

# Introdução às tecnologias Web - ITW

## Aula 7 – Javascript

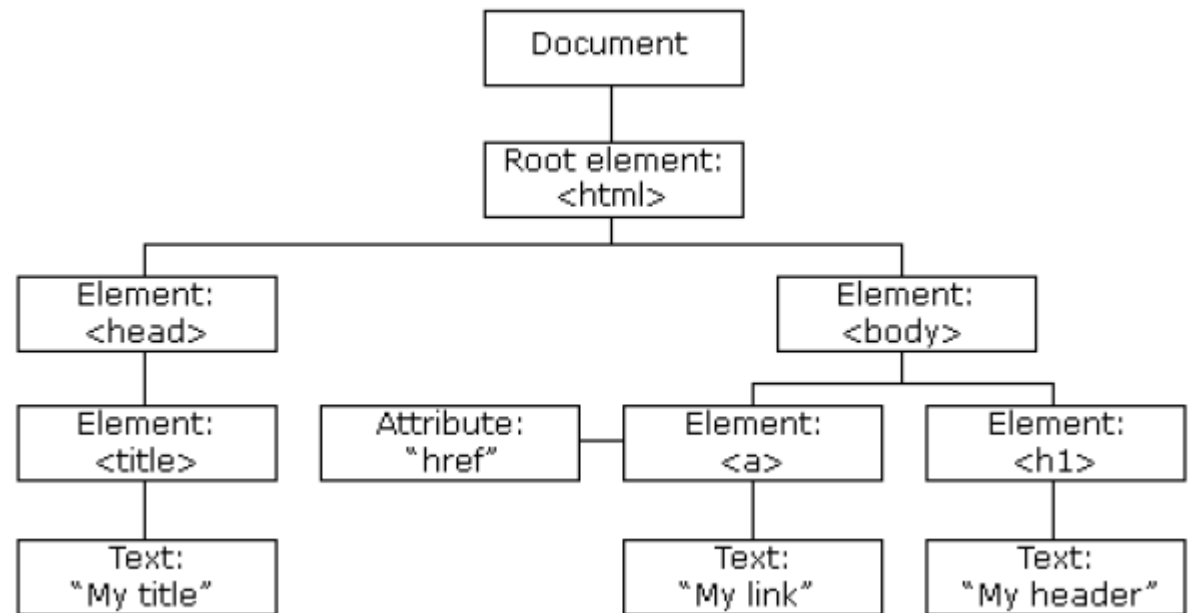
# Sumário

A linguagem Javascript  
Interacção com o DOM (revisitado)  
Temporizadores



# Revisão - Estrutura DOM de uma página html

```
<!doctype html>
<html>
<head>
  <title>My title</title>
</head>
<body>
  <h1>My header</h1>
  <a href="http://www.ua.pt">My link</a>
</body>
</html>
```



# Revisão de formulários html

Tal como foi apresentado em na aula 2, para que qualquer elemento de um formulário `<input></input>`, `<textarea></textarea>`, ... possa ser enviado para um servidor, tem de possuir um nome (atributo `name`)

```
<input type="text" name="FirstName" value="Joaquim" placeholder="Nome" />  
<input type="text" name="LastName" placeholder="Sobrenome" />
```

# Exemplo da aula 2

(com a inclusão dos estilos bootstrap)

```
1  <!DOCTYPE html>
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4      <title>Exemplod de um formulário com "names"</title>
5      <link href="Content/bootstrap.css" rel="stylesheet" />
6  </head>
7  <body>
8      <form name="PersonData" action="http://192.168.160.36/FormEcho.aspx" method="post">
9          <div class="page-header"><h1>Personal info</h1></div>
10         <input type="hidden" name="FormName" value="UserForm" />
11         <div class="col-xs-2 form-control-static text-right">First Name:</div>
12         <div class="col-xs-10"><input type="text" name="FirstName" placeholder="Digite o seu nome" class="form-control" /></div>
13         <div class="col-xs-2 form-control-static text-right">Last Name:</div>
14         <div class="col-xs-10"><input type="text" name="LastName" placeholder="Digite o seu sobrenome" class="form-control" /></div>
15         <div class="col-xs-2 form-control-static text-right">Password:</div>
16         <div class="col-xs-10"><input type="text" name="Password" placeholder="Digite a sua palavra passe" class="form-control" /></div>
17         <div class="col-xs-2 form-control-static text-right">Comments:</div>
18         <div class="col-xs-10"><textarea name="Comments" rows="4" cols="80" placeholder="Insira os seus comentários" class="form-control"></textarea></div>
19         <div class="col-xs-12 text-center"><hr /></div>
20         <div class="col-xs-12 text-center"><input name="Button" type="Submit" class="btn btn-primary" value="Submit data" /></div>
21     </form>
22     <script src="Scripts/jquery-2.1.4.min.js"></script>
23     <script src="Scripts/bootstrap.min.js"></script>
24
25 </body>
26 </html>
27
```

# Exemplo da aula 2

(com a inclusão dos estilos bootstrap)

The image shows two side-by-side browser windows. The left window displays a web form titled 'Personal info' with fields for First Name, Last Name, Password, and Comments, and a 'Submit data' button. A blue dotted arrow points from the button to the right window. The right window displays the 'Form Echo Variables' page, which shows the values entered in the form: FormName (UserForm), FirstName, LastName, Password, Comments, and Button (Submit data). The page also includes the text 'Form Echo Variables - V-1.0' and '© jsp / Sep2014'.

**Personal info**

First Name: Digite o seu nome

Last Name: Digite o seu sobrenome

Password: Digite a sua palavra passe

Comments: Insira os seus comentários

Submit data

**FORM VALUES:**

FormName	UserForm
FirstName	
LastName	
Password	
Comments	
Button	Submit data

Form Echo Variables - V-1.0  
© jsp / Sep2014

# Problema

Os dados foram submetidos ao servidor sem terem sido preenchidos!

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

# Revisão de formulários html

(agora em conjunto com javascript)

## Problema:

1. os javascript refere-se aos elementos pelo atributo `ID` e não pelo `name`
2. como converter números em texto?

Ficheiro dom.js

```
function calculadora(){
    var x = document.getElementById("op1");
    var y = document.getElementById("op2");
    console.log( parseFloat(x.value) );
    console.log( parseFloat(y.value) );
}
window.onload = calculadora;
```

Ficheiro dom.js

```
function calculadora(){
    var x = document.getElementById("op1");
    var y = document.getElementById("op2");
    console.log(x.value * 1.0 );
    console.log(y.value * 1.0 );
}
window.onload = calculadora;
```



# Revisão de formulários html

(agora em conjunto com javascript)

**Solução (1):** utilizar o atributo `ID` para identificar os elementos em vez do atributo `name`

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <title>Exemplod de um formulário com "id's"</title>
5   <link href="Content/bootstrap.css" rel="stylesheet" />
6 </head>
7 <body>
8   <form id="PersonData" action="http://192.168.160.36/FormEcho.aspx" method="post">
9     <div class="page-header"><h1>Personal info</h1></div>
10    <input type="hidden" id="FormName" value="UserForm" />
11    <div class="col-xs-2 form-control-static text-right">First Name:</div>
12    <div class="col-xs-10"><input type="text" id="FirstName" placeholder="Digite o seu nome" class="form-control" /></div>
13    <div class="col-xs-2 form-control-static text-right">Last Name:</div>
14    <div class="col-xs-10"><input type="text" id="LastName" placeholder="Digite o seu sobrenome" class="form-control" /></div>
15    <div class="col-xs-2 form-control-static text-right">Password:</div>
16    <div class="col-xs-10"><input type="text" id="Password" placeholder="Digite a sua palavra passe" class="form-control" /></div>
17    <div class="col-xs-2 form-control-static text-right">Comments:</div>
18    <div class="col-xs-10"><textarea id="Comments" rows="4" cols="80" placeholder="Insira os seus comentários" class="form-control"></textarea></div>
19    <div class="col-xs-12 text-center"><hr /></div>
20    <div class="col-xs-12 text-center"><input id="Button" type="Submit" class="btn btn-primary" value="Submit data" /></div>
21  </form>
22  <script src="Scripts/jquery-2.1.4.min.js"></script>
23  <script src="Scripts/bootstrap.min.js"></script>
24
25 </body>
26 </html>
```

07/11/2016

id= I  
Aa Abi \* Current D

Form Echo Variables - V-1.0  
© jsp / Sep2014

Não funciona!!!!!!!

# Revisão de formulários html

(agora em conjunto com javascript)

**Solução (2):** Utilização simultânea dos atributos `name` e `ID` dos elementos a enviar para o servidor

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <title>Exemplo de um formulário com "names" &Aacute; "ids"</title>
5   <link href="Content/bootstrap.css" rel="stylesheet" />
6 </head>
7 <body>
8   <form name="PersonData" action="http://192.168.160.36/FormEcho.aspx" method="post">
9     <div class="page-header"><h1>Personal info</h1></div>
10    <input type="hidden" name="FormName" id="PersonData_FormName" value="UserForm" />
11    <div class="col-xs-2 form-control-static text-right">First Name:</div>
12    <div class="col-xs-10"><input type="text" name="FirstName" id="PersonData_FirstName" placeholder="Digite o seu nome" class="form-control" /></div>
13    <div class="col-xs-2 form-control-static text-right">Last Name:</div>
14    <div class="col-xs-10"><input type="text" name="LastName" id="PersonData_LastName" placeholder="Digite o seu sobrenome" class="form-control" /></div>
15    <div class="col-xs-2 form-control-static text-right">Password:</div>
16    <div class="col-xs-10"><input type="text" name="LastName" id="PersonData_LastName" placeholder="Digite a sua palavra passe" class="form-control" /></div>
17    <div class="col-xs-2 form-control-static text-right">Comments:</div>
18    <div class="col-xs-10"><textarea name="Comments" id="PersonData_Comments" rows="4" cols="80" placeholder="Insira os seus comentários" class="form-control"></textarea>
19    <div class="col-xs-12 text-center"><hr /></div>
20    <div class="col-xs-12 text-center"><input name="Button" type="Submit" class="btn btn-primary" value="Submit data" /></div>
21  </form>
22  <script src="Scripts/jquery-2.1.4.min.js"></script>
23  <script src="Scripts/bootstrap.min.js"></script>
24
25 </body>
26 </html>
```

07/11/2016

# Validação de formulários

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

Passo 1:

Vamos, para começar garantir que `"FirstName"` / `"PersonalData_FirstName"` tem, no mínimo, 3 letras (Ex: Ivo, Ana, ...)

# Validação de formulários

## Passo 1 – a interface

The image shows two browser windows. The left window displays a web form titled 'Personal info' with the following fields:

- First Name: ana
- Last Name: Digite o seu sobrenome
- Password: Digite a sua palavra passe
- Comments: Insira os seus comentários

A blue 'Submit data' button is located at the bottom of the form. A dashed blue arrow points from this button to the right window.

The right window displays the 'Form Echo Variables' page, which shows the submitted data in a table:

FORM VALUES:	
FormName	UserForm
FirstName	ana
LastName	
Comments	
Button	Submit data

Below the table, the text 'Form Echo Variables - V-1.0' and '© jsp / Sep2014' is visible.

# Validação de formulários

## Passo 1 – o código

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <title>Exemplo de validação de um formulário</title>
5   <link href="Content/bootstrap.css" rel="stylesheet" />
6   <script type="text/javascript">
7     function testFormValid() {
8       var retVal = true;
9       var PersonData_FirstName = document.getElementById("PersonData_FirstName");
10      if (PersonData_FirstName.value.trim().length < 3)
11        retVal = false;
12      return retVal;
13    }
14  </script>
15 </head>
16 <body>
17   <form name="PersonData" action="http://192.168.160.36/FormEcho.aspx" method="post">
18     <div class="page-header"><h1>Personal info</h1></div>
19     <input type="hidden" name="FormName" id="PersonData_FormName" value="UserForm" />
20     <div class="col-xs-2 form-control-static text-right">First Name:</div>
21     <div class="col-xs-10"><input type="text" name="FirstName" id="PersonData_FirstName" placeholder="Digite o seu nome" class="form-control" /></div>
22     <!-- ... -->
29     <div class="col-xs-12 text-center"><input name="Button" type="Submit" class="btn btn-primary" value="Submit data" onclick="return testFormValid()" /></div>
30   </form>
31   <script src="Scripts/jquery-2.1.4.min.js"></script>
32   <script src="Scripts/bootstrap.min.js"></script>
33
34 </body>
35 </html>
36
```

# Validação de formulários

## Passo 2 – a interface

Em caso de erro, mostra uma mensagem explicativa de modo a avisar o utilizador...

The image displays two browser windows side-by-side, illustrating a form validation process. Both windows show a web page titled 'Personal info' with a form containing four fields: 'First Name', 'Last Name', 'Password', and 'Comments'. The 'First Name' field has a placeholder text 'Digite o seu nome'. Below the form are two buttons: 'Clear data' and 'Submit data'. The left window shows the form before submission, with a status bar at the bottom displaying 'http://192.168.160.36/FormEcho.aspx'. The right window shows the form after submission, with a red error message 'The First Name should have, at least, 3 chars!' displayed below the 'First Name' field. A blue dashed arrow points from the 'First Name' field in the left window to the error message in the right window. The status bar in the right window also displays 'http://192.168.160.36/FormEcho.aspx'.

# Validação de formulários

## Passo 2 – o código html

```
33 <body>
34 <form name="PersonData" action="http://192.168.160.36/FormEcho.aspx" method="post">
35   <div class="page-header"><h1>Personal info</h1></div>
36   <input type="hidden" name="FormName" id="PersonData_FormName" value="UserForm" />
37   <div class="col-xs-2 form-control-static text-right">First Name:</div>
38   <div class="col-xs-10">
39     <input type="text" name="FirstName" id="PersonData_FirstName" placeholder="Digite o seu nome" class="form-control" />
40     <div id="PersonData_FirstName_Error" class="no_error form-control-static text-danger bg-danger">
41       &nbsp;<i class="fa fa-exclamation-triangle"></i> The First Name should have, at least, 3 chars!&nbsp;</div>
42   </div>
43   <div class="col-xs-2 form-control-static text-right">Last Name:</div>
44   <div class="col-xs-10"><input type="text" name="LastName" id="PersonData_LastName" placeholder="Digite o seu sobrenome" class="form-control" /></div>
45   <div class="col-xs-2 form-control-static text-right">Password:</div>
46   <div class="col-xs-10"><input type="text" name="LastName" id="PersonData_LastName" placeholder="Digite a sua palavra passe" class="form-control" /></div>
47   <div class="col-xs-2 form-control-static text-right">Comments:</div>
48   <div class="col-xs-10"><textarea name="Comments" id="PersonData_Comments" rows="4" cols="80" placeholder="Insira os seus comentários" class="form-control"></textarea>
49   <div class="col-xs-12 text-center"><hr /></div>
50   <div class="col-xs-12 text-center">
51     <input name="Button" type="reset" class="btn btn-default" value="Clear data" />
52     <input name="Button" type="Submit" class="btn btn-primary" value="Submit data" onclick="return testFormValid()" />
53   </div>
54 </form>
55 <script src="Scripts/jquery-2.1.4.min.js"></script>
56 <script src="Scripts/bootstrap.min.js"></script>
57
58 </body>
59 </html>
```

# Validação de formulários

## Passo 2 – o código javascript & css

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <title>Exemplo de validação de um formulário</title>
5   <link href="Content/bootstrap.css" rel="stylesheet" />
6   <link href="Content/font-awesome.min.css" rel="stylesheet" />
7   <script type="text/javascript">
8     function testFormValid() {
9       var retVal = true;
10      var PersonData_FirstName = document.getElementById("PersonData_FirstName");
11      var PersonData_FirstName_Error = document.getElementById("PersonData_FirstName_Error");
12      if (PersonData_FirstName.value.trim().length < 3) {
13        if (PersonData_FirstName_Error.classList.contains("no_error"))
14          PersonData_FirstName_Error.classList.remove("no_error");
15        if (!PersonData_FirstName_Error.classList.contains("error"))
16          PersonData_FirstName_Error.classList.add("error");
17        retVal = false;
18      }
19      else {
20        if (PersonData_FirstName_Error.classList.contains("error"))
21          PersonData_FirstName_Error.classList.remove("error");
22        if (!PersonData_FirstName_Error.classList.contains("no_error"))
23          PersonData_FirstName_Error.classList.add("no_error");
24      }
25      return retVal;
26    }
27  </script>
28  <style type="text/css">
29    .no_error { display: none; }
30    .error { display: inline-block; }
31  </style>
32 </head>
```

- Propriedades úteis de uma string:
  - length: indica o tamanho da "string"
- Métodos úteis para manipulação de strings:
  - trim(): remove espaços nos extremos da "string":

A propriedade classList de um element retorna uma lista com os nomes das classes desse elemento.

- Métodos úteis para manipulação desta lista:
  - add("className") / contains("className") / remove("className")



# Sintaxe da linguagem Javascript

## Temporizadores

Visto que a linguagem JS é utilizada para aumentar a interactividade de páginas *Web*, uma das funcionalidades implementadas permite atrasar a execução das funções.

A possibilidade de controlar o tempo de execução é útil para implementar, por exemplo, animações e controlar a sua duração.

Por 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 desaparece ...

# Sintaxe da linguagem Javascript

## Temporizadores

A alternativa que a linguagem JS fornece é a utilização de **temporizadores**, 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.

# 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.

# Temporizadores

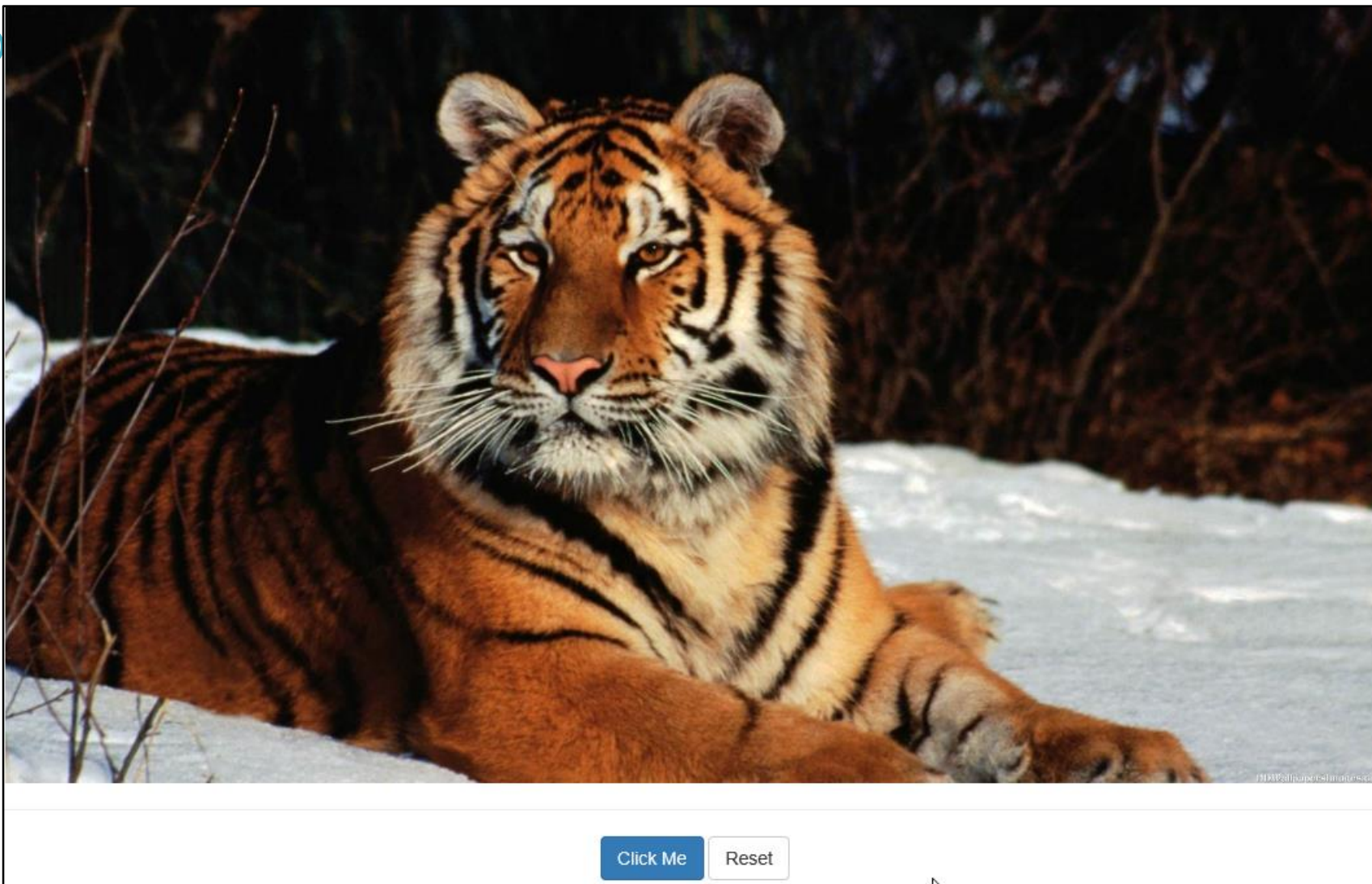
## Exemplo

Neste exemplo altera-se a altura de um elemento (uma imagem, por exemplo) em **10px** de cada vez. O processo é executado a cada **100ms**. A função além de reduzir a imagem, deteta através da variável **temp** que é a primeira execução e programa o temporizador. No final cancela-o.

```
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;
    }
}
```

# Temporizadores

## Exemplo



```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exemplo com temporizadores - Imagem resize</title>
  <link href="Content/bootstrap.min.css" rel="stylesheet" />
  <script type="text/javascript">
    var temporizador = null;

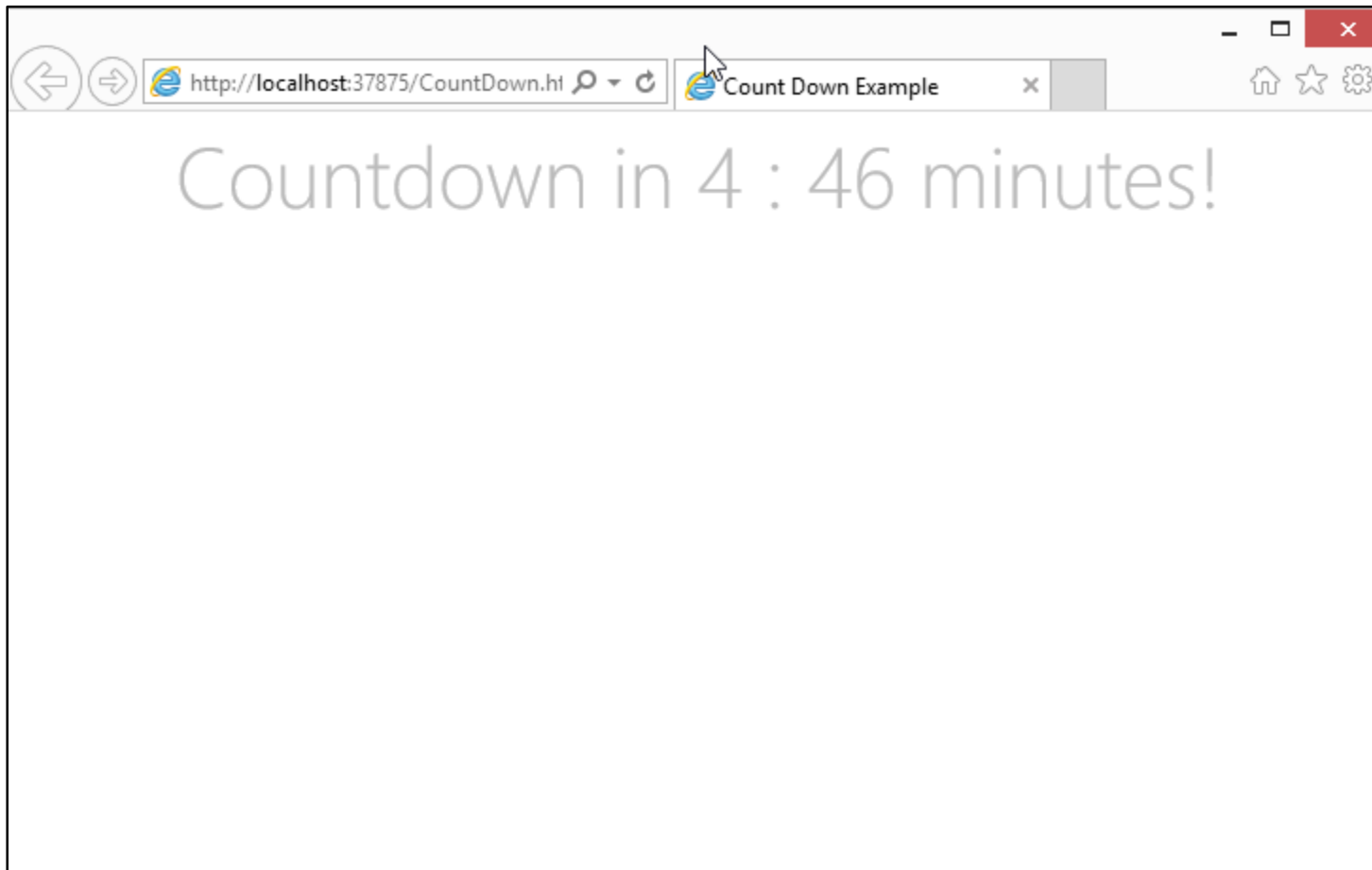
    function diminuirVertical(elemento) {
      if (temporizador == null) {
        temporizador = setInterval("diminuirVertical(" + elemento.id + ")", 100);
      }
      var altura = parseInt(elemento.style.height) - 10;
      elemento.style.height = altura + "px";
      if (altura <= 0) {
        elemento.style.display = "none";
        window.clearInterval(temporizador);
        temporizador = null;
      }
    }

    function resetImage(elemento) {
      elemento.style.display = "block";
      elemento.style.height = "600px";
    }
  </script>
</head>
<body>
  <div class="container">
    
  </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 - interface



# Temporizadores

## Exemplo – o código

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Count Down Example</title>
  <link href="Content/bootstrap.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--;
      }, 1000); // Colocar em 100 para testar...
    }
  </script>
</head>
<body>
  <div id="theMessage" class="text-center">Countdown in <span id="timer">05:00</span> minutes!</div>

</body>
</html>
```