da página pelo navegador, é criado um objeto que faz a representação estruturada do documento e define meios para que essa estrutura possa ser acessada e manipulada
 - Tal documento pode ser apresentado no navegador ou como fonte HTML e, devido a sua representação, ele pode ser modificado a partir de linguagens de *script*
- DOM é também considerado uma API que permite acesso, via JavaScript, aos elementos da página
- O DOM e o JavaScript eram inicialmente interligados, mas, com o passar do tempo, avançaram no caminho de se tornar entidades separadas
 - Apesar de o JavaScript ser a linguagem mais utilizada para manipular o DOM, suas implementações podem ser realizadas em qualquer linguagem, já que o DOM foi planejado para ser independente de uma linguagem de programação

DOM

Objetos DOM



DOM

- Quando o JavaScript roda em um navegador, o objeto raiz é chamado `window`
- O objeto `document` é o objeto pai de todos os outros objetos HTML da árvore e, por consequência, da página web
- `window` possui outros objetos filhos como
 - `location`: Acessa a URL atual e redireciona o navegador para outra página
 - `history`: Pode obter as URLs visitadas pelo usuário dentro do navegador, até chegar à página em questão

SELECIONANDO E MANIPULANDO ELEMENTOS

■ Seleccionando elementos

- Métodos `get*` retornam um *array* de elementos `HTMLCollection`
- Métodos `query*` retornam um objeto do tipo `NodeList` (*array-like*)

document (API DOM)

```
getElementById()  
getElementsByTagName()  
getElementsByClassName()  
getElementsByName()
```



```
<input  
  type="text"  
  placeholder="Digite seu nome"  
  id="nome_usuario"  
  class="campo_texto"  
  name="nome"  
>
```

Novos métodos

```
querySelector()  
querySelectorAll()
```


SELECIONANDO E MAN

■ Exemplo utilizando os métodos `get*`

```
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>DOM</title>
9  </head>
10
11 <body>
12     <input
13         type="text"
14         placeholder="Digite seu nome"
15         id="id_nome"
16         class="campo_texto"
17         name="nome">
18
19     <script>
20         console.log(document.getElementById('id_nome'));
21
22         console.log('\nLista de campos <input text>:');
23         console.log(document.getElementsByClassName('campo_texto'));
24
25         console.log('\nLista de campos <input>:');
26         elementos = document.getElementsByTagName('input');
27         for (const elemento of elementos) {
28             console.log(elemento);
29         }
30
31         console.log('\nLista de campos "nome":');
32         elementos = document.getElementsByName('nome');
33         elementos.forEach(elemento => {
34             console.log(elemento);
35         });
36     </script>
37 </body>
38
39 </html>
```

SELECIONANDO E MAN

■ Exemplo utilizando os métodos query*

```
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>DOM</title>
9  </head>
10
11 <body>
12     <input
13         type="text"
14         placeholder="Digite seu nome"
15         id="id_nome"
16         class="campo_texto"
17         name="nome">
18
19     <script>
20         console.log(document.querySelector('#id_nome'));
21
22         console.log('\nLista de campos <input text>:');
23         console.log(document.querySelectorAll('.campo_texto'));
24
25         console.log('\nLista de campos <input>:');
26         elementos = document.querySelectorAll('input');
27         for (const elemento of elementos) {
28             console.log(elemento);
29         }
30
31         console.log('\nLista de campos "nome":');
32         elementos = document.querySelectorAll('[name=nome]');
33         elementos.forEach(elemento => {
34             console.log(elemento);
35         });
36     </script>
37 </body>
38
39 </html>
```

SELECIONANDO E MANIPULANDO ELEMENTOS

■ Exemplo

The screenshot shows a web browser window with a single tab titled "DOM". The address bar displays the URL "127.0.0.1:5500/dom_get_query.html". The page content consists of a single text input field with the placeholder text "Digite seu nome".

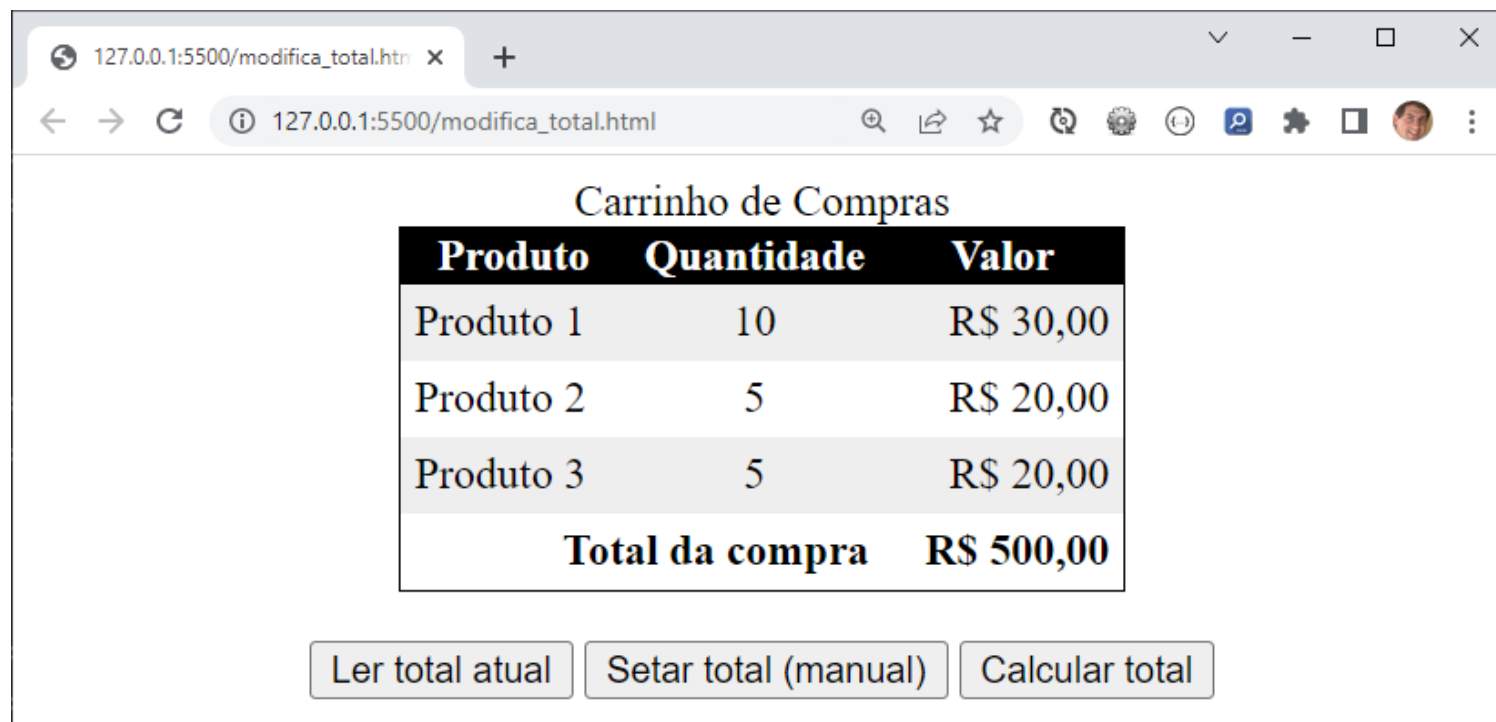
Below the browser window, the Chrome DevTools console is open, showing the "Elements" panel. The console displays the following log entries:

- `input#id_nome.campo_texto` (Source: `dom_get_query.html:20`)
- Lista de campos `<input text>`: (Source: `dom_get_query.html:22`)
- `HTMLCollection [input#id_nome.campo_texto, id_nome: input#id_nome.campo_texto, nome: input#id_nome.campo_texto]` (Source: `dom_get_query.html:23`)
 - `0: input#id_nome.campo_texto`
 - `id_nome: input#id_nome.campo_texto`
 - `nome: input#id_nome.campo_texto`
 - `length: 1`
 - `[[Prototype]]: HTMLCollection`
- Lista de campos `<input>`: (Source: `dom_get_query.html:25`)
- `input#id_nome.campo_texto` (Source: `dom_get_query.html:28`)
- Lista de campos "nome": (Source: `dom_get_query.html:31`)
- `input#id_nome.campo_texto` (Source: `dom_get_query.html:34`)

The console also shows a blue arrow icon at the bottom left, indicating the next log entry.

SELECIONANDO E MANIPULANDO ELEMENTOS

■ Exemplo de acesso e manipulação de elementos



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5500/modifica_total.htm". The page content is titled "Carrinho de Compras" and features a table with the following data:

Produto	Quantidade	Valor
Produto 1	10	R\$ 30,00
Produto 2	5	R\$ 20,00
Produto 3	5	R\$ 20,00
Total da compra		R\$ 500,00

Below the table, there are three buttons: "Ler total atual", "Setar total (manual)", and "Calcular total".

SELECIONANDO E MANIPULANDO ELEMENTOS

■ Exemplo de acesso e manipulação de elementos

```
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Modifica Total</title>
9
10     <style>
11         table {
12             width: 50%;
13             margin: 0 auto;
14             border: 1px solid black;
15             border-collapse: collapse;
16         }
17
18         th {
19             color: white;
20             background-color: black;
21         }
22
23         td { padding: 5px; }
24
25         td:nth-child(even) { text-align: center; }
26         td:last-child { text-align: right; }
27         td[colspan="2"] { text-align: right; }
28
29         table tr:nth-child(even) { background-color: #eee; }
30         table tr:nth-child(odd) { background-color: #fff; }
31         table tr:last-child { font-weight: bold; }
32     </style>
33 </head>
```

```
31 <body>
32     <div>
33         <table>
34             <caption>Carrinho de Compras</caption>
35             <tr>
36                 <th>Produto</th>
37                 <th>Quantidade</th>
38                 <th>Valor</th>
39             </tr>
40             <tr>
41                 <td>Produto 1</td>
42                 <td name="qtd">10</td>
43                 <td>R$ <span name="vlr">30,00</span></td>
44             </tr>
45             <tr>
46                 <td>Produto 2</td>
47                 <td name="qtd">5</td>
48                 <td>R$ <span name="vlr">20,00</span></td>
49             </tr>
50             <tr>
51                 <td>Produto 3</td>
52                 <td name="qtd">5</td>
53                 <td>R$ <span name="vlr">20,00</span></td>
54             </tr>
55             <tr>
56                 <td colspan="2">Total da compra</td>
57                 <td>
58                     R$ <span id="total">0,00</span>
59                 </td>
60             </tr>
61         </table>
62     </div>
63
64     <br>
```

SELECIONANDO E MANIPULANDO ELEMENTOS

■ Exemplo de acesso e manipulação de elementos

```
66 <div style="text-align: center;">
67   <button onclick="alert('Total: R$' + document.querySelector('#total').innerHTML);">
68     Ler total atual
69   </button>
70
71   <button onclick="setarValorTotal()">
72     Setar total (manual)
73   </button>
74
75   <button onclick="calcularValorTotal()">
76     Calcular total
77   </button>
78 </div>
```

SELECIONANDO E MANIPULANDO ELEMENTOS

■ Exemplo de acesso e manipulação de elementos

```
80  <script>
81      function converteValor(dado) {
82          let valor = parseFloat(dado.replace(',', ' '));
83
84          if (!isNaN(parseFloat(valor)) && isFinite(valor)) {
85              return valor.toLocaleString('pt-BR', {
86                  minimumFractionDigits: 2,
87                  maximumFractionDigits: 2
88              });
89          } else {
90              return '0,00';
91          }
92      }
93
94      function setarValorTotal() {
95          let entrada = prompt('Entre com o valor total:', '0,00');
96          document.querySelector('#total').innerHTML = converteValor(entrada);
97      }
```

SELECIONANDO E MANIPULANDO ELEMENTOS

■ Exemplo de acesso e manipulação de elementos

```
99     function calcularValorTotal() {
100         let qtds = document.getElementsByName('qtd');
101         // let qtds = document.querySelectorAll('[name=qtd]');
102         // let vlrs = document.querySelectorAll('[name=vlr]');
103
104         // qtds.forEach(num => { console.log(num.innerHTML); });
105
106         let total = 0;
107         for(let i=0; i<qtds.length; i++) {
108             let qtd = parseFloat(qtds[i].innerHTML);
109
110             total += qtd;
111
112             total = total.toLocaleString('pt-BR', {minimumFractionDigits: 2, maximumFractionDigits: 2});
113             document.querySelector('#total').innerHTML = total;
114         }
115
116         total = total.toLocaleString('pt-BR', {minimumFractionDigits: 2, maximumFractionDigits: 2});
117         document.querySelector('#total').innerHTML = total;
118     }
119
120 </script>
121
122 </body>
123
124 </html>
```


SELECIONANDO E MANIPULANDO ELEMENTOS

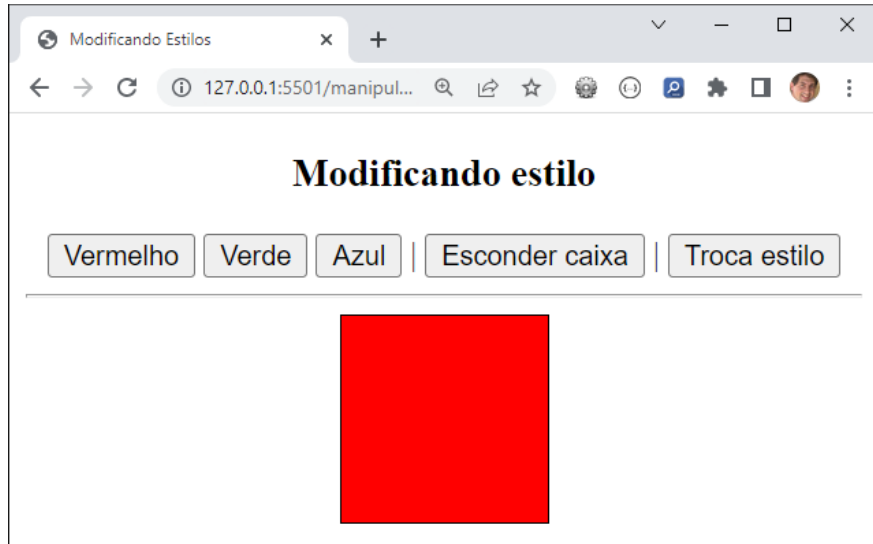
■ Exemplo de manipulação do DOM com JavaScript e jQuery

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>DOM</title>
8
9   <script src="js/jquery-3.6.0.min.js"></script>
10
11   <style>
12     table, th, td {
13       border: 1px solid black;
14       border-collapse: collapse;
15     }
16   </style>
17 </head>
18 <body>
19   <div style="text-align: center">
20     <p>
21       <button onclick="addRowJS()">Adicionar linha JS</button>
22       <button onclick="addRowJQ()">Adicionar linha jQuery</button>
23     </p>
24   </div>
```

```
26 <table id="pessoas">
27   <tr><th>Nome</th><th>E-mail</th></tr>
28   <tr><td>Fulano</td><td>fulano@gmail.com</td></tr>
29   <tr><td>Beltrano</td><td>beltrano@gmail.com</td></tr>
30   <tr><td>Sicrano</td><td>sicrano@gmail.com</td></tr>
31 </table>
32
33 <script>
34   function addRowJS() {
35     let tabela = document.getElementById("pessoas");
36     let linha = tabela.insertRow(tabela.rows.length);
37     let c1 = linha.insertCell(0);
38     let c2 = linha.insertCell(1);
39     c1.innerHTML = 'Zé das Couves';
40     c1.style.color="red";
41     c2.innerHTML = 'zedascouves@gmail.com';
42   }
43
44   function addRowJQ() {
45     let tabela = $("#pessoas tbody");
46     tabela.append(
47       '<tr>'+
48       '<td style="color:blue">Zé das Couves</td>'+
49       '<td>zedascouves@gmail.com</td>'+
50       '</tr>'
51     );
52   }
53 </script>
54 </body>
55 </html>
```

MANIPULANDO ESTILOS E CLASSES

■ Exemplo



MANIPULANDO ESTILOS E CLASSES

■ Exemplo

```
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Modificando Estilos</title>
9
10     <style>
11         .estilo1 {
12             border: solid 1px ■ #000;
13             width: 100px;
14             height: 100px;
15             margin: auto;
16         }
17
18         .estilo2 {
19             border: solid 2px ■ #000;
20             width: 200px;
21             height: 200px;
22             margin: auto;
23         }
24     </style>
25 </head>
```

MANIPUL

■ Exemplo

```
27 <body style="text-align: center;">
28   <h3>Modificando estilo</h3>
29   <button onclick="modificaEstilo('red');">Vermelho</button>
30   <button onclick="modificaEstilo('green');">Verde</button>
31   <button onclick="modificaEstilo('blue');">Azul</button>
32   | <button id='controle' onclick="modificaCaixa();">Esconder caixa</button>
33   | <button id='controle' onclick="trocaEstilo();">Troca estilo</button>
34
35   <hr>
36
37   <div id="caixa" class="estilo1"></div>
38
39   <script>
40     this.caixaInvisivel = false;
41     window.estilo = false;
42
43     function modificaEstilo(corDeFundo) {
44       | document.getElementById('caixa').style.background = corDeFundo;
45       | }
46
47     function modificaCaixa() {
48       | this.caixaInvisivel = !this.caixaInvisivel;
49       | document.getElementById('caixa').hidden = this.caixaInvisivel;
50
51       | if (this.caixaInvisivel) {
52       |   | document.getElementById('controle').innerHTML = 'Mostrar caixa';
53       | } else {
54       |   | document.getElementById('controle').innerHTML = 'Esconder caixa';
55       | }
56     }
57
58     function trocaEstilo() {
59       | document.getElementById('caixa').className = window.estilo ? 'estilo1' : 'estilo2';
60       | window.estilo = !window.estilo;
61     }
62   </script>
63 </body>
64
65 </html>
```

BOM

- **BOM** significa *Browser Object Model* (Modelo de Objetos do Navegador) e consiste em uma API que permite o acesso, via JavaScript, a recursos do navegador

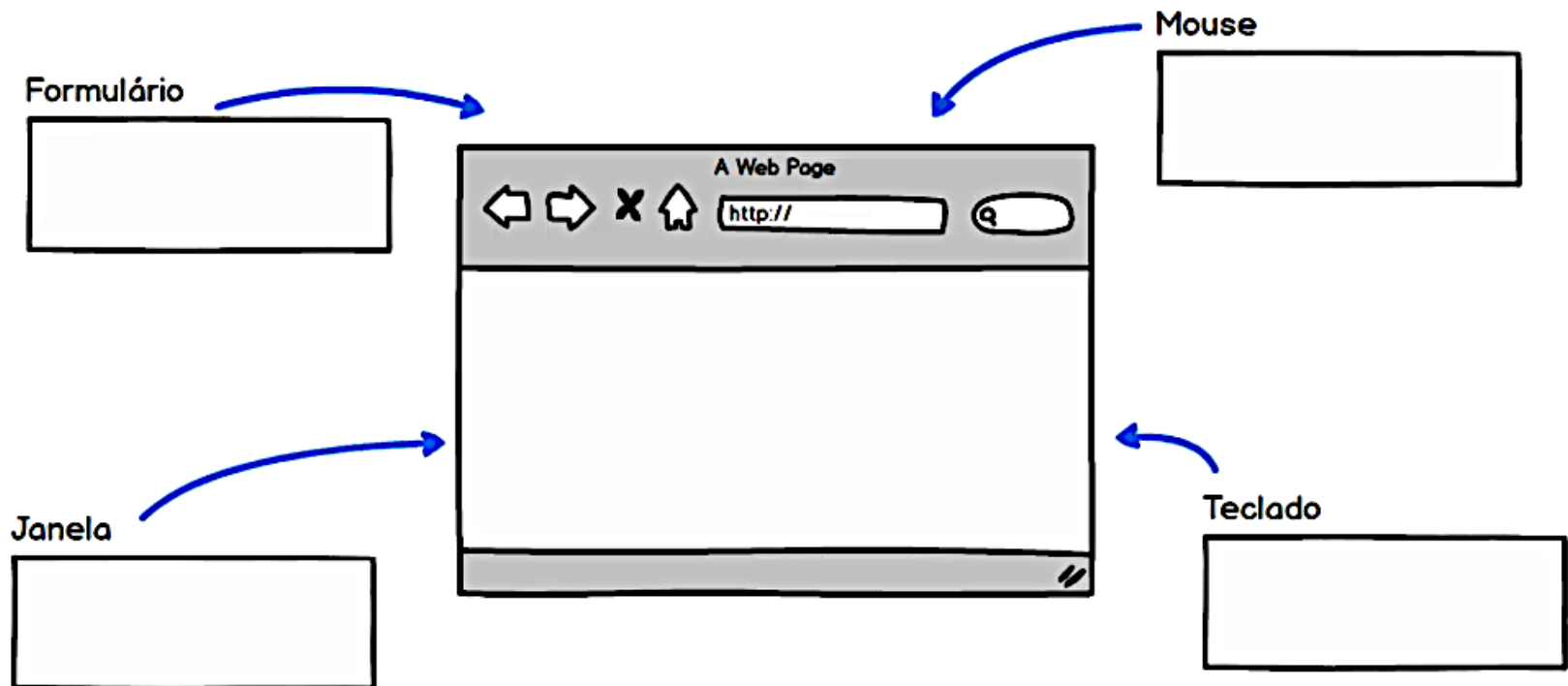
- Principais classes

- Window:
- Screen:
- Location:
- Timing:

EVENTOS

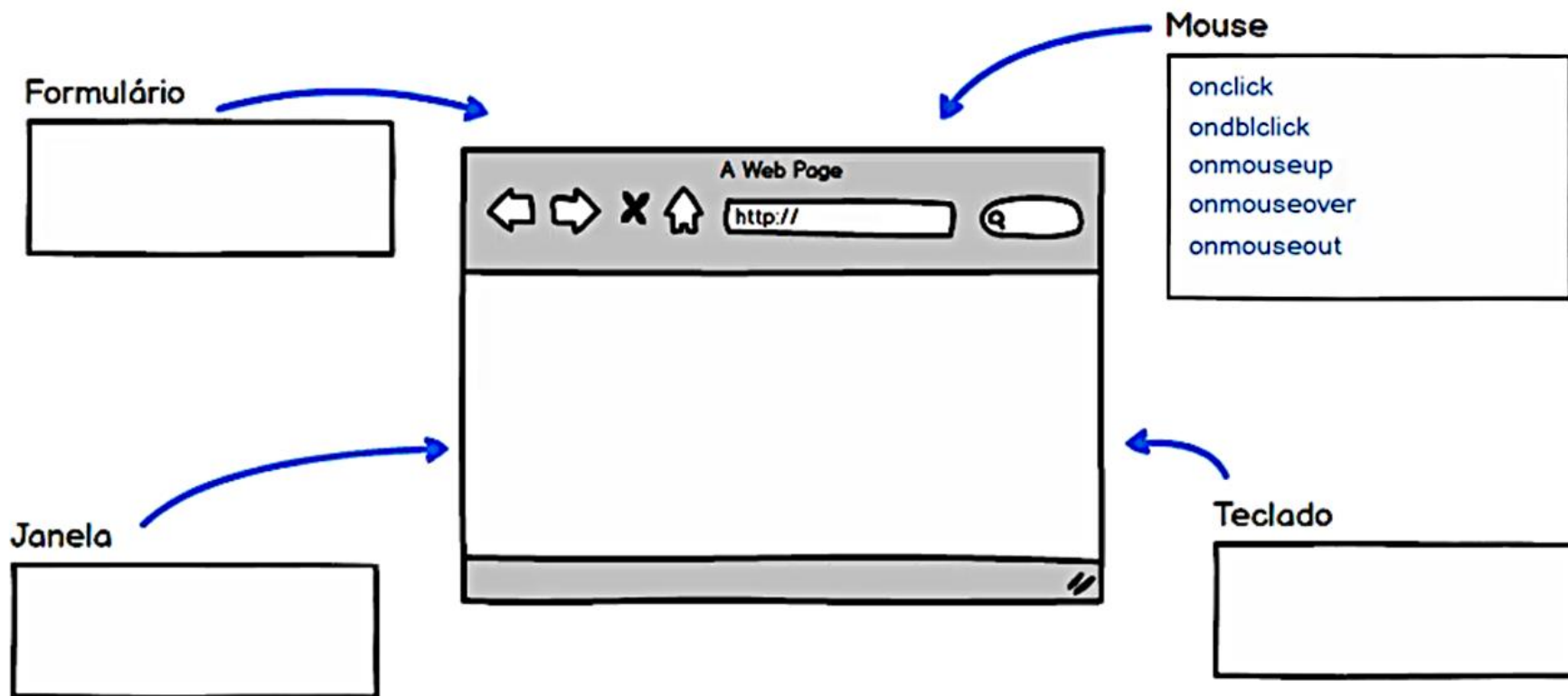
■ Eventos capturados

- Movimento do ponteiro do *mouse* sobre algum elemento da página
- Pressionamento de uma tecla
- Redimensionamento da janela
- Foco do cursor em algum elemento de um formulário
- Carregamento da página no navegador



EVENTOS

■ Eventos do *mouse*

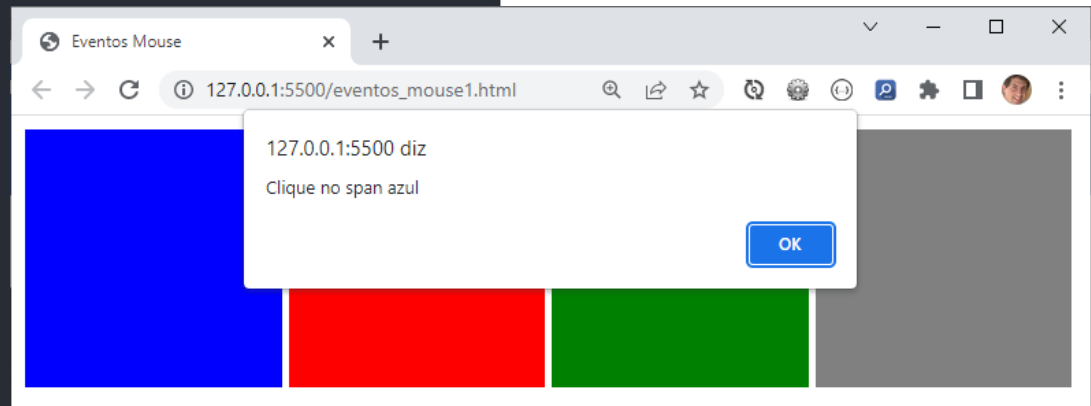


EVENTOS

■ Eventos do mouse

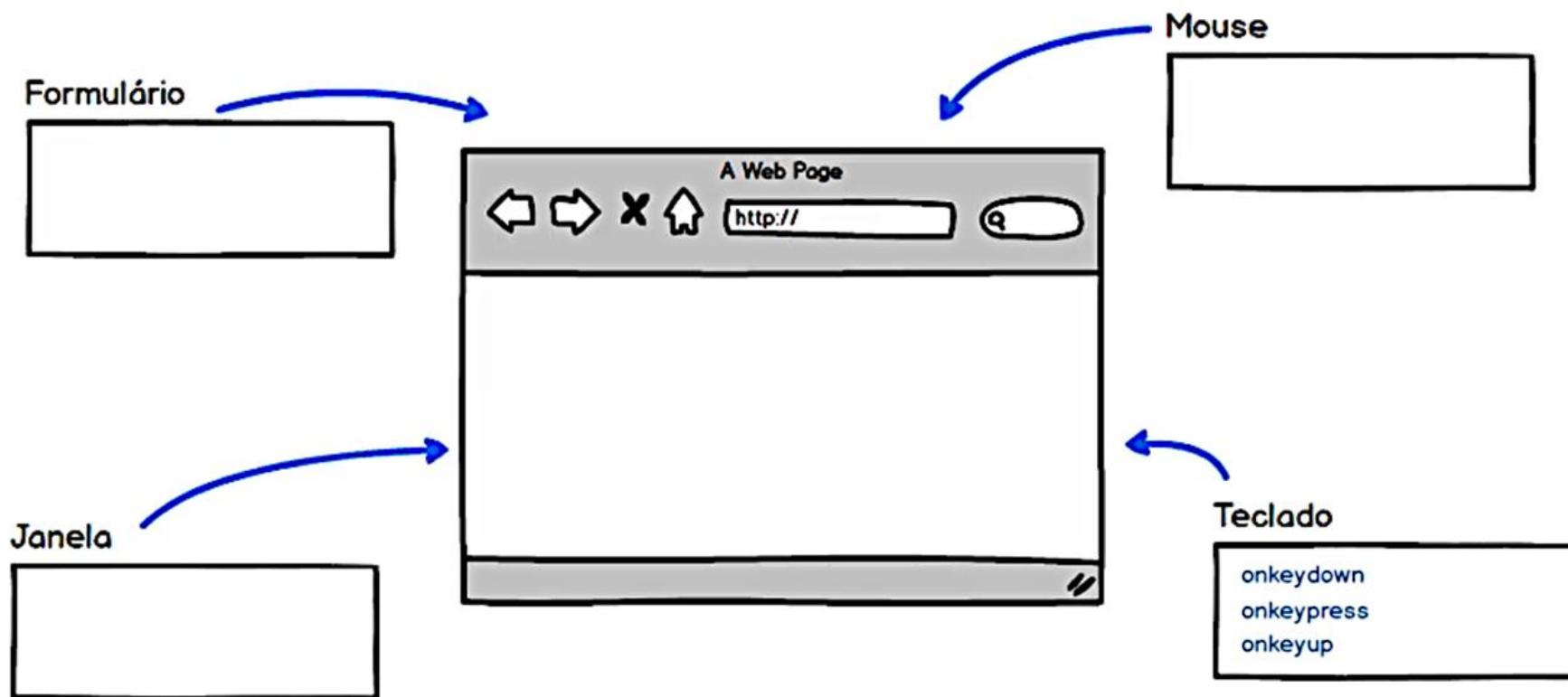
aula7 - eventos_mouse1.html

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Eventos Mouse</title>
8
9   <style>
10     span {
11       display: inline-block;
12       height: 150px;
13       width: 150px;
14     }
15   </style>
16
17   <script>
18     function clicar() {
19       alert('Clique no span vermelho')
20     }
21   </script>
22 </head>
23 <body>
24   <span onclick="alert('Clique no span azul');" style="background: blue;"></span>
25
26   <span onclick="clicar();" style="background: red;"></span>
27
28   <span ondblclick="alert('Duplo clique no span verde');" style="background: green;"></span>
29
30   <span onmouseover="alert('Passando pelo span cinza');" style="background: gray;"></span>
31 </body>
32 </html>
```



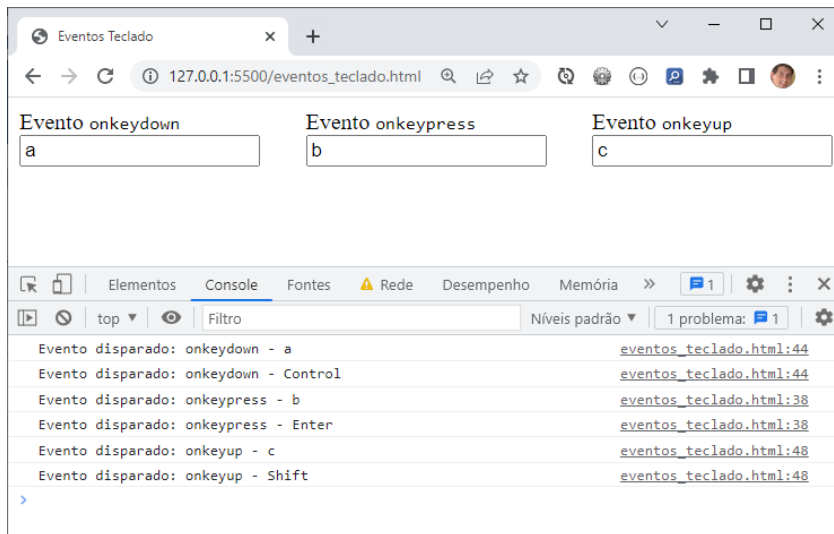
EVENTOS

■ Eventos do teclado



EVENTOS

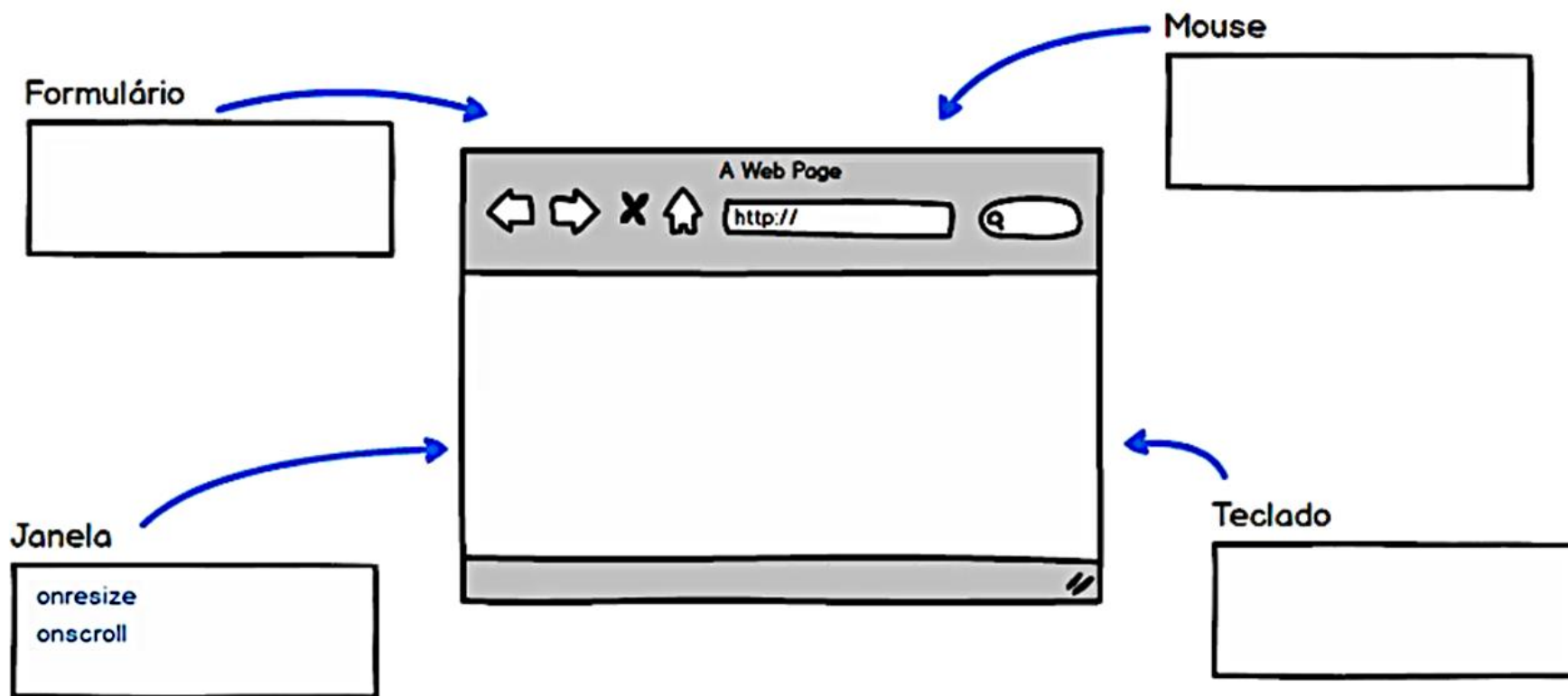
■ Eventos do teclado



```
aula7 - eventos_teclado.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Eventos Teclado</title>
9
10  <style>
11    .app {
12      display: flex;
13      justify-content: space-between;
14    }
15  </style>
16 </head>
17
18 <body>
19   <div class="app">
20     <div>
21       Evento <code>onkeydown</code><br>
22       <input onkeydown="acao1(event)" type="text">
23     </div>
24
25     <div>
26       Evento <code>onkeypress</code><br>
27       <input type="text" id="okpress">
28     </div>
29
30     <div>
31       Evento <code>onkeyup</code><br>
32       <input type="text" id="okup">
33     </div>
34   </div>
35
36   <script>
37     document.querySelector('#okpress').addEventListener('keypress', function acao2(e) {
38       console.log('Evento disparado: onkeypress - ' + e.key);
39     });
40
41     document.querySelector('#okup').addEventListener('keyup', acao3);
42
43     function acao1(e) {
44       console.log('Evento disparado: onkeydown - ' + e.key);
45     }
46
47     function acao3(e) {
48       console.log('Evento disparado: onkeyup - ' + e.key);
49     }
50   </script>
51 </body>
52
53 </html>
```

EVENTOS

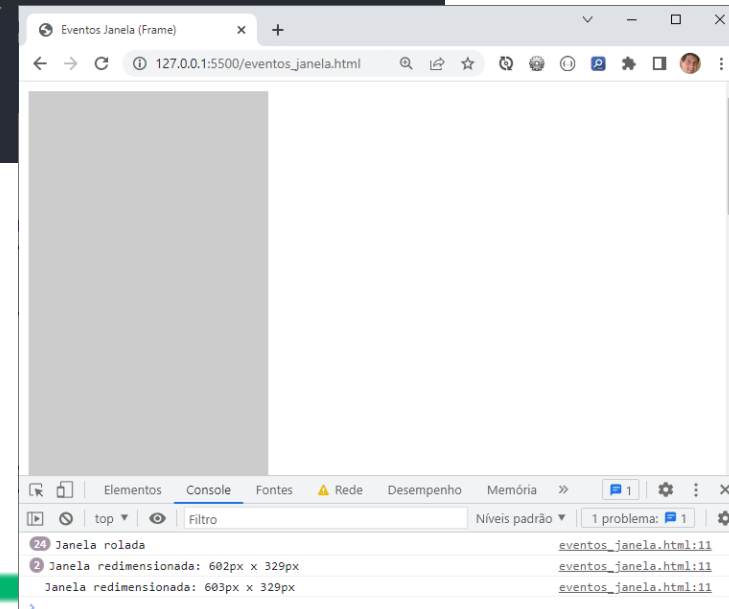
■ Eventos de janela



EVENTOS

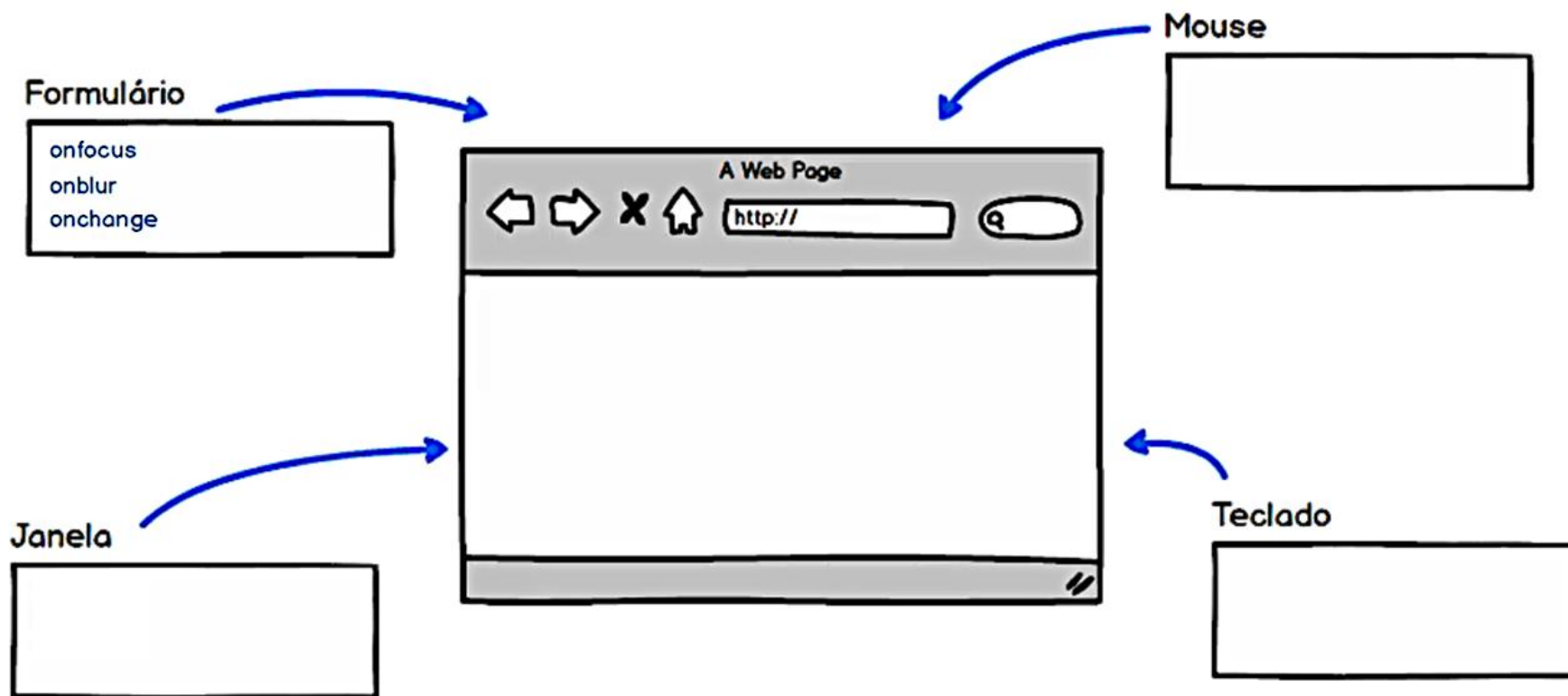
■ Eventos de janela

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Eventos Janela (Frame)</title>
9 </head>
10
11 <body onresize="console.log(`Janela redimensionada: ${window.innerWidth}px x ${window.innerHeight}px`)"
12   onscroll="console.log('Janela rolada');">
13
14   <div style="height: 1000px; width: 200px; background: #ccc;"></div>
15
16 </body>
17
18 </html>
```



EVENTOS

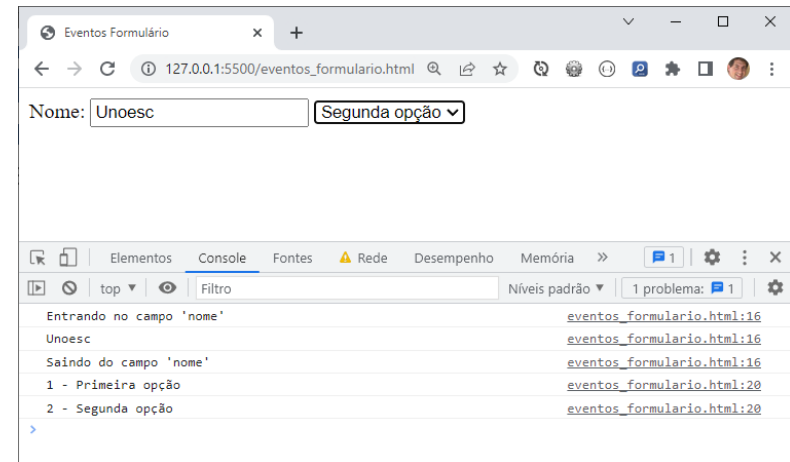
■ Eventos de formulário



EVENTOS

■ Eventos de formulário

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Eventos Formulário</title>
9 </head>
10
11 <body>
12   Nome:
13   <input onfocus="console.log('Entrando no campo \'nome\'')"
14     onblur="console.log('Saindo do campo \'nome\'')"
15     onchange="console.log(document.querySelector('#nome').value)"
16     type="text" id="nome">
17
18   <select onchange="console.log(this.value +
19     ' - ' +
20     this.options[this.selectedIndex].text);">
21     <option selected disabled>-- Selecione --</option>
22     <option value="1">Primeira opção</option>
23     <option value="2">Segunda opção</option>
24   </select>
25 </body>
26
27 </html>
```



FORMULÁRIOS

■ Calculadora

Calculadora

Valor 1:

41 ✓

Digite um número

Valor 2:

1 ✓

Digite um número

Operação

☒ Somar ☐ Subtrair ☐ Multiplicar ☐ Dividir

Calcular

FORMULÁRIOS

■ Exemplo de formulário

Sem Bootstrap

The screenshot shows a web browser window with the title 'Formulário de Contas'. The URL bar shows 'localhost:63342/php/form_operacoes.html?_ijt=...'. The form has a title 'Formulário' in bold. It contains two input fields: 'Valor 1:' with the value '41' and 'Valor 2:' with the value '1'. Below these is a section titled 'Operação' with four radio buttons: 'Somar' (selected), 'Subtrair', 'Multiplicar', and 'Dividir'. At the bottom is a button labeled 'Calcular' with a calculator icon.

Com Bootstrap

The screenshot shows the same web browser window, but the form is styled with Bootstrap. The title 'Formulário' is in a light blue header. The input fields for 'Valor 1' and 'Valor 2' have green borders and green checkmarks on the right. The 'Operação' section has a light gray border and the radio buttons are green. The 'Calcular' button is a blue button with a white calculator icon.

FORMULÁRIOS

- Crie uma pasta chamada *aula5*
- Dentro desta, crie outra com o nome *public*
- Nesta pasta crie o arquivo *index.html* com o código-fonte ao lado, do *front-end*
- Método de envio do formulário (linha 15), pode ser modificado para *post*

```
aula5 - index.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Calculadora</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
8     integrity="sha384-ZenH87qX5JnK21l0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css" crossorigin="anonymous">
10  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
11 </head>
12
13 <body>
14   <div class="container">
15     <form method="get" action="http://localhost:8080/processa-calc" class="was-validated">
16       <h2 class="alert alert-info" style="margin-top: 0.5rem">Calculadora</h2>
17       <div class="form-group col-lg col-md col-sm mb-3">
18         <label for="valor1">Valor 1:</label>
19         <input type="number" name="fValor1" id="iValor1" class="form-control" aria-describedby="valor1Help" placeholder="0" required>
20         <small class="form-text text-muted">Digite um número</small>
21       </div>
22
23       <div class="form-group col-lg col-md col-sm mb-3">
24         <label for="valor2">Valor 2:</label>
25         <input type="number" name="fValor2" id="iValor2" class="form-control" placeholder="0" required>
26         <small class="form-text text-muted">Digite um número</small>
27       </div>
28
29       <div class="form-group col-lg col-md col-sm mb-3">
30         <fieldset class="border p-2" id="operacao">
31           <legend class="w-auto">Operação</legend>
32           <div class="col-auto">
33             <div class="form-check form-check-inline col-auto">
34               <input type="radio" name="fOperacao" id="iSomar" value="1" class="form-check-input" checked>
35               <label class="form-check-label" for="somar">Somar</label>
36             </div>
37             <div class="form-check form-check-inline col-auto">
38               <input type="radio" name="fOperacao" value="2" id="iSubtrair" class="form-check-input">
39               <label class="form-check-label" for="subtrair">Subtrair</label>
40             </div>
41             <div class="form-check form-check-inline col-auto">
42               <input type="radio" name="fOperacao" value="3" id="iMultiplicar" class="form-check-input">
43               <label class="form-check-label" for="multiplicar">Multiplicar</label>
44             </div>
45             <div class="form-check form-check-inline col-auto">
46               <input type="radio" name="fOperacao" value="4" id="iDividir" class="form-check-input">
47               <label class="form-check-label" for="dividir">Dividir</label>
48             </div>
49           </div>
50         </fieldset>
51       </div>
52
53       <div class="form-group col-lg-6 col-md-8 col-sm-10 mb-3">
54         <button type="submit" class="btn btn-primary">
55           <span class="fas fa-calculator"></span>
56           Calcular
57         </button>
58       </div>
59     </form>
60   </div>
61   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
62     integrity="sha384-OERCA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbN3ua0e923+mo//f6V8Qbsw3"
63     crossorigin="anonymous"></script>
64 </body>
65
66 </html>
```

FORMULÁRIOS

■ Criação do *back-end*: Dentro da pasta *aula5*, execute os seguintes comandos

- `npm init -y`
- `npm install express`
- `npm install body-parser`
- `npm install --save-dev nodemon`

■ Acrescentar no *package.json*

- `"scripts": {`
- `"dev": "nodemon server.js",`
- `"test": "echo \"Error: no test specified\" && exit 1"`
- `},`

```
aula5 - package.json
1 {
2   "name": "aula5",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "dev": "nodemon server.js",
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "dependencies": {
14    "body-parser": "^1.20.0",
15    "express": "^4.18.1"
16  },
17  "devDependencies": {
18    "nodemon": "^2.0.20"
19  }
20 }
21
```

FORMULÁRIOS

- Crie o arquivo *server.js* no diretório raiz do projeto (pasta *aula5*)

- Código-fonte do *back-end*

```
1 const express = require('express');
2 const bodyParser = require('body-parser');
3
4 const app = express();
5 const porta = 8080;
6
7 app.use(express.static('public'));
8 app.use(bodyParser.urlencoded({ extended: false }));
9
10 let valor1, valor2, operacao;
11
12 app.get('/processa-calc', (req, res) => {
13   valor1 = parseInt(req.query['fValor1']);
14   valor2 = parseInt(req.query['fValor2']);
15   operacao = parseInt(req.query['fOperacao']);
16
17   processaRequisicao(res);
18 });
19
20 app.post('/processa-calc', (req, res) => {
21   valor1 = parseInt(req.body.fValor1);
22   valor2 = parseInt(req.body.fValor2);
23   operacao = parseInt(req.body.fOperacao);
24
25   processaRequisicao(res);
26 });
27
28 app.get('/', (request, response) => {
29   response.status(200).send();
30 });
31
32 app.listen(porta, () => console.log('Servidor iniciado na porta: ${porta}'));
33
34 function processaRequisicao(res) {
35   let msgAlert = 'O resultado da operação foi: ';
36   let resultado = 0;
37
38   switch (operacao) {
39     case 1: resultado = valor1 + valor2; break;
40     case 2: resultado = valor1 - valor2; break;
41     case 3: resultado = valor1 * valor2; break;
42     case 4:
43       if (valor2 == 0) {
44         msgAlert = 'Impossível dividir por zero!';
45         resultado = '';
46       } else {
47         resultado = valor1 / valor2;
48       }
49       break;
50     default:
51       resultado = 0;
52   }
53
54   msgAlert += resultado;
55   let mensagem = `<h3><div class="alert alert-primary">${msgAlert}</div></h3>`;
56
57   HTML = `
58   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
59   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.2.0/css/all.min.css" crossorigin="anonymous">
60
61   <div class="container" style="margin-top: 0.5rem">
62     ${mensagem}
63
64     <button type="button" onclick="window.history.back()" class="btn btn-outline-danger">
65       <i class="fas fa-door-open"></i>
66       Voltar
67     </button>
68     &nbsp;
69     <button type="button" onclick="alert('${msgAlert}');" class="btn btn-primary">
70       <i class="fab fa-js-square"></i></i>
71       JavaScript
72     </button>
73   </div>
74   `;
75
76   res.send(HTML);
77 }
```

FORMULÁRIOS

■ Processamento das requisições GET e POST no *back-end*

