

Instituto Federal Goiano - Campus Ceres  
Bacharelado em Sistemas de Informação  
Prof. Me. Ronneesley Moura Teles

Davi Ildeu de Faria  
Marcos Antonio Arriel Rodrigues  
Paulo Henrique Rodrigues Araujo  
Salmi Nunes de Paula Junior  
Thalia Santos de Santana  
Warley Rodrigues de Andrade  
Wesley Moraes Felix  
Willian Wallace de Matteus Silva

## ***jQuery***

### *Principais Funcionalidades*

Outubro  
2017

# Sumário

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>O que é?</b>   | <b>2</b> |
| <b>2</b> | <b>Como incluir</b>   | <b>2</b> |
| <b>3</b> | <b>Função do \$</b>   | <b>2</b> |
| <b>4</b> | <b>Alteração de CSS</b>                                     | <b>3</b> |
| <b>5</b> | <b>Função HTML()</b>  | <b>5</b> |
| <b>6</b> | <b>Função append(), appendTo(), prepend() e prependTo()</b> | <b>6</b> |
| <b>7</b> | <b>Referências</b>  | <b>9</b> |

# O jQuery

## 1 O que é?

O jQuery trata-se de uma biblioteca de JavaScript com uma série de diversas funções, capazes de ajudar desenvolvedores. Considerada como uma biblioteca popular e padrão para programação *front-end* (refere-se a interfaces de *websites*), é mantida pela Fundação jQuery. De acordo com o site oficial é “[...] rápida, pequena e rica em recursos”. Ademais, acaba trabalhando com a manipulação de eventos, AJAX e HTML, proporcionando uma maior integração, compatibilidade e funcionamento nos mais variados navegadores. No caso de páginas *web*, auxilia principalmente para o uso de JavaScript (JS) com interações HTML.

A biblioteca foi criada em 2006, por John Resig, diante das frustrações decorrentes de códigos JS incompatíveis em navegadores diferentes. Para sua concepção, pensou-se em um agregado de funções com o intuito de resolver problemas mais complexos de serem resolvidos puramente com JavaScript.

## 2 Como incluir

Inicialmente, deve-se realizar o *download* da versão mais recente (3.2.1) da biblioteca jQuery no site oficial (<http://jquery.com>). Depois, é necessária sua referência no cabeçalho da página HTML, conforme código abaixo. Ressalta-se que caminho do arquivo e o nome do *.js* deve ser o mesmo usado ao baixar a biblioteca. Também pode-se incluir por meio de referência externa.

```
1 <!-- Referência local -->
2 <script language="javascript" type="text/javascript" src="jquery.js">
3 </script>
4
5 <!-- Referência externa -->
6 <script src="https://code.jquery.com/jquery-3.2.1.js" type="text/
  javascript"></script>
```

codigos/incluir.html

## 3 Função do \$

A função `$()` permite selecionar elementos com maior facilidade, maior compatibilidade, com menos código e de maneira intuitiva.

- Código em JavaScript “Puro”

```
1 var cabecalho = document.getElementById("cabecalho");
2
3 if (cabecalho.attachEvent) {
4     cabecalho.attachEvent("onclick", function (event) {
5         alert("Você clicou no cabeçalho, usuário do IE!");
6     });
7 } else if (cabecalho.addEventListener) {
```

```

8 | cabecalho.addEventListener("click", function (event) {
9 |     alert("Você clicou no cabeçalho!");
10 | }, false);
11 | }

```

codigos/cabecalho.js.js

O código acima adiciona um evento onde o id é “cabeçalho”.

- Código com jQuery

```

1 | $("#cabecalho").click(function (event) {
2 |     alert("Você clicou no cabeçalho!");
3 | });

```

codigos/cabecalho-jquery.js

A sintaxe é bem menor, e a biblioteca se encarrega de encontrar o modo mais compatível para adicionar o evento ao elemento de id “cabeçalho”.

## 4 Alteração de CSS

A biblioteca jQuery fornece algumas funções para interagir com o estilo (CSS) dos elementos, permitindo obter e modificar os valores de vários atributos especificados. Abaixo, há exemplos de estruturas HTML e CSS, para aplicação das funções:

```

1 | <div id="divExterna">
2 |     <div id="divInterna1">
3 |         Div Interna 1
4 |     </div>
5 |     <div id="divInterna2">
6 |         Div Interna 2
7 |     </div>
8 |     <div id="divInterna3">
9 |         Div Interna 3
10 |    </div>
11 |    <div id="divInterna3">
12 |        Div Interna 4
13 |    </div>
14 |    <div id="divInterna4">
15 |        Div Interna 5
16 |    </div>
17 | </div>

```

codigos/divHtml.html

```

1 | div{border:solid 1px;}
2 | #divExterna{
3 |     width:510px;
4 |     height:100px;
5 |     padding:10px;
6 | }
7 | #divExterna div{
8 |     float:left;
9 |     width:100px;

```

```

10         height:100px;
11     }
12     .divVermelha{
13         background:red;
14     }
15     divAzul{
16         border-color:blue;
17         border-width:5px
18     }

```

codigos/divCss.html

Neste, faz-se uso de funções de *css()*, *addClass()*, *removeClass()*, *toggleClass()* e *hasClass()*.

- *css()*: Esta função permite obter ou definir um atributo CSS do(s) elemento(s) selecionados.

```

1 //Obtenção de valor
2 var divFloat = $("#divExterna div").css("float");
3 alert(divFloat);
4 //Definição de valor
5 $("#divInterna3").css("border-color", "yellow")

```

codigos/css.html

- *addClass()*: Utilizada para adicionar uma classe CSS pré-definida a um ou a vários elementos. Na folha de estilos exemplificada, já temos duas classes prontas, “divVermelha” e “divAzul”, as quais serão de uso como argumento desta função para alterar a aparência de duas das divs internas.

```

1 $("#divInterna1").addClass("divVermelha");
2 $("#divInterna4").addClass("divAzul");

```

codigos/addClass.html

- *removeClass()*: Busca remover uma classe CSS de certo elemento.

```

1 $("#divInterna1").addClass("divVermelha");
2 $("#divInterna4").addClass("divAzul");
3 $("#divInterna1").removeClass("divVermelha");

```

codigos/removeClass.html

- *toggleClass()*: Adiciona ou remove uma classe a um elemento, dependendo de sua existência ou não. Se o elemento já a tiver, será removida e se não, será adicionada.

```

1 $("#divInterna2").toggleClass("divVermelha");
2 $("#divInterna1").toggleClass("divAzul", true);

```

codigos/toggleClass.html

- *hasClass()*: A função retorna *true*, caso o elemento selecionado tenha a classe passada como parâmetro, ou *false* em negativo.

```

1 var temClasse = $("#divInterna2").hasClass("divAzul");
2 if(temClasse)
3     alert("A div 2 tem a classe divAzul");

```

```

4 | else
5 |     alert("A div 2 NAO tem a classe divAzul");

```

codigos/hasClass.html

## 5 Função HTML()

A função HTML() tem dois usos distintos: Pode-se usá-la para obter o conteúdo em html de um determinado elemento e para definir esse conteúdo. Na primeira forma, basta que o método seja invocado sem a passagem de parâmetros, na segunda deve-se passar o conteúdo total do elemento.

```

1 | <!doctype>
2 | <head>
3 |   <title>html demo</title>
4 |   <style>
5 |     p {
6 |       margin: 8px;
7 |       font-size: 20px;
8 |       color: blue;
9 |       cursor: pointer;
10 |    }
11 |    b {
12 |      text-decoration: underline;
13 |    }
14 |    button {
15 |      cursor: pointer;
16 |    }
17 |   </style>
18 |   <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
19 | </head>
20 | <body>
21 |
22 |
23 | <input id="texto" type="text" placeholder="digite o texto aqui!" /> <
24 |   button>comentar</button>
25 |
26 | <div id='target '>-----</div>
27 |
28 | <script>
29 | $( "button" ).click(function() {
30 |
31 |   var txt = $('#texto').val();
32 |   $('#target').html(txt);
33 |
34 | });
35 | </script>
36 |
37 | </body>
38 | </html>

```

codigos/html.html

## 6 Função `append()`, `appendTo()`, `prepend()` e `prependTo()`

Em alguns casos pode ser necessário adicionar conteúdos dentro do código HTML. Os métodos *`append()`*, *`appendTo()`*, *`prepend()`* e *`prependTo()`* podem ser usados para esse fim, adicionando novos objetos a um elemento existente.

- `append()`: Usa um conteúdo passado como parâmetro inserindo-o no final de um controle alvo, também passado como parâmetro. Cada vez que o botão for clicado o conteúdo será inserido novamente.

```
1 <html>
2 <head>
3   <title>Append</title>
4   <script src="https://code.jquery.com/jquery-3.2.1.js" type="text
5     /javascript"></script>
6
7   <script type="text/javascript">
8     $(document).ready(function() {
9       $("#append").click(function() {
10        $('#conteudo').append("<div class='novo_conteudo'>
11          Inserido com Append</div>");
12      });
13    });
14  </script>
15
16  <style type="text/css">
17    .conteudo { padding:5px; border:1px solid #000; width
18      :200px; height:200px; }
19    .novo_conteudo {border:1px solid #F00; width:198px;
20      height:50px; }
21  </style>
22 </head>
23
24 <body>
25   <div class="conteudo">Conteúdo Normal</div>
26
27   <br/>
28   <input type="button" name="append" id="append" value="
29     Executar Append()">
30 </body>
31 </html>
```

codigos/append.html

- `prepend()`: A diferença do `Append()` é apenas que o conteúdo passada no parâmetro será adicionado no início do controle alvo, antes do conteúdo inicial.

```
1 <html>
2 <head>
3   <title>Prepend</title>
```

```

4 <script src="https://code.jquery.com/jquery-3.2.1.js" type="text
  /javascript"></script>
5
6 <script type="text/javascript">
7     $(document).ready(function() {
8         $("#prepend").click(function() {
9             $('.conteudo').prepend("<div class='novo_conteudo
  '>Inserido com Prepend</div>");
10         });
11     });
12
13 </script>
14
15 <style type="text/css">
16     .conteudo { padding:5px; border:1px solid #000; width
  :200px; height:200px; }
17     .novo_conteudo {border:1px solid #F00; width:198px;
  height:50px; }
18 </style>
19
20 </head>
21
22 <body>
23
24     <div class="conteudo">Conteúdo Normal</div>
25
26 <br/>
27
28     <input type="button" name="prepend" id="prepend" value="
  Executar Prepend()">
29
30 </body>
31 </html>

```

codigos/prepend.html

- appendTo(): Nesse modelo o conteúdo é transferido de um alvo para outro, funcionando como um recorte de conteúdo. A div “novo\_conteudo” é transferida para dentro da div “conteudo” sendo inserida após o conteúdo original da div.

```

1 <html>
2 <head>
3     <title>AppendTo</title>
4     <script src="https://code.jquery.com/jquery-3.2.1.js" type="text
  /javascript"></script>
5
6     <script type="text/javascript">
7         $(document).ready(function() {
8             $("#appendTo").click(function() {
9                 $(".novo_conteudo").appendTo(".conteudo");
10             });
11         });
12
13 </script>
14
15 <style type="text/css">
16     .conteudo { padding:5px; border:1px solid #000; width
  :200px; height:200px; }

```



```

17         .novo_conteudo {border:1px solid #F00; width:198px;
18         height:50px; }
19         </style>
20
21 </head>
22
23 <body>
24
25     <div class="novo_conteudo">Conteúdo AppendTo</div>
26
27     <div class="conteudo">Conteúdo Normal</div>
28
29 <br/>
30
31     <input type="button" name="appendTo" id="appendTo" value="
32     Executar appendTo()">
33
34 </body>
</html>

```

codigos/appendTo.html

- prependTo(): No caso do PrependTo() acontece da mesma forma do AppendTo(), porém a div “novo\_conteudo” é inserida antes do conteúdo original da div alvo.

```

1 <html>
2 <head>
3     <title>PrependTo</title>
4     <script src="https://code.jquery.com/jquery-3.2.1.js" type="text
5     /javascript"></script>
6
7     <script type="text/javascript">
8         $(document).ready(function() {
9             $("#prependto").click(function() {
10                 $(".novo_conteudo").prependTo(".conteudo");
11             });
12         });
13     </script>
14
15     <style type="text/css">
16         .conteudo { padding:5px; border:1px solid #000; width
17         :200px; height:200px; }
18         .novo_conteudo {border:1px solid #F00; width:198px;
19         height:50px; }
20     </style>
21
22 </head>
23
24 <body>
25
26     <div class="novo_conteudo">Conteúdo PrependTo</div>
27
28     <div class="conteudo">Conteúdo Normal</div>
29
30 <br/>

```

```

30 |
31 |         <input type="button" name="prependto" id="prependto" value=
32 |         "Executar prependTo()">
33 |     </body>
34 | </html>

```

codigos/prependTo.html

## 7 Referências

1. KHAN ACADEMY. **Pausa para a história: como John criou o jQuery?** Disponível em: <<http://https://pt.khanacademy.org/computing/computer-programming/html-js-jquery/jquery-dom-access/a/history-of-jquery>>. Acesso em: 22 de setembro de 2017.
2. JQUERY. **jQuery, write less, do more.** Disponível em: <<http://jquery.com>>. Acesso em: 21 de setembro de 2017.
3. CAELUM. **Apostila Desenvolvimento Web com HTML, CSS e JavaScript.** Disponível em: <<https://www.caelum.com.br/apostila-html-css-javascript/jquery/>>. Acesso em: 22 de setembro de 2017.
4. W3SCHOOLS. **jQuery html() Method.** Disponível em: <[https://www.w3schools.com/jquery/html\\_html.asp](https://www.w3schools.com/jquery/html_html.asp)>. Acesso em: 28 de setembro de 2017.
5. LINHA DE CÓDIGO. **Trabalhando com inserção de conteúdo com jQuery.** Disponível em: <<http://www.linhadecodigo.com.br/artigo/3499/trabalhando-com-insercao-de-conteudo-com-jquery.aspx>>. Acesso em: 03 de outubro de 2017.
6. SPIGANDO E APRENDENDO. **jQuery ? Os métodos append() e prepend().** Disponível em: <<https://spigandoeaprendendo.wordpress.com/2014/03/12/jquery-os-metodos-append-e-prepend/>>. Acesso em: 03 de outubro de 2017.
7. FÓRUM IMASTERS. **Entendendo prepend(), prependTo(), appeend() e appeendTo() do JQuery.** Disponível em: <<https://forum.imasters.com.br/topic/409747-entendendo-prepend-prependto-appeend-e-appeendto-do-jquery/>>. Acesso em: 03 de outubro de 2017.
8. LINHA DE CÓDIGO. **Interação com classes e atributos CSS no jQuery.** Disponível em: <<http://www.linhadecodigo.com.br/artigo/3475/interacao-com-classes-e-atributos-css-no-jquery.aspx>>. Acesso em: 06 de outubro de 2017.