Introdução às tecnologias Web - ITW

Aula 6 – Javascript (continuação)

10/11/2020

©2014-20, JOAQUIM SOUSA PINTO

Sumário

- A linguagem javascript
  - Variáveis públicas e privadas
  - Temporizadores
- Programando com o javascript
  - Interacção com o DOM (revisitado)
  - Validação de formulários

©2014-20, JOAQUIM SOUSA PINTO

Revisões

Ciclos

©2014-20, JOAQUIM SOUSA PINTO

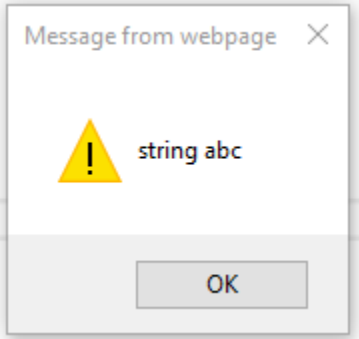
Sintaxe da linguagem Javascript

Condições (switch … case) - (revisão)

Quando há mais que uma condição para testar, é possível a utilização de um conjunto de instruções `if ... else` encadeadas ou, em alternativa, a utilização da instrução `switch … case`.

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script>
        var a = "abc";
        switch (a) {
            case "abc":
                alert("string abc");
                break;
            case 3:
                alert("inteiro 3");
                break;
            default:
                alert("outro");
        }
    </script>
</head>
<body>
    <!-- Conteúdo html aqui ...-->
</body>
</html>
```

- Para cada comparação há uma instrução `case`.
  - Cada instrução `case` deve ser separada por uma instrução `break`. Caso contrário, o programa continuará a fazer as comparações seguintes.
- A instrução `default` será executada caso nenhuma das instruções de comparação tenha sido válida.
  - Esta instrução não precisa do separador `break`.



©

Sintaxe da linguagem Javascript

Condições (switch … case) - (revisão)

```html
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script>
        var a = "abc";
        switch (a) {
            case "abc":
                alert("string abc");
                break;
            case 3:
                alert("inteiro 3");
                break;
            default:
                alert("outro");
        }
    </script>
</head>
<body>
    <!-- Conteúdo html aqui ...-->
</body>
</html>
```

Conversão de um switch em múltiplos if's

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>document</title>
    <script>
        var a = "abc";
        if (a == "abc") {
            alert("string abc");
        }
        else
            if (a = 3) {
                alert("inteiro 3");
            }
            else {
                alert("outro");
            }
    </script>
</head>
<body>
    <!-- Conteúdo html aqui ...-->
</body>
</html>
```


©2014-20, JOAQUIM SOUSA PINTO

Sintaxe da linguagem Javascript

Ciclos (revisão)

Para implementar ciclos, a linguagem JS suporta as instruções `while`, `do-while`, e `for`:

```
do {
     /* instruções */
} while (condição);
```

```
while (condição) {
    /* instruções */
}
```

```
 for (inicio; comparação; incremento) {
    /* instruções */
}
```

Diferenças entre os diversos tipos de ciclos:

- `do-while` – as instruções do ciclo são executadas pelo menos uma vez porque a `condição` de comparação é executada no fim do ciclo;
- `while` – as instruções do ciclo são executadas 0 ou mais vezes, pois o ciclo só se realiza se a `condição` se verificar à partida;
- `for` – as instruções do ciclo são executadas um número fixo de vezes – desde o `início` até à `comparação` com um `incremento`.

©2014-20, JOAQUIM SOUSA PINTO

Sintaxe da linguagem Javascript

Ciclos (revisão)

```
var i = 7;
var a = 2;
do {
    a += ++i;
}
while (i < 5);
console.log(a);


var i = 7;
var a = 2;
while (i < 5) {
    a += i++;
}
console.log(a);

var a = 1;
for (var i = 2; i <= 5; i++) {
    a *= i;
}
console.log(a);
```

?

©2014-20, JOAQUIM SOUSA PINTO

Variáveis públicas e privadas

Revisitando a aula prática anterior ....

©2014-20, JOAQUIM SOUSA PINTO

Variáveis globais e locais

Variáveis globais são variáveis declaradas fora de uma função, ao contrário das locais que são declaradas dentro de uma função.

- Quando uma variável é declarada dentro de uma função, ela só poderá ser usada dentro da mesma.
- As variáveis globais podem ser usadas em qualquer contexto.

Apesar da facilidade que as variáveis globais oferecem, também têm as suas desvantagens.

- Quando uma variável é declarada como global ela ocupará memória até que o programa seja finalizado – por isso devem ser usadas apenas quando necessário.
- As variáveis locais só ocupam memória enquanto aquela função for executada, ao término dessa função ela é destruída.

©2014-20, JOAQUIM SOUSA PINTO

```
var operacao = "+";

function getOperacao() {
    var e = document.getElementById("operacao");
    operacao = e.options[e.selectedIndex].value;
    console.log(operacao);
}

function calcula() {
    /* Vamos precisar de código aqui ... */
    var op1 = document.getElementById("op1"); /* op1 é o marcador
<input id="op1" ...> todo! */
    var op2 = document.getElementById("op2");
    var res = document.getElementById("res");
    switch (operacao) {
        case "+":
            res.value = parseFloat(op1.value) + parseFloat(op2.value);
            //res.value = op1.value + op2.value;
            break;
        case "-":
            res.value = op1.value - op2.value;
            break;
        default:
            alert("Erro: operação não definida ...");
    }
}

console.log(operacao);
console.log(op1.value);
console.log(op2.value);
```

Qual o resultado destas operações?

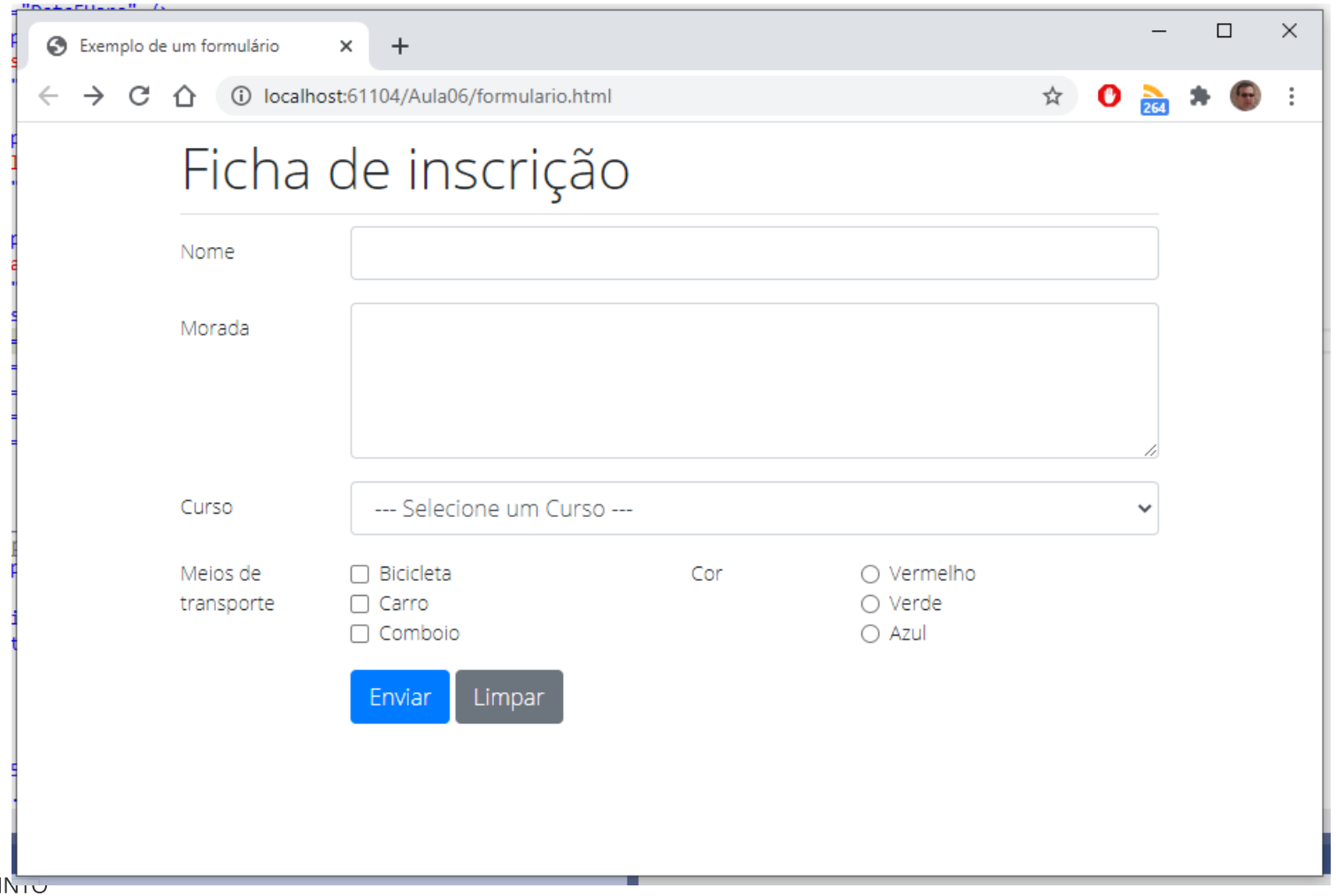
©2014-20, JOAQUIM SOUSA PINTO

Validação de formulários

O javascript em ação…

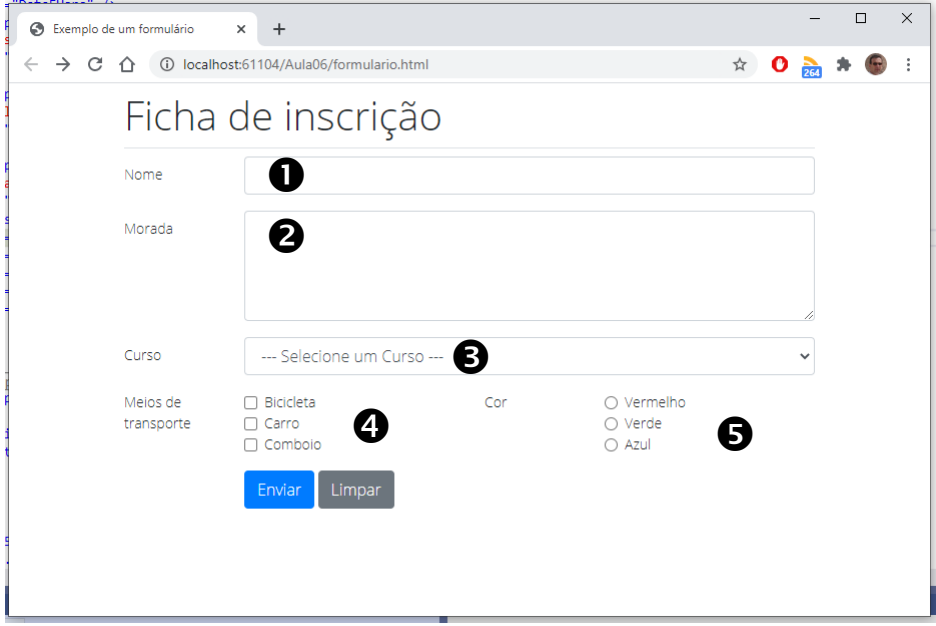
©2014-20, JOAQUIM SOUSA PINTO

Revisitando o exemplo da aula 2
(com a inclusão dos estilos bootstrap)

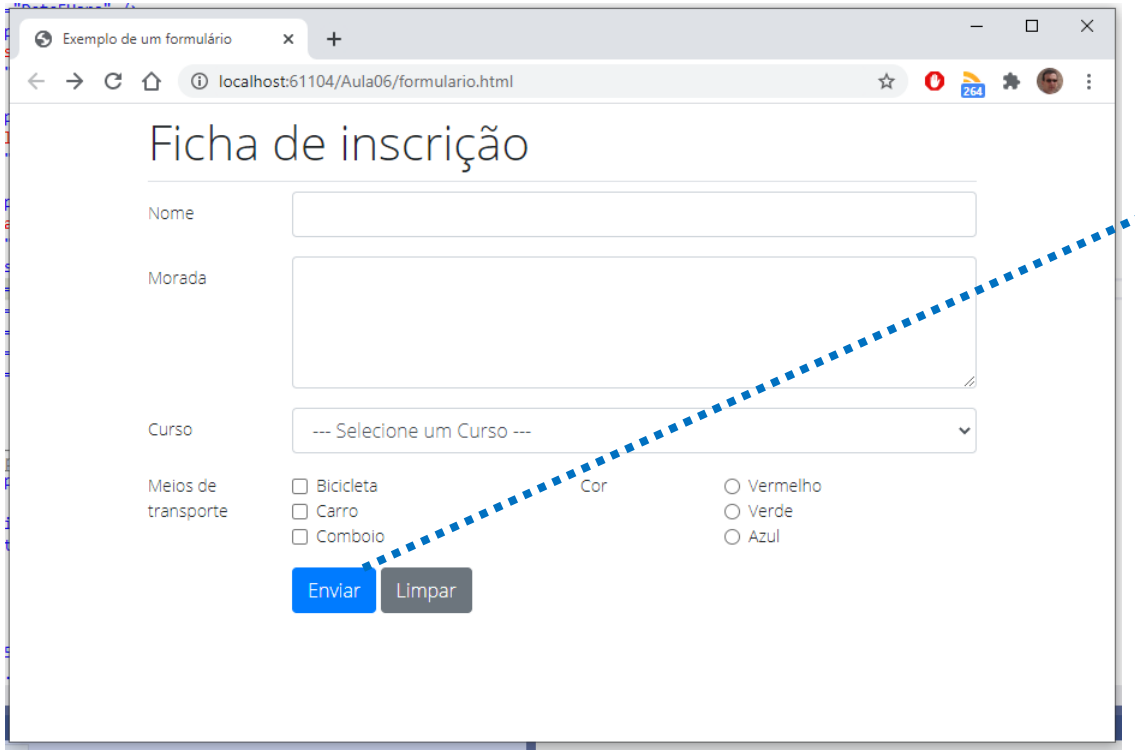


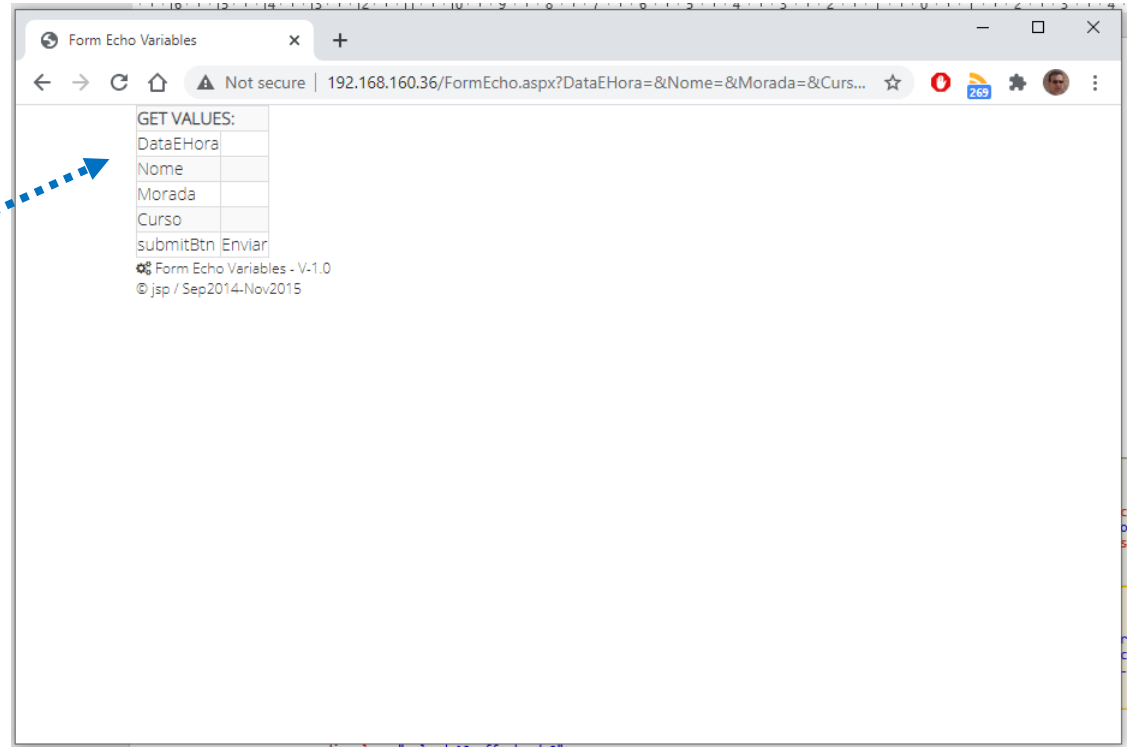
©2014-20, JOAQUIM SOUSA PINTO

```html
<body>
    <div class="container">
        <h1 class="border-bottom py-2 mb-2">Ficha de inscrição</h1>
        <form action="http://192.168.160.36/FormEcho.aspx" name="FichaInscricao">
❶       <div class="row form-group">
            <label for="Nome" class="col-md-2 col-form-label">Nome</label>
            <div class="col-md-10"><input type="text" name="Nome" id="Nome" class="form-control" /></div>
        </div>
❷       <div class="row form-group">
            <label for="Morada" class="col-md-2 col-form-label">Morada</label>
            <div class="col-md-10"><textarea name="Morada" id="Morada" class="form-control" rows="4"></textarea></div>
        </div>
❸       <div class="row form-group">
            <label for="Curso" class="col-md-2 col-form-label">Curso</label>
            <div class="col-md-10">
                <select name="Curso" id="Curso" class="form-control">
                    <option value="">--- Selecione um Curso ---</option>
                    <option value="1">Licenciatura em Engenharia Informática</option>
                    <option value="2">Mestrado Integrado em Engenharia de Computadores e Telemática</option>
                    <option value="3">Mestrado Integrado em Engenharia Eletrónica e Telecomunicações</option>
                    <option value="9">Outro</option>
                </select>
            </div>
        </div>
        <div class="row form-group">
❹           <div class="col-md-2">Meios de transporte</div>
            <div class="col-md-4 ">
                <div class="form-check">
                    <input type="checkbox" name="vehicle" class="form-check-input" value="Bicicleta" id="bicicleta"><label class="form-check-label" for="bicicleta">Bicicleta</label><br />
                    <input type="checkbox" name="vehicle" class="form-check-input" value="Carro" id="carro"><label class="form-check-label" for="carro">Carro</label><br />
                    <input type="checkbox" name="vehicle" class="form-check-input" value="Comboio" id="comboio"><label class="form-check-label" for="comboio">Comboio</label>
                </div>
            </div>
❺           <div class="col-md-2">Cor</div>
            <div class="col-md-4 ">
                <div class="form-check">
                    <input type="radio" name="cor" class="form-check-input" value="Vermelho" id="vermelho"><label class="form-check-label" for="vermelho">Vermelho</label><br />
                    <input type="radio" name="cor" class="form-check-input" value="Verde" id="verde"><label class="form-check-label" for="verde">Verde</label><br />
                    <input type="radio" name="cor" class="form-check-input" value="Azul" id="azul"><label class="form-check-label" for="azul">Azul</label>
                </div>
            </div>
        </div>
        <div class="row form-group">
            <div class="col-md-10 offset-md-2">
                <input type="Submit" name="submitBtn" value="Enviar" class="btn btn-primary" />
                <input type="reset" name="resetBtn" value="Limpar" class="btn btn-secondary" />
            </div>
        </div>
    </form>
    </div>
    <script src="../Scripts/jquery-3.5.1.min.js"></script>
    <script src="../Scripts/bootstrap.bundle.min.js"></script>
</body>
```



Exemplo da aula 2
(com a inclusão dos estilos bootstrap)





©2014-20, JOAQUIM SOUSA PINTO

Problema

Os dados foram submetidos ao servidor sem terem sido preenchidos!

- Esta situação deve ser evitada pois implica que o servidor seja obrigado a fazer a validação dos dados antes de fazer o processamento desejado.
- Se pensarmos num sistema com milhões de utilizadores, obrigaremos o servidor a uma quantidade de trabalho que poderia ser distribuída por cada um dos computadores individuais (se os clientes validassem os dados antes do seu envio).

Para fazer a validação dos campos do formulário do lado do cliente, deve ser utilizado um `<script></script>` javascript de modo a garantir que os dados estão de acordo com o desejado antes de serem enviados ao servidor.

©2014-20, JOAQUIM SOUSA PINTO

Revisão de formulários html
(agora em conjunto com javascript)

MUITO IMPORTANTE:

1. O CSS e o javascript referem-se aos elementos pelo atributo `ID`
2. Os `label's` referem-se aos elementos pelo atributo `ID`
3. Os dados dos formulários são enviados ao servidor pelo atributo `name`

©2014-20, JOAQUIM SOUSA PINTO

formulario.html* site.css

name Current Document

```html
10  <div class="container">
11      <h1 class="border-bottom py-2 mb-2">Ficha de inscrição</h1>
12      <form action="http://192.168.160.36/FormEcho.aspx" name="FichaInscricao">
13          <div class="row form-group">
14              <label for="Nome" class="col-md-2 col-form-label">Nome</label>
15              <div class="col-md-10"><input type="text" name="Nome" id="Nome" class="form-control" /></div>
16          </div>
17          <div class="row form-group">
18              <label for="Morada" class="col-md-2 col-form-label">Morada</label>
19              <div class="col-md-10"><textarea name="Morada" id="Morada" class="form-control" rows="4"></textarea></div>
20          </div>
21          <div class="row form-group">
22              <label for="Curso" class="col-md-2 col-form-label">Curso</label>
23              <div class="col-md-10">
24                  <select name="Curso" id="Curso" class="form-control">
25                      <option value="">--- Selecione um Curso ---</option>
26                      <option value="1">Licenciatura em Engenharia Informática</option>
27                      <option value="2">Mestrado Integrado em Engenharia de Computadores e Telemática</option>
28                      <option value="3">Mestrado Integrado em Engenharia Eletrónica e Telecomunicações</option>
29                      <option value="9">Outro</option>
30                  </select>
31              </div>
32          </div>
33          <div class="row form-group">
34              <div class="col-md-2">Meios de transporte</div>
35              <div class="col-md-4 ">
36                  <div class="form-check">
37                      <input type="checkbox" name="vehicle" class="form-check-input" value="Bicicleta" id="bicicleta"><label class="form-check-label" for="bicicleta">Bicicleta</label><br />
38                      <input type="checkbox" name="vehicle" class="form-check-input" value="Carro" id="carro"><label class="form-check-label" for="carro">Carro</label><br />
39                      <input type="checkbox" name="vehicle" class="form-check-input" value="Comboio" id="comboio"><label class="form-check-label" for="comboio">Comboio</label>
40                  </div>
41              </div>
42              <div class="col-md-2">Cor</div>
43              <div class="col-md-4 ">
44                  <div class="form-check">
45                      <input type="radio" name="cor" class="form-check-input" value="Vermelho" id="vermelho"><label class="form-check-label" for="vermelho">Vermelho</label><br />
46                      <input type="radio" name="cor" class="form-check-input" value="Verde" id="verde"><label class="form-check-label" for="verde">Verde</label><br />
47                      <input type="radio" name="cor" class="form-check-input" value="Azul" id="azul"><label class="form-check-label" for="azul">Azul</label>
48                  </div>
49              </div>
50          </div>
51          <div class="row form-group">
52              <div class="col-md-10 offset-md-2">
53                  <input type="Submit" name="submitBtn" value="Enviar" class="btn btn-primary" />
54                  <input type="reset" name="resetBtn" value="Limpar" class="btn btn-secondary" />
55              </div>
56          </div>
```

105 % No issues found Ln: 15 Ch: 63 SPC CRLF

©2014-20, JOAQUIM SOUSA PINTO

formulario.html* site.css

id Current Document

```html
8   </head>
9   <body>
10      <div class="container">
11          <h1 class="border-bottom py-2 mb-2">Ficha de inscrição</h1>
12          <form action="http://192.168.160.36/FormEcho.aspx" name="FichaInscricao">
13              <div class="row form-group">
14                  <label for="Nome" class="col-md-2 col-form-label">Nome</label>
15                  <div class="col-md-10"><input type="text" name="Nome" id="Nome" class="form-control" /></div>
16              </div>
17              <div class="row form-group">
18                  <label for="Morada" class="col-md-2 col-form-label">Morada</label>
19                  <div class="col-md-10"><textarea name="Morada" id="Morada" class="form-control" rows="4"></textarea></div>
20              </div>
21              <div class="row form-group">
22                  <label for="Curso" class="col-md-2 col-form-label">Curso</label>
23                  <div class="col-md-10">
24                      <select name="Curso" id="Curso" class="form-control">
25                          <option value="">--- Selecione um Curso ---</option>
26                          <option value="1">Licenciatura em Engenharia Informática</option>
27                          <option value="2">Mestrado Integrado em Engenharia de Computadores e Telemática</option>
28                          <option value="3">Mestrado Integrado em Engenharia Eletrónica e Telecomunicações</option>
29                          <option value="9">Outro</option>
30                      </select>
31                  </div>
32              </div>
33              <div class="row form-group">
34                  <div class="col-md-2">Meios de transporte</div>
35                  <div class="col-md-4 ">
36                      <div class="form-check">
37                          <input type="checkbox" name="vehicle" class="form-check-input" value="Bicicleta" id="bicicleta"><label class="form-check-label" for="bicicleta">Bicicleta</label><br />
38                          <input type="checkbox" name="vehicle" class="form-check-input" value="Carro" id="carro"><label class="form-check-label" for="carro">Carro</label><br />
39                          <input type="checkbox" name="vehicle" class="form-check-input" value="Comboio" id="comboio"><label class="form-check-label" for="comboio">Comboio</label>
40                      </div>
41                  </div>
42                  <div class="col-md-2">Cor</div>
43                  <div class="col-md-4 ">
44                      <div class="form-check">
45                          <input type="radio" name="cor" class="form-check-input" value="Vermelho" id="vermelho"><label class="form-check-label" for="vermelho">Vermelho</label><br />
46                          <input type="radio" name="cor" class="form-check-input" value="Verde" id="verde"><label class="form-check-label" for="verde">Verde</label><br />
47                          <input type="radio" name="cor" class="form-check-input" value="Azul" id="azul"><label class="form-check-label" for="azul">Azul</label>
48                      </div>
49                  </div>
50              </div>
51              <div class="row form-group">
52                  <div class="col-md-10 offset-md-2">
53                      <input type="Submit" name="submitBtn" value="Enviar" class="btn btn-primary" />
54                      <input type="reset" name="resetBtn" value="Limpar" class="btn btn-secondary" />
```

105 % No issues found Ln: 15 Ch: 73 SPC CRLF

Validação de formulários

Neste momento estão criadas as condições para a validação dos valores de um formulário antes do seu envio para o servidor.

Condições de validação do formulário:

- O nome tem de ter no mínimo três letras;
- A morada tem de ter no mínimo três palavras;
- Tem de escolher um curso
- Tem de escolher (pelo menos) dois meios de transporte
- Tem de escolher uma cor

©2014-20, JOAQUIM SOUSA PINTO

Validação de formulários

o botão de submit

Na linha do botão de submit deve ser acrescentada a instrução `onclick="return validate()"`

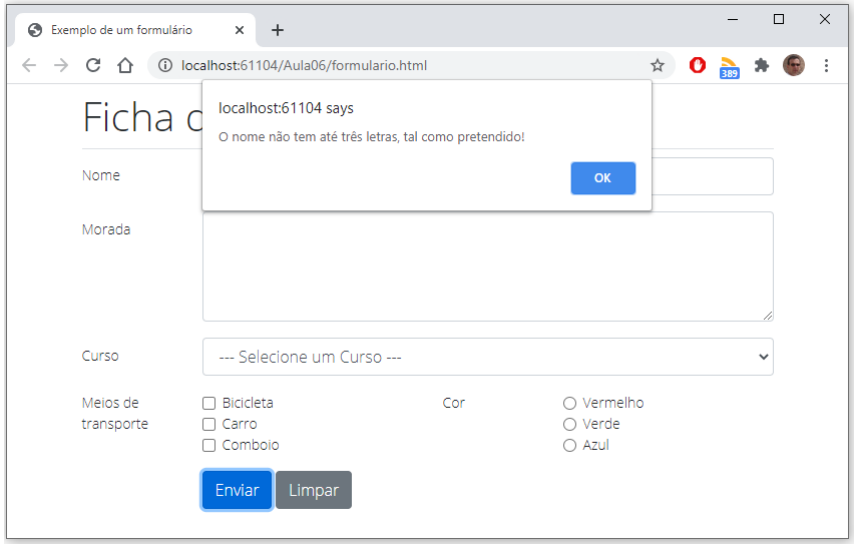
- Quando o botão for carregado, a função `validate()` será executada antes do envio do formulário.
  - se retornar `true` o formulário será enviado;
  - se retornar `false` o formulário não será enviado e o utilizador poderá corrigir os valores.

```
<form action="http://192.168.160.36/FormEcho.aspx" name="FichaInscricao">
(…)
   <div class="row form-group">
       <div class="col-md-10 offset-md-2">
           <input type="Submit" name="submitBtn" value="Enviar" class="btn btn-primary" onclick="return validate()" />
           <input type="reset" name="resetBtn" value="Limpar" class="btn btn-secondary" />
       </div>
    </div>
</form>
```


©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

1. O nome tem de ter no mínimo três letras



Ação: limpar os espaços e verificar o tamanho do texto inserido.

Se o formulário for submetido sem nome deve aparecer um `alert` indicando o erro.

```
<script>
   /* Função de validação  */
   function validate() {
       var retVal = true; /* Vamos partir do princípio de que o formulário está válido ... */
       var _nome = document.getElementById("Nome");
       if (_nome.value.trim().length < 3) {
            retVal = false;
            alert("O nome não tem o mínimo de três letras! ");
        }
        return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Funções e métodos utilizados

`trim()` - O método trim() remove os espaços em branco (whitespaces) do início e/ou fim de um texto.

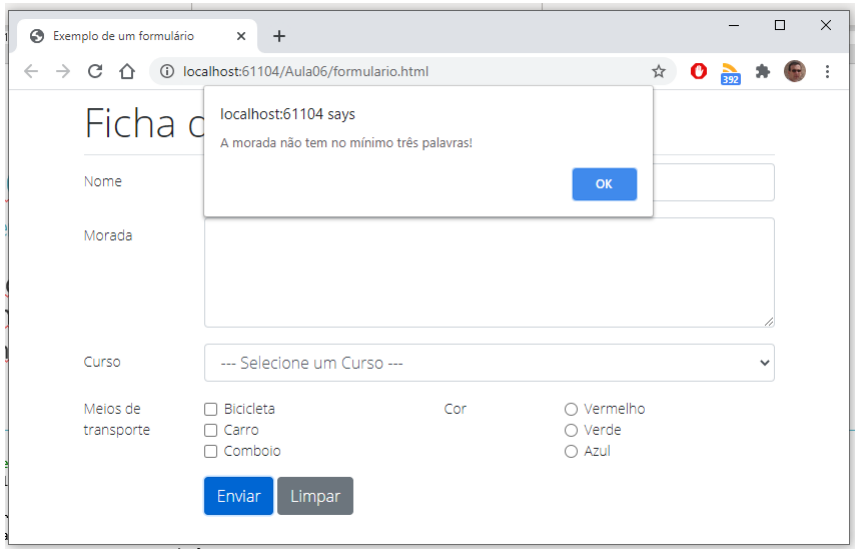
- É considerado espaço em branco (espaço, tabulação, espaço fixo/rígido, etc.) e qualquer sinal de fim de linha de texto (LF, CR, etc.).

`length` - A propriedade length de um objeto String contém o comprimento da string.

©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

2. A morada tem de ter no mínimo três palavras



Ação: dividir a morada em palavras e contar o número de palavras.

Se o formulário for submetido sem nome deve aparecer um `alert` indicando o erro.

```
<script>
    /* Função de validação  */
    function validate() {
(…)
        var _morada = document.getElementById("Morada");
        var palavrasArray = _morada.value.split(' ');
        if (palavrasArray.length < 3) {
            retVal = false;
            alert("A morada não tem no mínimo três palavras!");
        }
        return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Funções e métodos utilizados

`split()` - O método split() divide uma String em uma lista ordenada de substrings, coloca essas substrings num `array` e retorna o array.

A divisão é feita procurando um padrão, onde o padrão é fornecido como o primeiro parâmetro na chamada do método.

`array` - Um array é um conjunto de dados. Arrays são utilizados para armazenar mais de um valor numa única variável. Isso é comparável a uma variável que pode armazenar apenas um valor.

Cada item do array tem índice numérico que permite o acesso ao "valor" armazenado na váriavel.

Em JavaScript, um array começa no índice zero e pode ser manipulado a partir de vários métodos.

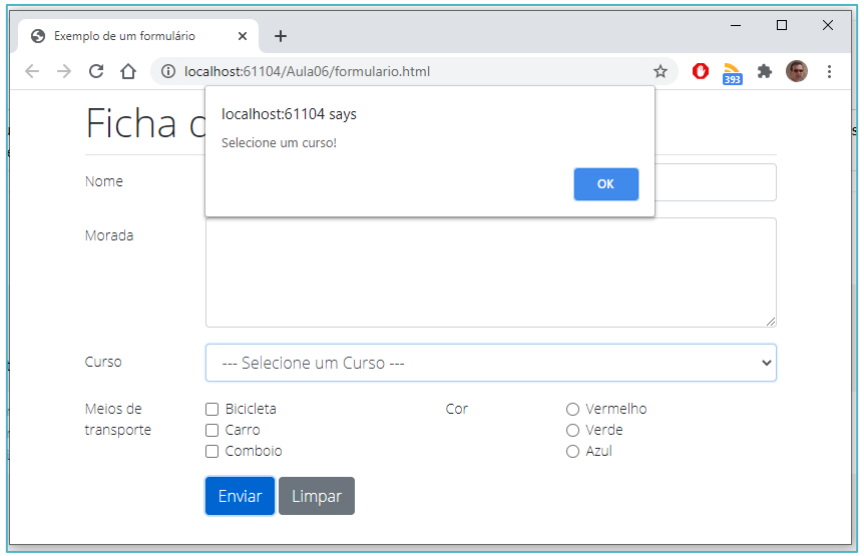
```
var myArray = [1, 2, 3, 4];
var nomesArray = ["Joaquim", "Ana", "Pedro"];
console.log(nomesArray[0]) /* Retornará Joaquim */
```


©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

3. Tem de escolher um curso (opção 1)

Ação: determinar qual o índice selecionado.
Se o índice for = 0, não há curso selecionado

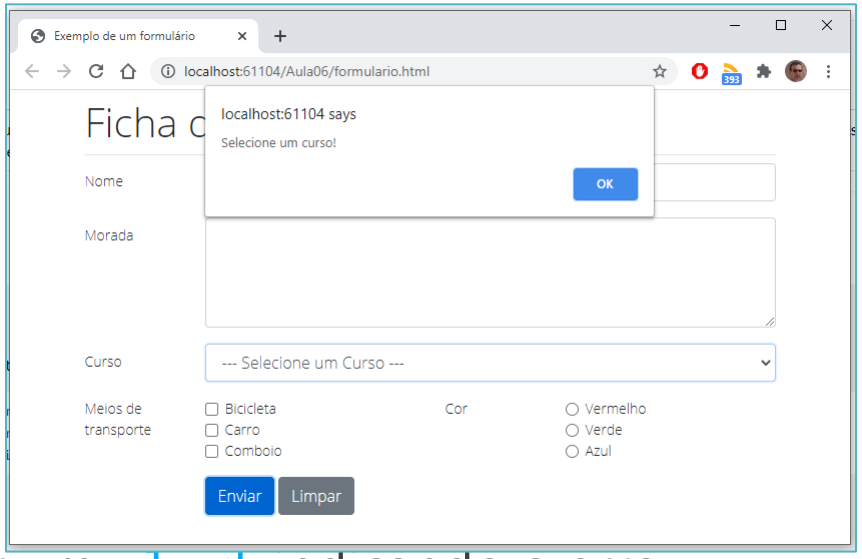



```
<script>
    /* Função de validação  */
    function validate() {
(...)
        var _cursoSelecionado = document.getElementById("Curso").selectedIndex;
        /* Se o indice é zero, não está nenhum curso selecionado */
        if (_cursoSelecionado == 0) {
            retVal = false;
            alert("Selecione um curso!");
        }
    return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

3. Tem de escolher um curso (opção 2)



Ação: determinar o índice selecionado e verificar nas opções o valor da opção do índice. Se for = "", não há curso selecionado

Se o formulário for submetido sem nome deve aparecer um `alert` indicando o erro.

```
<script>
    /* Função de validação  */
    function validate() {
(…)
        var _cursoSelecionado = document.getElementById("Curso").selectedIndex;
        var cursosArray = document.getElementById("Curso").options;
        /* Se o valor da opção _cursoSelecionado tem valor = "", não está nenhum curso selecionado */
        if (cursosArray[_cursoSelecionado].value == "") {
            retVal = false;
            alert("Selecione um curso!");
        }
        return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Propriedades utilizadas

`HTMLSelectElement.selectedIndex` - O `HTMLSelectElement.selectedIndex` é um longo (`long`) que reflete o índice do elemento `<option>` selecionado. O valor -1 indica que nenhum elemento está selecionado.

`HTMLSelectElement.options` - A propriedade `HTMLSelectElement.options` retorna um `HTMLOptionsCollection` dos elementos `<option>` contidos no elemento `<select>`. A propriedade `HTMLSelectElement.options` é somente de leitura.

©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

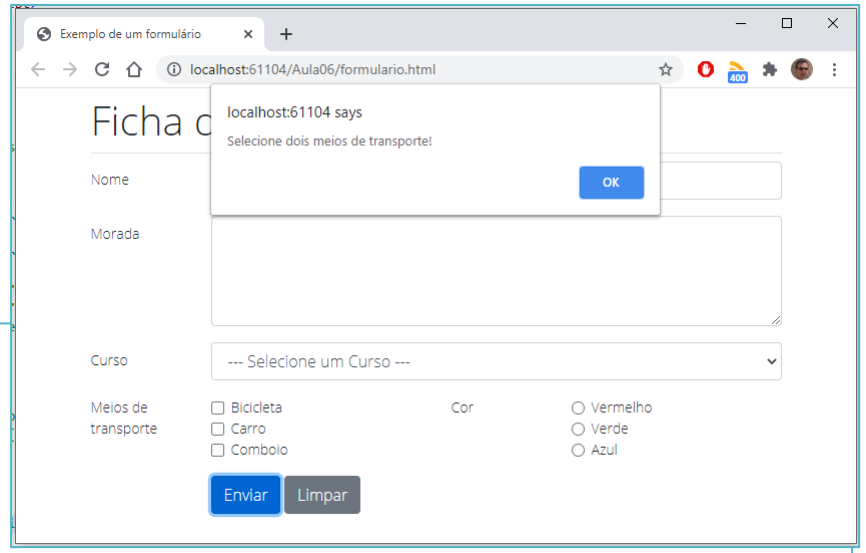
4. Tem de escolher (pelo menos) dois meios de transporte (opção 1)

```
<script>
    /* Função de validação  */
    function validate() {
(…)
        /* Recolher todos os inputs */
        var inputElemsArray = document.getElementsByTagName("input");
        var count = 0;
        for (var i = 0; i < inputElemsArray.length; i++) {
            /* Contar quais os que são do tipo checkbox e estão selecionados */
            if (inputElemsArray[i].type == "checkbox" && inputElemsArray[i].checked == true) {
                count++;
            }
        }
        if (count < 2) {
            retVal = false;
            alert("Selecione dois meios de transporte!");
        }
        return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

4. Tem de escolher (pelo menos) dois meios de transporte (opção 1)

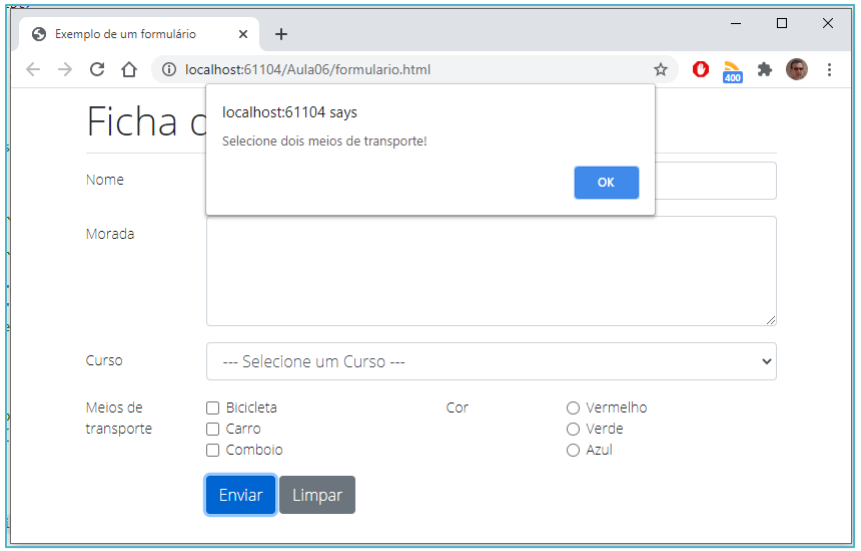



```
<script>
    /* Função de validação  */
    function validate() {
(…)
        /* Recolher todos os inputs */
        var inputElemsArray = document.getElementsByTagName("input");
        var count = 0;
        for (var i = 0; i < inputElemsArray.length; i++) {
            /* Contar quais os que são do tipo checkbox e estão selecionados */
            if (inputElemsArray[i].type == "checkbox" && inputElemsArray[i].checked == true) {
                count++;
            }
        }
        if (count < 2) {
            retVal = false;
            alert("Selecione dois meios de transporte!");
        }
        return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

4. Tem de escolher (pelo menos) dois meios de transporte (opção 2)

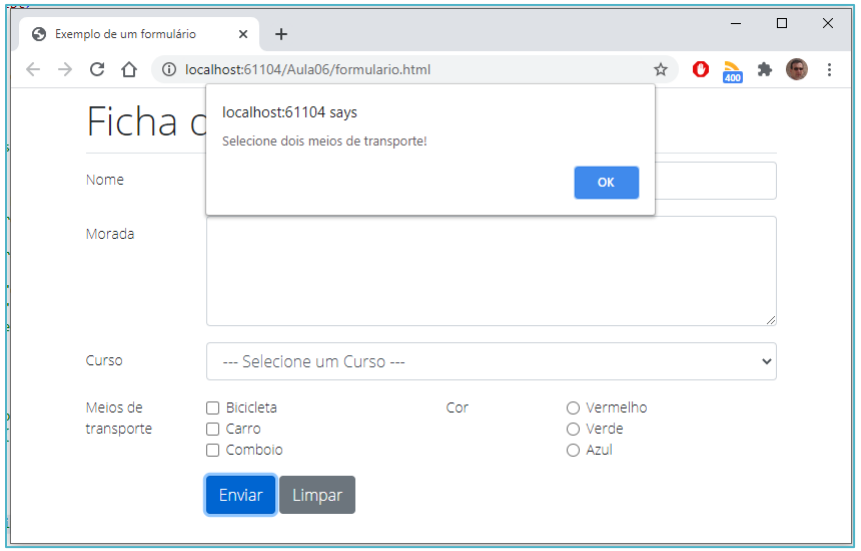



```
<script>
    /* Função de validação  */
    function validate() {
(…)
        /* Opção 2: recolher todos os input[type="checkbox"] que estejam checked */
        var _transportesSelecionados = document.querySelectorAll('input[type="checkbox"]:checked').length;
        if (_transportesSelecionados < 2) {
            retVal = false;
            alert("Selecione dois meios de transporte!");
        }
        return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

4. Tem de escolher (pelo menos) dois meios de transporte (opção 3)




```
<script>
    /* Função de validação  */
    function validate() {
(…)
        /* Opção 2: recolher todos os input[name="vehicle"] que estejam checked */
        var _transportesSelecionados = document.querySelectorAll('input[name="vehicle"]:checked').length;
        if (_transportesSelecionados < 2) {
            retVal = false;
            alert("Selecione dois meios de transporte!");
        }
        return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Métodos utilizados

`Element.getElementsByTagName(tag)` - O método `Element.getElementsByTagName()` retorna uma `HTMLCollection` de elementos com o nome de `tag` fornecido.

`document.querySelectorAll(seletores)` - O método `document.querySelectorAll(seletores)` retorna uma `NodeList` que representa a lista dos elementos do documento que correspondem ao grupo de seletores especificado.

©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

4. Tem de escolher uma cor

```
<script>
    /* Função de validação  */
    function validate() {
(…)
        /* Recolher todos os input[type="radio"] que estejam checked */
        var _cores = document.querySelectorAll('input[name="cor"]:checked').length;
        if (_cores == 0) {
            retVal = false;
            alert("Selecione uma cor!");
        }
        return retVal;
    }
</script>
```

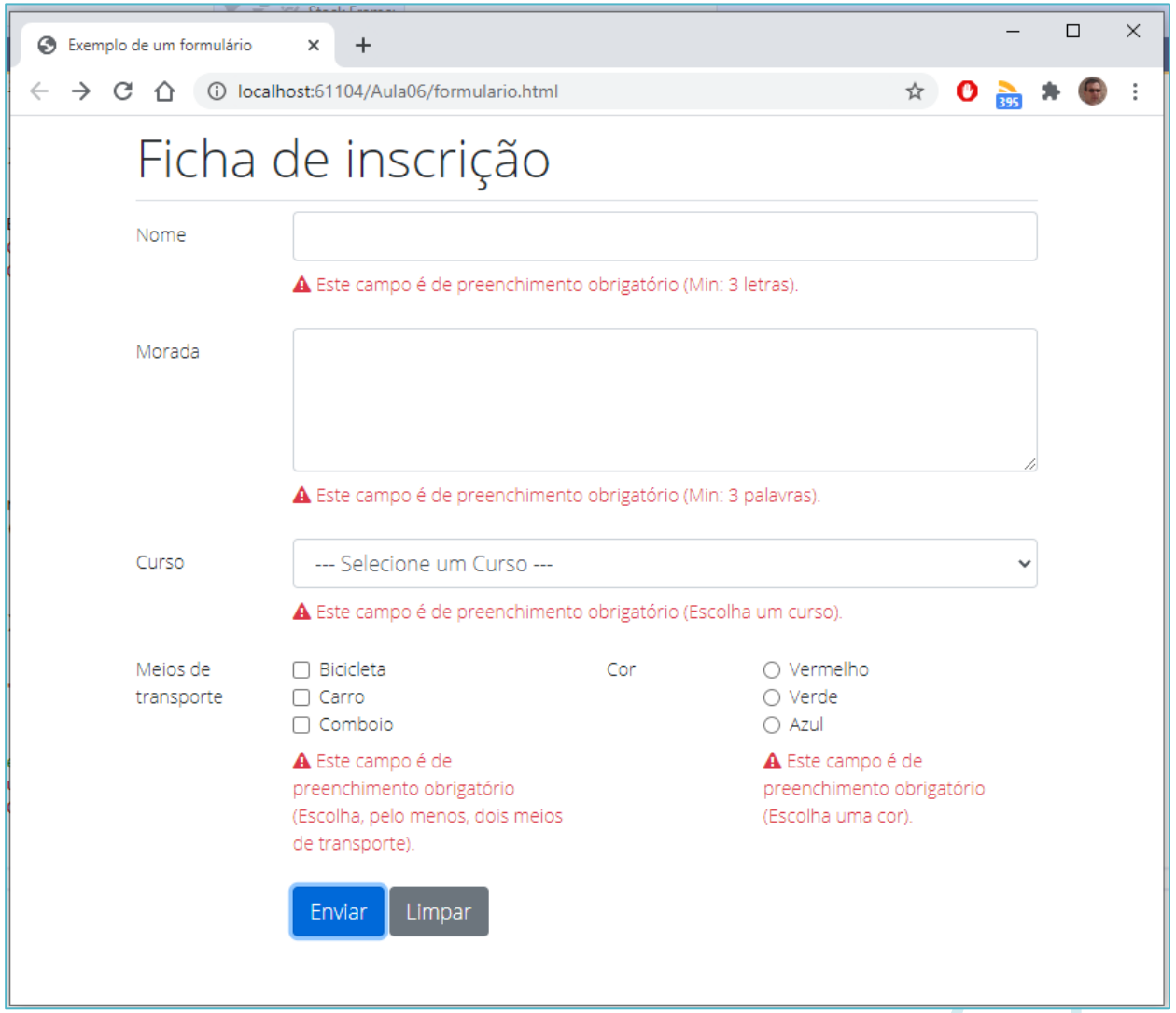

©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

Erros assinalados na interface

A validação utilizada implica que para cada falha seja apresentada uma mensagem de alert.

- Esta não é a forma mais adequada. A forma mais adequada é assinalar na interface os erros.



©2014-20, JOAQUIM SOUSA PINTO

Validação do formulário

Erros assinalados na interface – o código html

```
<form action="http://192.168.160.36/FormEcho.aspx" name="FichaInscricao">
    <div class="row form-group">
        <label for="Nome" class="col-md-2 col-form-label">Nome</label>
        <div class="col-md-10"><input type="text" name="Nome" id="Nome" class="form-control" />
        <div class="d-none text-danger col-form-label" id="NomeError"><i class="fa fa-exclamation-triangle"></i> Este campo é
de preenchimento obrigatório (Min: 3 letras).</div>
        </div>
    </div>
    <div class="row form-group">
        <label for="Morada" class="col-md-2 col-form-label">Morada</label>
        <div class="col-md-10">
            <textarea name="Morada" id="Morada" class="form-control" rows="4"></textarea>
            <div class="d-none text-danger col-form-label" id="MoradaError"><i class="fa fa-exclamation-triangle"></i> Este
campo é de preenchimento obrigatório (Min: 3 palavras).</div>
        </div>
    </div>
(…)
</form>
```


©2014-20, JOAQUIM SOUSA PINTO

```
<script>
    /* Função de validação  */
    function validate() {
        var retVal = true; /* Vamos partir do princípio de que o formulário está válido ... */
        var _nome = document.getElementById("Nome");
        var _nomeError = document.getElementById("NomeError");
        if (_nome.value.trim().length < 3) {
            retVal = false;
            _nomeError.classList.add("d-block");
            _nomeError.classList.remove("d_none");
        }
        else {
            _nomeError.classList.remove("d-block");
            _nomeError.classList.add("d_none");
        }
        var _morada = document.getElementById("Morada");
        var _moradaError = document.getElementById("MoradaError");
        var palavrasArray = _morada.value.split(' '); /* Partir a Morada em palavras */
        if (palavrasArray.length < 3) { /* Contar o número de palavras */
            retVal = false;
            _moradaError.classList.add("d-block");
            _moradaError.classList.remove("d_none");
        }
        else {
            _moradaError.classList.remove("d-block");
            _moradaError.classList.add("d_none");
        }
(…)
    return retVal;
    }
</script>
```


©2014-20, JOAQUIM SOUSA PINTO

Propriedades e métodos utilizados

`Element.classList` - O `Element.classList` é uma propriedade somente leitura que retorna uma coleção `DOMTokenList` dos atributos de classe do elemento. Isso pode ser usado para manipular a lista de classes.

- `tokenList.add(token1[, token2[, ...tokenN]])` - O método `add()` da interface `DOMTokenList` adiciona o(s) token(s) fornecido(s) à lista;
- `tokenList.remove(token1[, token2[, ...tokenN]])` - O método `remove()` da interface `DOMTokenList` remove o(s) token(s) fornecido(s) à lista;
- `tokenList.contains(token)` - O método `contains()` da interface `DOMTokenList` retorna um Booleano - verdadeiro se a lista contém o token fornecido; caso contrário, é falso.

Exemplo:

```
_moradaError.classList.remove("d-block");
_moradaError.classList.add("d_none");
```


©2014-20, JOAQUIM SOUSA PINTO

Javascript - Temporizadores

©2014-20, JOAQUIM SOUSA PINTO

Javascript para fazer animações

Uma das vantagens da utilização da linguagem javascript em páginas html é permitir a criação de animações e aumentar a interatividade.

Exemplo:

Considere o seguinte código (que faz um elemento diminuir de altura até desaparecer):

```
function diminuirVertical(elemento) {
    var altura = parseInt(elemento.style.height, 10);
    for (; altura > 0; altura--) {
        elemento.style.height = altura + "px";
    }
}
```

Problema deste exemplo: Como não há forma de controlar o tempo de execução, quanto mais rápido for o computador, mais rápido o texto desaparecerá …

©2014-20, JOAQUIM SOUSA PINTO

Sintaxe da linguagem Javascript
Temporizadores

Para controlar o tempo de execução do código, fazendo com que o comportamento seja igual em todos os computadores a linguagem javascript permite a utilização de **temporizadores**, que são “relógios programáveis” com uma resolução (diferença entre dois estados) de 1 milisegundo.

- É assim possível ativar funções de forma periódica, sendo igualmente possível controlar o intervalo entre execuções.

No caso de animações, é possível controlar a duração da animação.

©2014-20, JOAQUIM SOUSA PINTO

Sintaxe da linguagem Javascript

Temporizadores - Funções relevantes

`setInterval("função", intervalo)` - Define qual a **`função`** a ser invocada, de modo repetitivo, a cada **`intervalo`** de tempo.

O intervalo de tempo é expresso em milisegundos.

A função devolve um objeto para que seja possível cancelar o temporizador;

`clearInterval(temporizador)` - Apaga o **`temporizador`** passado no argumento;

`setTimeout("função", atraso)` - Define qual a `função` a ser invocada depois do **`atraso`** especificado, em milisegundos.

Neste caso, a função é executada apenas uma vez.

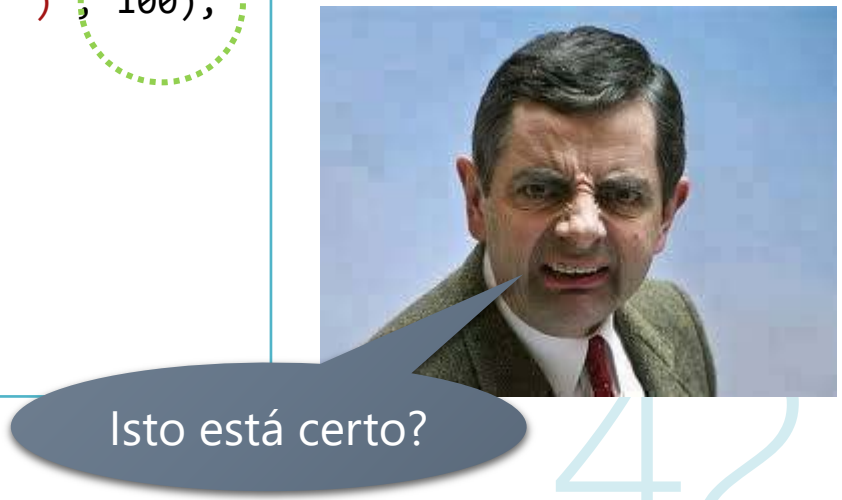
©2014-20, JOAQUIM SOUSA PINTO

Temporizadores
Exemplo

Neste exemplo altera-se a altura de uma imagem em 10px de cada vez que o processo é executado – a cada 100ms.

A função além de reduzir a imagem, deteta se a altura da imagem é zero e cancela o temporizador nessa altura (evitando tamanhos negativos).

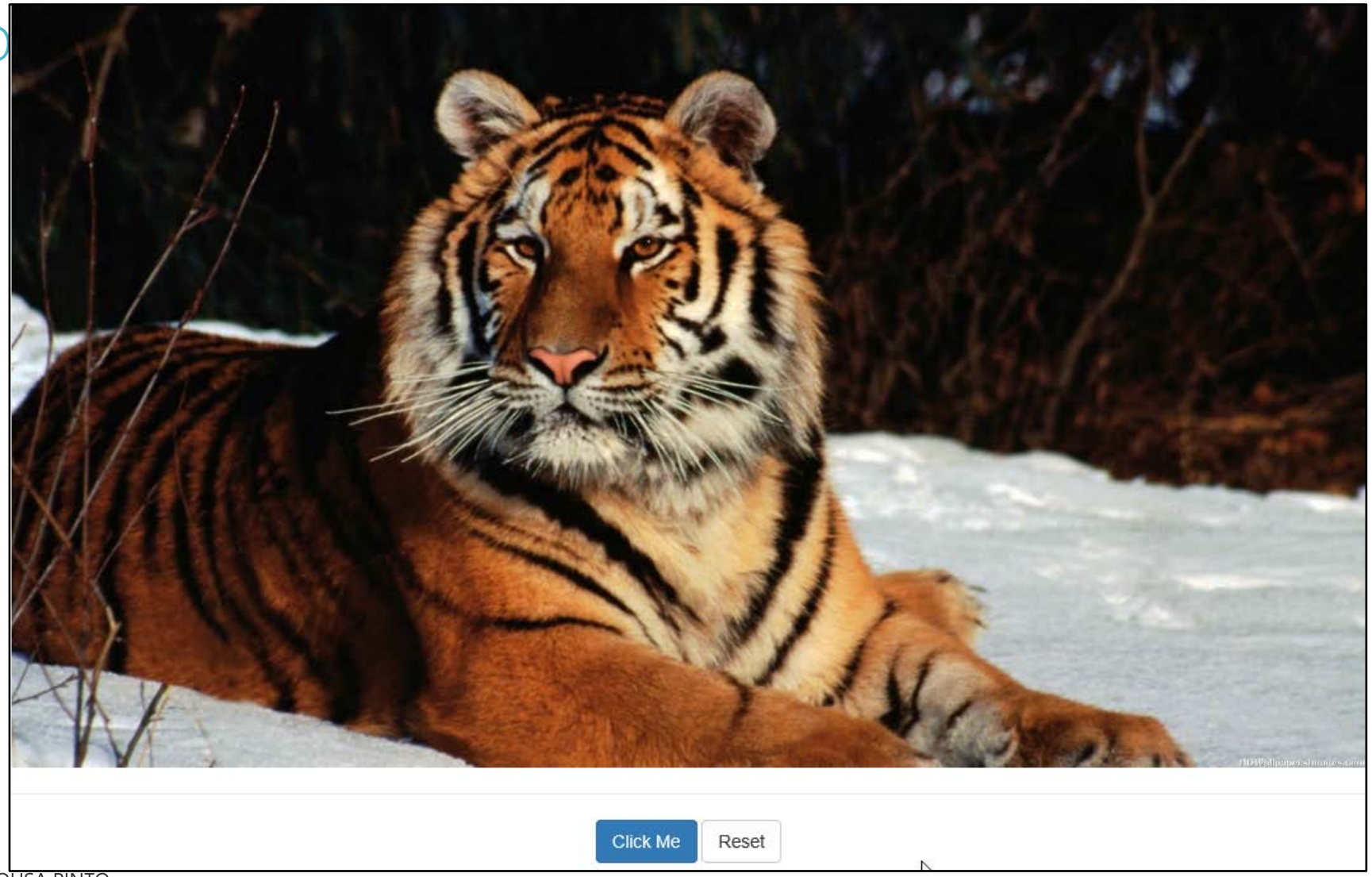
```
var temporizador = null;
function diminuirVertical(elemento) {
    if (temporizador == null) {
        temporizador = setInterval("diminuirVertical(" + elemento.id + ")", 100);
    }
    var altura = parseInt(elemento.style.height) - 10;
    elemento.style.height = altura + "px";
    if (altura == 0) {
        window.clearInterval(temporizador);
        temporizador = null;
    }
}
```



©2014-20, JOAQUIM SOUSA PINTO

Temporizadores

Exemplo



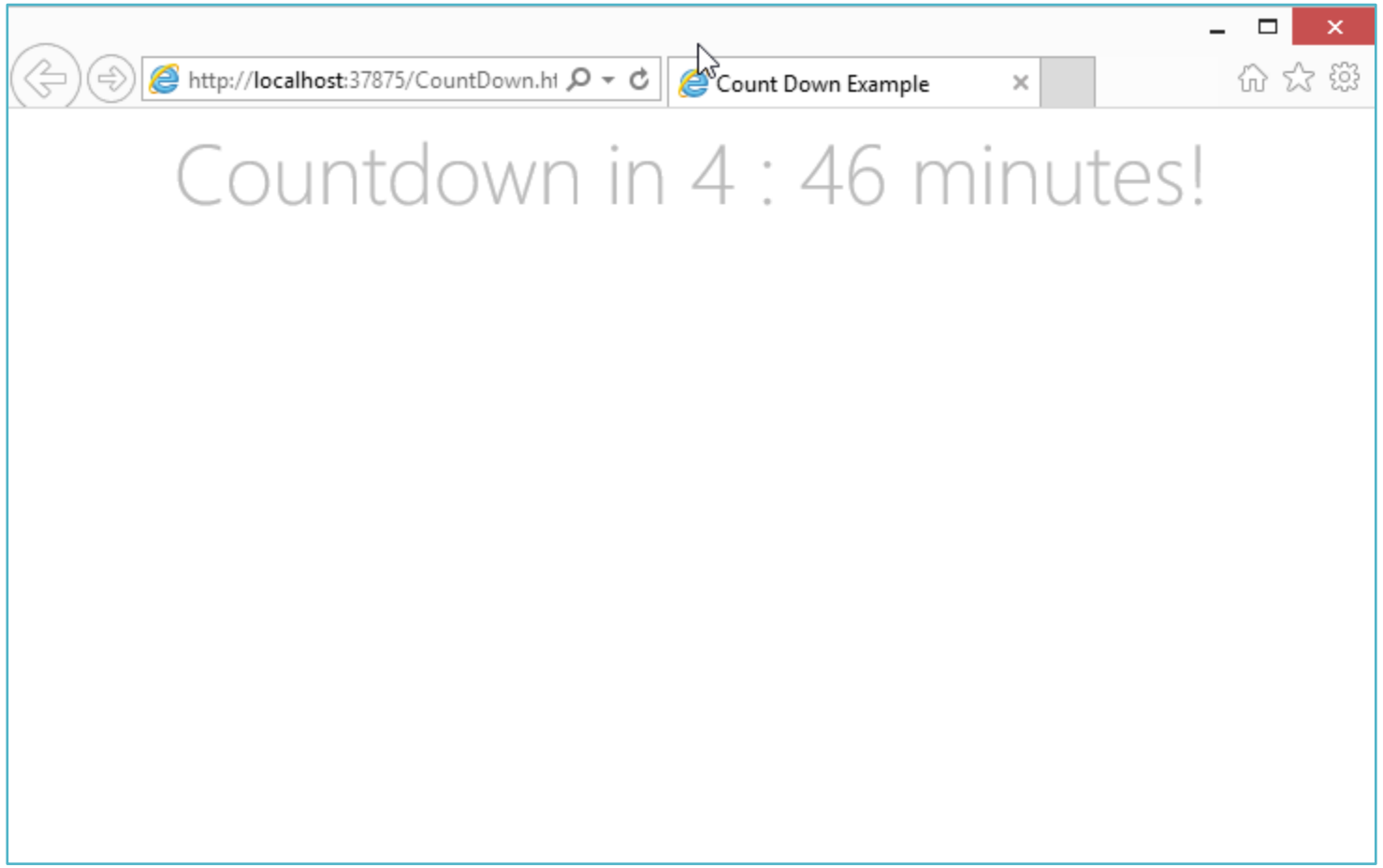
©2014-20, JOAQUIM SOUSA PINTO

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Exemplo com temporizadores - Imagem resize</title>
    <link href="../Content/bootstrap.min.css" rel="stylesheet" />
    <link href="../Content/bootstrap-theme.min.css" rel="stylesheet" />
    <script type="text/javascript">
        var temporizador = null;

        function diminuirVertical(elemento) {
            if (temporizador == null) {
                temporizador = setInterval("diminuirVertical(" + elemento.id + ")", 100);
            }
            var altura = parseInt(elemento.style.height) - 10;
            elemento.style.height = altura + "px";
            if (altura <= 0) {
                elemento.style.display = "none";
                window.clearInterval(temporizador);
                temporizador = null;
            }
        }
        function resetImage(elemento) {
            elemento.style.display = "block";
            elemento.style.height = "600px";
        }
    </script>
</head>
<body>
    <div class="container">
        <img id="myImage" src="https://tinyurl.com/y7cypkom" style="height:600px" class="center-block" />
    </div>
    <div class="row"><hr /></div>
    <div class="row text-center">
        <input type="button" onclick="diminuirVertical(document.getElementById('myImage'))" value="Click Me" class="btn btn-primary" />
        <input type="reset" value="Reset" class="btn btn-default" onclick="resetImage(document.getElementById('myImage'))" />
    </div>
</body>
</html>
```

Temporizadores

Exemplo 2 – Countdown - interface



©2014-20, JOAQUIM SOUSA PINTO

Temporizadores

Exemplo 2 – o código

Isto funciona?

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Count Down Example</title>
    <link href="../Content/bootstrap.min.css" rel="stylesheet" />
    <link href="../Content/bootstrap-theme.min.css" rel="stylesheet" />
    <style type="text/css">
        #theMessage { font-family: 'Segoe UI Light'; font-size: 64px; color: #bbb; }
    </style>
    <script>
        window.onload = function () {
            var min = 5;
            var sec = 0;
            setInterval(function () {
                document.getElementById("timer").innerHTML = min + " : " + sec;
                if (sec == 0) {
                    sec = 60;
                    min--;
                    if (min == -1) {  //--- aqui acaba
                        min = 4;
                    }
                }
                sec--;
            }, 50); // normal: 1000 | Colocar em 50 para testar…
        }
    </script>
</head>
<body>
    <div id="theMessage" class="text-center">Countdown in <span id="timer">05:00</span> minutes!</div>
</body>
</html>
```


©2014-20, JOAQUIM SOUSA PINTO

Para a semana temos teste teórico prático

©2014-20, JOAQUIM SOUSA PINTO

O que sai num teste teórico-prático?

Sai tudo o que está nos acetatos! (tudo, é tudo - sem exceção), história, http, html, css, etc, etc, etc.

Concretizando:

- "Pode sair" um exercício em que é apresentado o aspeto final e é necessário fazer o código html e o css;
- "Pode sair" um exercício com um formulário e é necessário escrever a(s) função(ões) javascript – neste caso, o html será fornecido e terão "apenas" de escrever o código relativo às funções já identificadas.

©2014-20, JOAQUIM SOUSA PINTO

E não temos mais nenhuma aula teórico prática antes do teste prático

©2014-20, JOAQUIM SOUSA PINTO

Teste prático: dia 23 segunda-feira de tarde

Quem não tem aula prática nesse dia (turmas de terça e quarta) vai ter de escolher uma hora para realizar o teste (**em grupo**!)

14:00 às 16:00

Ou

16:30 às 18:30

Mas eu ainda não tenho grupo!

©2014-20, JOAQUIM SOUSA PINTO

Vamos jogar no totobola

Um dos testes práticos do ano 2016/17

©2014-20, JOAQUIM SOUSA PINTO

Como se joga?

O **Totobola** é um jogo de apostas mútuas desportivas, baseado em prognósticos para resultados de jogos de futebol.

São 13 jogos base em cada coluna, nos quais se prognostica o resultado de vitória "1", empate "X" ou derrota "2" da equipa visitada.

As apostas simples agrupam-se em pares de colunas e preenchem-se com a marcação de um só prognóstico por jogo.

©2014-20, JOAQUIM SOUSA PINTO

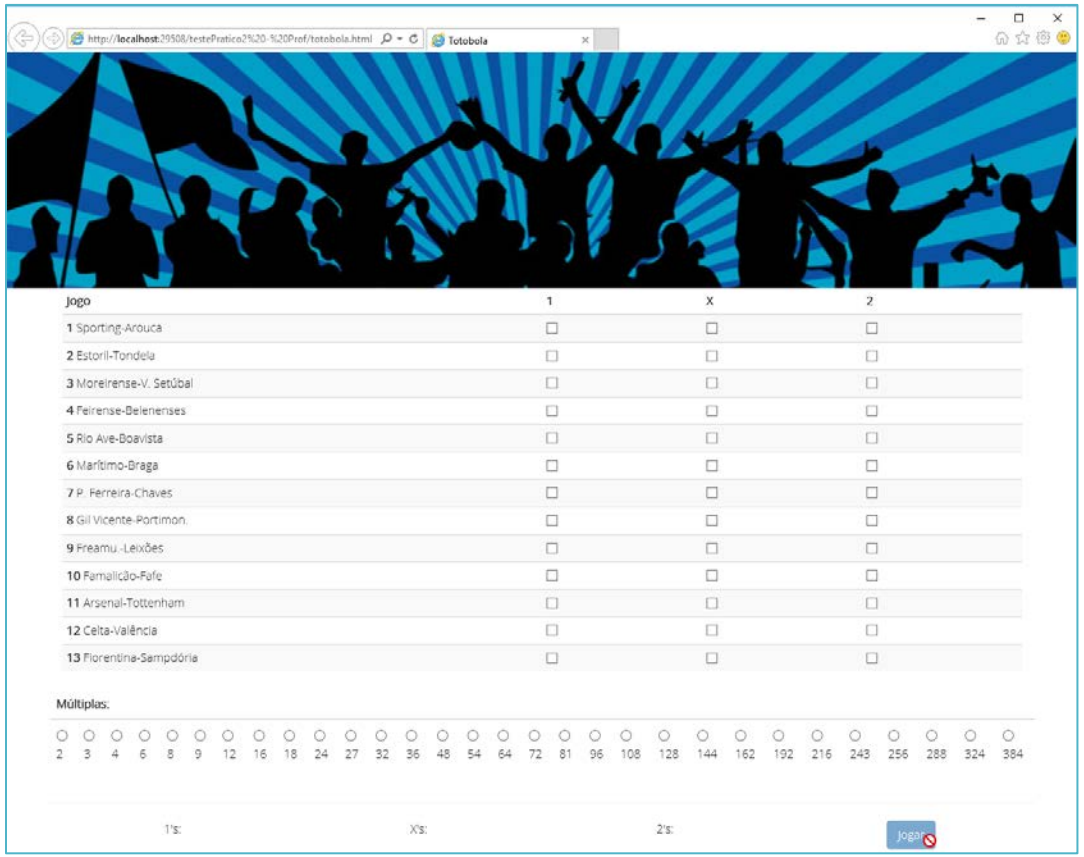
Exemplos de apostas

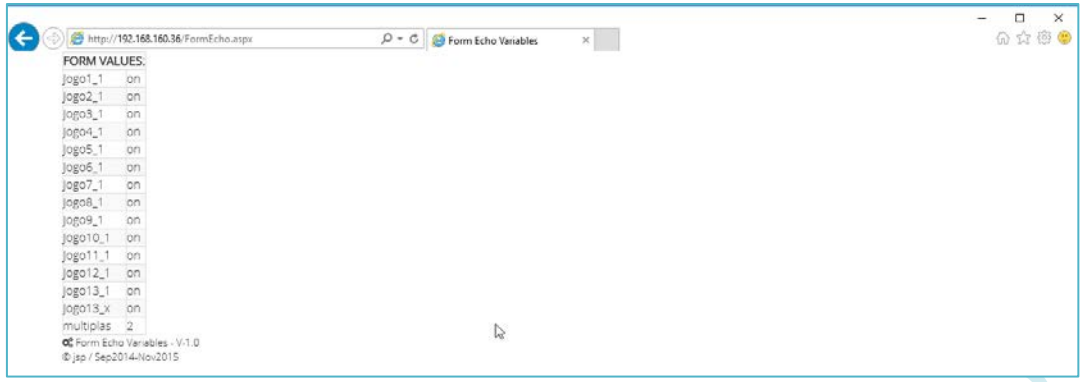
Apostas simples

Apostas múltiplas

©2014-20, JOAQUIM SOUSA PINTO

Problema do teste prático 2016/17…





©2014-20, JOAQUIM SOUSA PINTO

File Edit View Website Build Debug Team Tools Test Analyze Window Help Full Screen

Quick Launch (Ctrl+Q) Joaquim Pinto JP

totobola.js | totobola.html | myScript.js | dom.js | HtmlPage.html | index.html | lampada.html | calculadora.html

```
3   <head>...</head>
13  <body>
14      <div class="bannerTop"></div>
15      <div class="">
16          <form action="" method="post">
17              <table class="">
18                  <thead>
19                      <tr>...</tr>
25                  </thead>
26                  <tbody>
27                      <tr>
28                          <td><strong>1</strong> Sporting-Arouca</td>
29                          <td><input type="checkbox" ID="Jogo1_1" name="Jogo1_1" onclick="boxClicked()"></td>
30                          <td><input type="checkbox" ID="Jogo1_x" name="Jogo1_x" onclick="boxClicked()"></td>
31                          <td><input type="checkbox" ID="Jogo1_2" name="Jogo1_2" onclick="boxClicked()"></td>
32                      </tr>
33                      <!-- ... -->
106                 </tbody>
107             </table>
108             <div class="">
109                 <table class="">
110                     <thead>
111                     <tr><th colspan="30">Múltiplas:</th></tr>
112                     </thead>
113                     <tbody>
114                         <tr>
115                             <td><input type="radio" ID="multiplas2" name="multiplas" value="2" /><br />2</td>
116                             <td><input type="radio" ID="multiplas3" name="multiplas" value="3" /><br />3</td>
117                             <td><input type="radio" ID="multiplas4" name="multiplas" value="4" /><br />4</td>
118                             <td><input type="radio" ID="multiplas6" name="multiplas" value="6" /><br />6</td>
119                             <!-- ... -->
145                         </tr>
146                     </tbody>
147                 </table>
148             </div>
149             <div class="row">
150                 <hr />
151                 <div class="">1's: <span ID="Status1"></span></div>
152                 <div class="">X's: <span ID="StatusX"></span></div>
153                 <div class="">2's: <span ID="Status2"></span></div>
154                 <div class=""><input type="submit" class="" id="botao" value="Jogar" disabled="disabled" /></div>
155             </div>
156         </form>
157     </div>
158     <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
```

100 %

Ready Ln 111 Col 62 Ch 62 INS Add to Source Control

```javascript
//--- variáveis globais
var Status1 = document.getElementById("Status1");
var StatusX = document.getElementById("StatusX");
var Status2 = document.getElementById("Status2");
var botao = document.getElementById("botao");

var linhas = 0;
var apostas = 1;
var selected1 = 0;
var selectedX = 0;
var selected2 = 0;
//--- função ativada sempre que se carrega em qualquer das caixas
function boxClicked() {
    linhas = 0;
    apostas = 1;
    selected1 = 0;
    selectedX = 0;
    selected2 = 0;
    for (var i = 1; i <= 13; i++) {
        var apostasNaLinha = 1;
        var x = document.getElementById("Jogo" + i + "_1").checked;
        var y = document.getElementById("Jogo" + i + "_x").checked;
        var z = document.getElementById("Jogo" + i + "_2").checked;
        if (x) selected1++;
        if (y) selectedX++;
        if (z) selected2++;
        if (x | y | z) linhas++;
        if ((x && y) || (x && z) || (y && z)) apostasNaLinha = 2;
        if (x && y && z) apostasNaLinha = 3;
        apostas *= apostasNaLinha;
    }
    console.log(apostas);
    setStatus();
}
```

```javascript
function setStatus() {
    Status1.innerText = selected1;
    StatusX.innerText = selectedX;
    Status2.innerText = selected2;
    if (linhas == 13 && apostas > 1 && apostas <= 384) {
        var x = document.getElementById("multiplas" + apostas)
        if (x != null)
            x.checked = true;
        botao.disabled = "";
    }
    else {
        var arr = document.getElementsByName("multiplas");
        for (var i = 0; i < arr.length; i++) {
            arr[i].checked = false;
        }
        botao.disabled = "disabled";
    }
}
```

boxClicked() – conta as linhas, conta os 1, X e 2 e calcula o n.º das apostas;

setStatus() – coloca os valores nas caixas respetivas e desliga o botão e jogar quando o número de postas é inválido ou quando não estão as linhas todas preenchidas.

©2014-20, JOAQUIM SOUSA PINTO