



**BOSCH**

Invented for life

## Aula 6 - Javascript

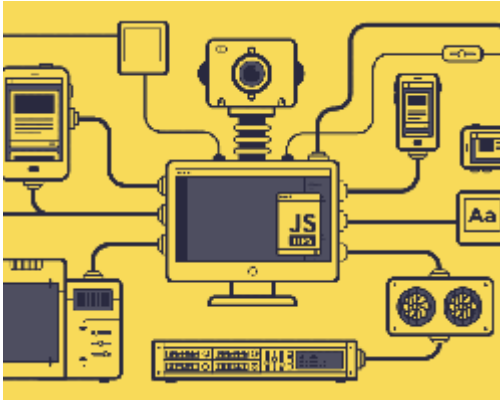
### Docupedia Export

Author: Siqueira Joao (CtP/ETS)  
Date: 21-Dec-2022 13:20

## Table of Contents

<b>1 Funções</b>	<b>4</b>
<b>2 Eventos de Temporização</b>	<b>5</b>
<b>3 JQuery</b>	<b>6</b>
<b>4 Desafio:</b>	<b>10</b>

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-  
xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script> <script  
type="text/javascript">
```



JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web. A definição mais correta e completa que posso proferir é que *JavaScript* é uma linguagem de programação leve, dinâmica e interpretada, que permite que o programador utilize tanto a programação imperativa quanto a procedural, ou mesmo a orientada a objetos. O JavaScript surgiu como uma linguagem exclusivamente para programação de aplicações Web, mais especificamente dando dinamismo às páginas no lado cliente (navegadores).

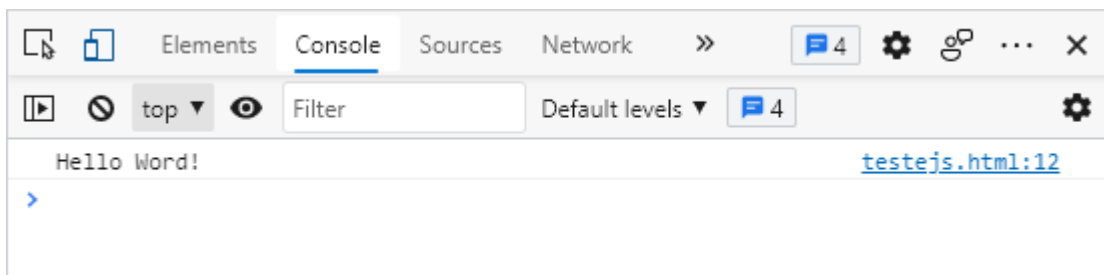
Todos os códigos JavaScript devem ser incorporados dentro da tag `<script>` no `<head>` de qualquer arquivo HTML.

# 1 Funções

Para criar uma função no JavaScript usamos "function":

```
1 <script type="text/javascript">
2     function escreverConsole(){
3         console.log("Hello World!");
4     }
5     escreverConsole();
6 </script>
```

Para abrir o console vá ao navegador e aperte a tecla "F12", depois clique em "Console" .



Podemos fazer também algumas operações utilizando funções com retorno:

```
1 <script type="text/javascript">
2     function converterTemperatura(celsius){
3         var fah = (celsius * 1.8)+32;
4         return fah;
5     }
6     var fahrenheit = converterTemperatura(30);
7     console.log(fahrenheit)
8 </script>
```

Assim temos 86, valor retornado pela função.

## 2 Eventos de Temporização

O Javascript pode ser executado em intervalos de tempo. Isso é chamado de eventos de temporização. Os dois métodos principais a serem usados são:

- `setTimeout(função, milissegundos)`- Executa uma função, depois de aguardar um número específico de tempo.
- `window.setTimeout(função, milissegundos);`

Pode ser gravado sem o prefixo `window`.

- `setInterval(função, milissegundos)` – o mesmo que `setTimeout()`, mas repete a execução da função continuamente.

Vamos criar um programa HTML que ao clicar no botão espera por 3 segundos e a página cria um alerta:

```
1 <body>
2   <button onclick="setTimeout(myfunction,3000)">Clique Aqui</button>
3   <script type="text/javascript">
4       function myfunction(){
5           alert("Alerta!!")
6       }
7   </script>
8 </body>
```

Para parar a execução devemos usar o método `clearTimeout()` interrompe a execução da função especificada em `setTimeout()`. Este método usa a variável retornada em `setTimeout()`. Se a função não tiver sido executada, você pode interromper a execução chamando o método `clearTimeout()`.

```
1 <body>
2   <button onclick="myVar = setTimeout(myfunction,3000)">Clique Aqui</
button>
3   <button onclick="clearTimeout(myVar)">Cancelar</button>
4   <script type="text/javascript">
5       function myfunction(){
6           alert("Alerta!!")
7       }
8   </script>
9 </body>
```

O método de `setInterval()` – repete uma determinada função a cada intervalo de tempo:

```
1 <body>
2   <button onclick="setInterval(myfunction,3000)">Clique Aqui</button>
3   <script type="text/javascript">
4       function myfunction(){
5           alert("Alerta!!")
6       }
7   </script>
8 </body>
```

## 3 JQuery

O JQuery é uma biblioteca JavaScript leve com objetivo de facilitar o uso do JavaScript no seu site. Pega tarefas comuns que exigem muitas linhas de código JavaScript para ser executadas e as agrupa em métodos que você pode chamar com uma única linha de código. Para usar o jQuery em seu site, você pode: fazer o download da biblioteca jQuery em [jQuery.com](https://jquery.com/), ou incluir jQuery de uma CDN(Content Delivery Network), como o Google.

Os seletores jQuery permitem selecionar e manipular elementos HTML. São usados para “encontrar” (ou selecionar) elementos HTML com base em seu nome, ID, classes, tipos, atributos, valores de atributos e muito mais. Ele é baseado nos seletores CSS existentes e, além disso, possui alguns seletores personalizados.

A biblioteca jQuery é um único arquivo JavaScript e você a referencia com a tag HTML <script>, deve estar dentro do <head>.

- Vamos usar o jQuery do Google:

```
1 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/
  jquery.min.js"></script>
```

Todos devem começar com o cifrão e parênteses: \$(). Você pode selecionar todos os elementos <p> de uma página : \$("p"). Exemplos: quando o usuário clica em um botão, todos os elementos <p> ficam ocultos:

```
1 <head>
2   <title>Aula 7</title>
3   <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="s
  ha256-/xUj+30JU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous
  "></script> <script type="text/javascript">
4       $(document).ready(function(){
5           $("button").click(function(){
6               $("p").hide(1000);
7           });
8       });
9   </script>
10 </head>
11 <body>
12   <button><p>Se clicar aqui, irei desaparecer</p></button>
13   <p>Clique aqui também</p>
14   <span>Eu não irei desaparecer</span>
15 </body>
```

Você pode utilizar todos os elementos com o mesmo id ou a mesma classe também (o id deve estar em apenas um elemento).

```
1 <head>
2   <title>Aula 7</title>
3   <script src="https://code.jquery.com/jquery-3.6.0.min.js"
  integrity="sha256-/xUj+30JU5yExlq6GSYGSHk7tPXikynS7ogEvD
4
5       $(document).ready(function(){
6           $("p").click(function(){
7               $("#teste").hide(1000);
8           });
9       });
```

```

10
11
12     $(document).ready(function(){
13         $("span").click(function(){
14             $(".teste").hide(5000);
15         });
16     });
17 </script>
18 </head>
19 <body>
20     <p id="teste">Se clicar aqui, irei desaparecer</p>
21     <p>Clique aqui também</p>
22     <span class="teste">Eu não irei desaparecer</span>
23 </body>

```

Os alertas do bootstrap servem para fornecem mensagens de feedback contextual para ações típicas do usuário com um punhado de mensagens de alerta disponíveis e flexíveis. Os alertas estão disponíveis para qualquer tamanho de texto, nas oito classes contextuais do bootstrap. Como:

```

1 <div class="alert alert-primary" role="alert">
2   This is a primary alert-check it out!
3 </div>
4 <div class="alert alert-secondary" role="alert">
5   This is a secondary alert-check it out!
6 </div>
7 <div class="alert alert-success" role="alert">
8   This is a success alert-check it out!
9 </div>
10 <div class="alert alert-danger" role="alert">
11   This is a danger alert-check it out!
12 </div>

```

Ao clicar em um alerta a mensagem “alerta bloqueados” irá aparecer:

```

1 <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha2
56-/xUj+30JU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"><
/script> <script type="text/javascript">
2     $(document).ready(function(){
3         $("span").click(function(){
4             $(".teste").html("ALERTAS BLOQUEADOS");
5         });
6     });
7 </script>
8 <body>
9     <span class="teste alert alert-primary" role="alert">Alerta</span>
10 </body>

```

Podemos também mudar a cor do alerta:

```

1 <body>
2     <center style="padding: 50px;">
3         <button type="button" id="vermelho" class="btn btn-outline-
danger btn-lg">Vermelho</button>
4         <br><br>
5         <div class="um alert alert-dark" role="alert">Alerta</div>
6

```

```

7         </center>
8
9         <script type="text/javascript">
10             var v=document.getElementById('vermelho');
11
12             v.onclick=danger;
13
14             function danger(){
15                 $('.um').removeClass('alert-primary alert-danger alert-success
16                 alert-dark').addClass('alert-danger');
17             };
18         </script>
19     </body>

```

Agora vamos criar um formulário dinâmico com protocolos no jquery:

```

1     <body>
2         <form id="form">
3             <div class="form" class="contact-form">
4                 <div class="col-md-4">
5                     <label><h5>O colaborador conhece o caminho do refeitório?</h5></
6                     label>
7                 </div>
8                 <div class="col-md-4">
9                     <label class="radio-inline">Sim
10                     <input type="radio" class="form-control" name="ControleProtol"
11                     value="Sim" id="ProtocolSim">
12                     </label>
13                 </div>
14                 <div class="col-md-4">
15                     <label class="radio-inline" for="radios-1">Não
16                     <input type="radio" class="form-control" name="ControleProtol"
17                     value="Nao" id="ProtocolNao">
18                     </label>
19                 </div>
20                 <div class="col-md-4">
21                     <label class="radio-inline" for="radios-1">
22                     <input type="radio" class="form-control" name="ControleProtol"
23                     value="NULL" checked hidden>
24                     </label>
25                 </div>
26                 <div id="NaoProtocol" style="display: none;">
27                     <input type="text" name="texto" id="NaoProtocol">
28                 </div>
29                 <div>
30                     <button id="submitform">Enviar</button>
31                 </div>
32             </form>
33         </body>

```

Para executar usamos o script:

```

1     <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-
2     6-/xUj+30JU5yExlq6GSGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></
3     script> <script type="text/javascript">
4         var ProtocolNao = document.getElementById('ProtocolNao');
5         var ProtocolSim = document.getElementById('ProtocolSim');
6         var Submit = document.getElementById('submitform');

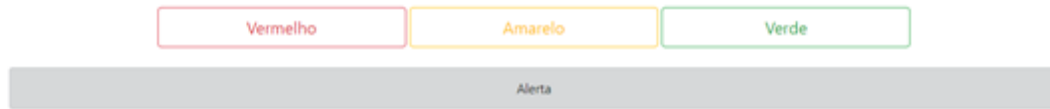
```



```
5      Submit.onclick = submitform;
6      ProtocolNao.onclick = elemento1;
7      ProtocolSim.onclick = elemento2;
8
9      function submitform(){
10         document.getElementById('form').submit();
11     }
12     function elemento1(){
13         document.getElementById('NaoProtocol').style.display="block";
14     }
15     function elemento2(){
16         document.getElementById('NaoProtocol').style.display="none";
17     }
18 }
19 </script>
```

## 4 Desafio:

Crie três um alerta e três opções de cores com botões, conforme os botões forem clicados a cor do alerta irá mudar, assim como a mensagem, use vermelho para “perigo”, amarelo para “aviso” e verde para “sucesso”. Como no exemplo:



Código:

```

1  <body>
2
3      <center style="padding: 50px;">
4          <button type="button" id="vermelho" class="btn btn-outline-
5          danger btn-lg">Vermelho</button>
6          <button type="button" id="azul" class="btn btn-outline-warning
7          btn-lg">Amarelo</button>
8          <button type="button" id="verde" class="btn btn-outline-
9          success btn-lg">Verde</button>
10         <br><br>
11         <div class="um alert alert-dark" id="alerta" role="alert">Alert
12         a</div>
13
14     </center>
15     <script type="text/javascript">
16
17         $("#vermelho").click(function(){
18             $('.um').removeClass('alert-primary alert-danger alert-success
19             alert-dark').addClass('alert-danger');
20             $('#alerta').html('Danger');
21         });
22
23         $("#verde").click(function(){
24             $('.um').removeClass('alert-primary alert-danger alert-success
25             alert-dark alert-warning').addClass('alert-success');
26             $('#alerta').html('Success');
27         });
28
29         $("#azul").click(function(){
30             $('.um').removeClass('alert-primary alert-danger alert-success
31             alert-dark').addClass('alert-warning');
32             $('#alerta').html('warning');
33         });
34     </script>
35 </body>

```