

# Bootstrap - JS - Plugins: Tab, Tolltip, Popover; Alert, Button

APRESENTAR TÉCNICAS PARA UTILIZAÇÃO DOS PLUGINS TAB, TOOLTIP, POPOVER, ALERT E BUTTON COM JAVASCRIPT.

AUTOR(A): PROF. JORGE HENRIQUE PESSOTA

## Tab

Tab é um componente Bootstrap que gera um painel com abas para navegação. Para utilizar este componente, deve-se criar uma lista não ordenada contendo as classes Bootstrap .nav e .nav-tabs no elemento <ul>.

Exemplo:

```
1. <ul class="nav nav-tabs">
2.   <li class="active"><a href="#">Home</a></li>
3.   <li><a href="#">Menu 1</a></li>
4.   <li><a href="#">Menu 2</a></li>
5.   <li><a href="#">Menu 3</a></li>
6. </ul>
```

Tab com JavaScript

O componente Tab pode ser acionado com JavaScript das seguintes maneiras:

```
1. // Acionamento da aba clicada pelo usuario
2. $('nav-tabs a').click(function(){
3.     $(this).tab('show');
4. })
5.
6. // Acionamento pelo nome da aba
7. $('nav-tabs a[href="#menu"]').tab('show');
8.
9. // Acionamento da primeira aba
10. $('nav-tabs a:first').tab('show');
11.
12. // Acionamento da última aba
13. $('nav-tabs a:last').tab('show');
14.
15. // Acionamento da uma aba pelo índice (o índice inicia em 0)
16. $('nav-tabs li:eq(2) a').tab('show');
```

#### Opções - Plugin Tab

Não há opções para o *plugin* Tab.

#### Métodos - Plugin Tab

Os seguintes métodos podem ser invocados com JavaScript:

Método	Descrição
.tab("show")	Aciona a aba

Exemplo:

```
1. $('nav-tabs a').click(function(){
2.     $(this).tab('show');
3. });
```

## Eventos - Plugin Tab

Os seguintes eventos podem ser disparados com JavaScript:

Evento	Descrição
show.bs.tab	O evento é disparado antes do acionamento da aba.
shown.bs.tab	O evento é disparado após o acionamento da aba.
hide.bs.tab	O evento é disparado antes de a aba ser ocultada.
hidden.bs.tab	O evento é disparado após a aba ser ocultada.

Exemplo:

```
1. $( '.nav-tabs a' ).on('show.bs.modal', function(){  
2.    // Instruções;  
3. });
```

Abaixo é apresentado um código completo que utiliza JavaScript para acionar Tab.

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width, initial-scale=1">
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.    <script src="js/jquery-3.2.1.min.js"></script>
11.    <script src="js/bootstrap.min.js"></script>
12.
13.  </head>
14.
15.  <body>
16.
17.    <div class="container">
18.
19.      <h2>Bootstrap - Tabs com JavaScript</h2> <br>
20.
21.      <ul class="nav nav-tabs">
22.        <li class="active"><a href="#home">Home</a></li>
23.        <li><a href="#html">HTML</a></li>
24.        <li><a href="#css">CSS</a></li>
25.        <li><a href="#javascript">JavaScript</a></li>
26.        <li><a href="#bootstrap">Bootstrap</a></li>
27.      </ul>
28.
29.      <div class="tab-content">
30.        <div id="home" class="tab-pane fade in active">
31.          <h2>Home</h2>
32.          <p>lorem ipsum pulvinar ornare ac venenatis elementum, egestas
33.            potenti curabitur praesent erat, sollicitudin quisque mollis
34.            egestas est. potenti porta cubilia ipsum adipiscing curae nec
35.            quis sem imperdiet leo dictumst litora mollis amet, feugiat
36.          </p>
37.        </div>
38.        <div id="html" class="tab-pane fade">
39.          <h2>HTML</h2>
40.          <p>lorem ipsum pulvinar ornare ac venenatis elementum, egestas
```

```
41.         potenti curabitur praesent erat, sollicitudin quisque mollis
42.         egestas est. potenti porta cubilia ipsum adipiscing curae ne
43.         quis sem imperdiet leo dictumst litora mollis amet, feugiat
44.     </p>
45. </div>
46. <div id="css" class="tab-pane fade">
47.     <h2>CSS</h2>
48.     <p>lorem ipsum pulvinar ornare ac venenatis elementum, egestas
49.     potenti curabitur praesent erat, sollicitudin quisque mollis
50.     egestas est. potenti porta cubilia ipsum adipiscing curae ne
51.     quis sem imperdiet leo dictumst litora mollis amet, feugiat
52.     </p>
53. </div>
54. <div id="javascript" class="tab-pane fade">
55.     <h2>JavaScript</h2>
56.     <p>lorem ipsum pulvinar ornare ac venenatis elementum, egestas
57.     potenti curabitur praesent erat, sollicitudin quisque mollis
58.     egestas est. potenti porta cubilia ipsum adipiscing curae ne
59.     quis sem imperdiet leo dictumst litora mollis amet, feugiat
60.     </p>
61. </div>
62. <div id="bootstrap" class="tab-pane fade">
63.     <h2>Bootstrap</h2>
64.     <p>lorem ipsum pulvinar ornare ac venenatis elementum, egestas
65.     potenti curabitur praesent erat, sollicitudin quisque mollis
66.     egestas est. potenti porta cubilia ipsum adipiscing curae ne
67.     quis sem imperdiet leo dictumst litora mollis amet, feugiat
68.     </p>
69. </div>
70.
71. </div>
72.
73. </div>
74.
75. <script>
76.     $(document).ready(function () {
77.         $(".nav-tabs a").click(function () {
78.             $(this).tab('show');
79.         });
80.         $('.nav-tabs a').on('show.bs.tab', function () {
81.             alert('Evento disparado antes do acionamento da aba.');
```

```
82.         });
83.         $('.nav-tabs a').on('shown.bs.tab', function () {
84.             alert('Evento disparado após o acionamento da aba.');
```

```
85.         });
86.         $('.nav-tabs a').on('hide.bs.tab', function (e) {
87.             alert('Evento disparado antes de a aba ser ocultada.');
```

```
88.         });
89.         $('.nav-tabs a').on('hidden.bs.tab', function () {
90.             alert('Evento disparado após de a aba ser ocultada.');
```

```
91.         });
92.     });
93. </script>
94.
95. </body>
96. </html>
```

## Tolltip

O *plugin* Tolltip consiste em uma pequena caixa *popup* que é apresentada quando o cursor é posicionado sobre um elemento.

Para utilizá-lo, é necessário adicionar o atributo `data-toggle="tooltip"` e o atributo `title` no elemento desejado. No atributo `title` é especificado o texto que deve ser apresentado no *popup*.

Para implementar este *plugin*, é necessário o uso de jQuery (JavaScript).

Exemplo:

```

1. <!-- HTML -->
2. <a href="#" data-toggle="tooltip" title="Este é um elemento <a>">Tolltip <
3.
4.
5. <script>
6.     // jQuery
7.     $(document).ready(function(){
8.         $('[data-toggle="tooltip"]').tooltip();
9.     });
10. </script>

```

Resultado:



file:///C:/Users/Jorge/Desktop/Conteudista\_AVA/bootstrap/bootstrap-3.3.7-dist/bootstrap-3.3.7-dist/Bootstrap\_JavaScript\_Tolltip\_Exemplo.html#

Tolltip com JavaScript

Para adicionar um Tolltip com JavaScript, os seguintes métodos podem ser utilizados:

```
1. // Tooltip é aplicado em todos os elementos que possuam
2.   o atributo data-toggle="tooltip"
3. $('[data-toggle="tooltip"]').tooltip();
4.
5. // Tooltip é aplicado para um elemento específico
6. $('#id-Elemento').tooltip();
```

#### Opções - Plugin Tooltip

As seguintes opções estão disponíveis para o *plugin* Tooltip:

Opção	Tipo	Valor Default	Descrição
animation	booleano	true	Esta opção permite adicionar o efeito de transição CSS no tooltip. O valor default "true" adiciona o efeito e o valor "false" inibe.
container	string ou o valor booleano "false"	false	Adiciona o tooltip em um elemento específico. Exemplo: <i>container</i> : 'body'
delay	number	0	Esta opção permite especificar o tempo em milisegundos para que o tooltip seja apresentado e o tempo para que ele seja ocultado. Exemplo: delay: {show: 100, hide: 50}



html	booleano	false	<p>Esta opção permite especificar se tags HTML são renderizadas no tooltip. O valor "true" permite e o valor "false" não permite.</p> <p>As tags HTML devem ser inseridas no atributo "title" ou então na opção do plugin.</p>
placement	string	"top"	<p>Especifica a posição em que o tooltip deve aparecer. Os seguintes valores são permitidos: "top", "bottom", "left", "right" e "auto" (o navegador decide onde posicionar o tooltip)</p>
selector	string ou o valor booleano "false"	false	<p>Esta opção adiciona o tooltip em um seletor em específico.</p>
title	string	""	<p>Esta opção é utilizada para definir o texto que será apresentado no tooltip.</p>
trigger	string	"hover focus"	<p>Esta opção permite definir como o tooltip será disparado. Os valores possíveis são click  hover   focus   manual.</p> <p>É possível definir mais de uma trigger, separadas por espaço.</p>

Exemplo:

```
1. <script>
2.     $(document).ready(function(){
3.         $('.btn-success').tooltip({trigger: "click"});
4.         $('.btn-warning').tooltip({trigger: "hover"});
5.         $('.btn-danger').tooltip({trigger: "focus", delay: {show: 0, hide: 100}});
6.     });
7. </script>
```

#### Métodos - Plugin Tooltip

Os seguintes métodos estão disponíveis para o *plugin* Tooltip:

Método	Descrição
.tooltip(options)	Ativa o tooltip com a opção informada.
.tooltip("show")	Apresenta o tooltip.
.tooltip("hide")	Oculto o tooltip.
.tooltip("toggle")	Alterna o tooltip.
.tooltip("destroy")	Oculto e destrói o tooltip.

Exemplo:

```
1. <script>
2.   $(document).ready(function(){
3.       $(".btn-default").click(function(){
4.           $('[data-toggle='tooltip']").tooltip('show');
5.       });
6.   });
7. </script>
```

## Eventos - Plugin Tooltip

Os seguintes eventos estão disponíveis para o *plugin* Tooltip:

Evento	Descrição
show.bs.tooltip	O evento é disparado antes da apresentação do tooltip.
shown.bs.tooltip	O evento é disparado após a completa apresentação do tooltip.
hide.bs.tooltip	O evento é disparado antes do tooltip ser ocultado.
hidden.bs.tooltip	O evento é disparado após o tooltip ser ocultado.

Exemplo:

```
1. document).ready(function(){
2.   $( "[data-toggle='tooltip']").on('show.bs.tooltip', function(){
3.       // instrução;
4.   });
```

Abaixo é apresentado um código completo que utiliza JavaScript para adicionar um Tooltip:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width, initial-scale=1">
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.    <script src="js/jquery-3.2.1.min.js"></script>
11.    <script src="js/bootstrap.min.js"></script>
12.
13.  </head>
14.
15.  <body>
16.
17.    <div class="container">
18.
19.      <h2>Bootstrap - Tooltip com JavaScript</h2> <br>
20.
21.      <button type="button" class="btn btn-success"
22.        data-toggle="tooltip" title="Cabeçalho tooltip"
23.        data-content="Este é um elemento <a>">
24.        Success</button>
25.    </button>
26.    <button type="button" class="btn btn-warning"
27.      data-toggle="tooltip" title="Cabeçalho tooltip"
28.      data-content="Este é um elemento <a>">
29.      Warning</button>
30.    </button>
31.    <button type="button" class="btn btn-danger"
32.      data-toggle="tooltip" title="Cabeçalho tooltip"
33.      data-content="Este é um elemento <a>">
34.      Danger</button>
35.    </button>
36.
37.  </div>
38.
39.  <script>
40.    $(document).ready(function(){
```

```

41.     $('.btn-success').tooltip({trigger: "click"});
42.     $('.btn-warning').tooltip({trigger: "hover"});
43.     $('.btn-danger').tooltip({trigger: "focus", delay: {show: 0, hide:
44.     });
45.     </script>
46.
47. </body>
48. </html>

```

## Popover

O *plugin* Popover é semelhante ao *plugin* Tooltip. O *plugin* Popover exibe uma pequena caixa *popup* quando o usuário clica sob um elemento. Esta caixa *popup* suporta mais conteúdo do que o *popup* exibido pelo Tooltip.

Para utilizá-lo, é necessário adicionar os atributos `data-toggle="popover"`, `title` e `data-content` no elemento desejado. No atributo `title` é especificado o texto que deve ser apresentado no cabeçalho do *popup* e o atributo `data-content` é utilizado para especificar o conteúdo do corpo do *popup*.

Para implementar este *plugin*, é necessário o uso de jQuery.

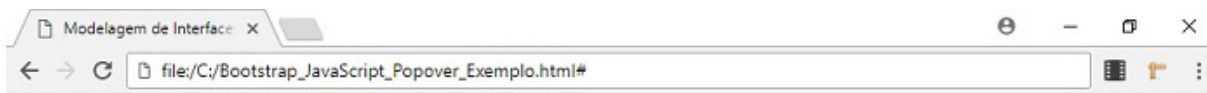
Exemplo:

```

1. <!-- HTML -->
2. <a href="#" data-toggle="popover" title="Cabeçalho popover"
3.     data-content="Este é um elemento <a>">
4.     Exemplo popover
5. </a>
6.
7. <script>
8.     // jQuery
9.     $(document).ready(function(){
10.         $('[data-toggle="popover"]').popover();
11.     });
12. </script>

```

Resultado:



# Bootstrap - Popover com JavaScript

Exemplo popover

Cabeçalho popover

Este é um elemento <a>

## Popover com JavaScript

Para adicionar um Popover com JavaScript, os seguintes métodos podem ser utilizados:

1. // Popover é aplicado em todos os elementos que possuam
2. o atributo data-toggle="popover"
3. `$('[data-toggle="popover"]').popover();`
- 4.
5. // Popover é aplicado para um elemento específico
6. `$('#id-Elemento').popover();`

## Opções - Plugin Popover

As seguintes opções estão disponíveis para o *plugin* Popover:

Opção	Tipo	Valor Default	Descrição
-------	------	---------------	-----------

animation	booleano	true	<p>Esta opção permite adicionar o efeito de transição CSS no popover.</p> <p>O valor default "true" adiciona o efeito e o valor "false" inibe.</p>
container	string ou o valor booleano "false"	false	<p>Adiciona o popover em um elemento específico.</p> <p>Exemplo: <i>container: 'body'</i></p>
content	string	“”	<p>Esta opção é utilizada para definir o texto que será apresentado no corpo do popover.</p>
delay	number	0	<p>Esta opção permite especificar o tempo em milisegundos para que o popover seja apresentado e o tempo para que ele seja ocultado.</p> <p>Exemplo: delay: {show: 100, hide: 50}</p>
html	booleano	false	<p>Esta opção permite especificar se tags HTML são renderizadas no popover. O valor "true" permite e o valor "false" não permite.</p> <p>As tags HTML devem ser inseridas no atributo "title" ou então na opção do <i>plugin</i>.</p>

placement	string	"right"	<p>Especifica a posição em que o popover deve aparecer. Os seguintes valores são permitidos: "top", "bottom", "left", "right" e "auto" (o navegador decide onde posicionar o popover)</p>
selector	string ou o valor booleano "false"	false	<p>Esta opção adiciona o popover em um seletor em específico.</p>
title	string	""	<p>Esta opção é utilizada para definir o texto que será apresentado no cabeçalho do popover.</p>
trigger	string	"click"	<p>Esta opção permite definir como o popover será disparado. Os valores possíveis são click  hover   focus   manual.</p> <p>É possível definir mais de uma <i>trigger</i>, separadas por espaço.</p>

Exemplo:



```

1. <script>
2.     $(document).ready(function(){
3.         $('.btn-success').popover({trigger: "click"});
4.         $('.btn-warning').popover({trigger: "hover"});
5.         $('.btn-danger').popover({trigger: "focus", delay: {show: 0, hide: 100}});
6.     });
7. </script>

```

## Métodos - Plugin Popover

Os seguintes métodos estão disponíveis para o *plugin* Popover:

Método	Descrição
.popover(options)	Ativa o popover com a opção informada.
.popover("show")	Apresenta o popover.
.popover("hide")	Oculto o popover.
.popover("toggle")	Alterna o popover.
.popover("destroy")	Oculto e destrói o popover.

Exemplo:

```

1. <script>
2.     $(document).ready(function(){
3.         $(".btn-default").click(function(){
4.             $('[data-toggle=popover]').popover('show');
5.         });
6.     });
7. </script>

```

## Eventos - Plugin Popover

Os seguintes eventos estão disponíveis para o *plugin* Popover:

Evento	Descrição
show.bs.popover	O evento é disparado antes da apresentação do popover.
shown.bs.popover	O evento é disparado após a completa apresentação do popover.
hide.bs.popover	O evento é disparado antes do popover ser ocultado.
hidden.bs.popover	O evento é disparado após o popover ser ocultado.

Exemplo:

```
1. $(document).ready(function(){
2.   $( "[data-toggle='popover']" ).on('show.bs.popover', function(){
3.     // instrução;
4.   });
```

Abaixo é apresentado um código completo que utiliza JavaScript para adicionar um Popover:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width, initial-scale=1">
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.    <script src="js/jquery-3.2.1.min.js"></script>
11.    <script src="js/bootstrap.min.js"></script>
12.
13.  </head>
14.
15.  <body>
16.
17.    <div class="container">
18.
19.      <h2>Bootstrap - Popover com JavaScript</h2> <br>
20.
21.      <button type="button" class="btn btn-success"
22.        data-toggle="popover" title="Cabeçalho popover"
23.        data-content="Este é um elemento <a>">
24.        Success</button>
25.    </button>
26.    <button type="button" class="btn btn-warning"
27.      data-toggle="popover" title="Cabeçalho popover"
28.      data-content="Este é um elemento <a>">
29.      Warning</button>
30.    </button>
31.    <button type="button" class="btn btn-danger"
32.      data-toggle="popover" title="Cabeçalho popover"
33.      data-content="Este é um elemento <a>">
34.      Danger</button>
35.    </button>
36.
37.  </div>
38.
39.  <script>
40.    $(document).ready(function(){
```

```
41.     $('.btn-success').popover({trigger: "click"});
42.     $('.btn-warning').popover({trigger: "hover"});
43.     $('.btn-danger').popover({trigger: "focus", delay: {show: 0, hide:
44.     });
45.     </script>
46.
47. </body>
48. </html>
```

## Alert

Alerts são recursos que permitem transmitir alguma mensagem contextual para o usuário. Pode ser uma mensagem informativa, a confirmação de sucesso de alguma ação, ou ainda um alerta sobre algo que não deu certo.

Para adicionar o componente alert na página web, deve-se utilizar a classe `.alert`, acompanhada de uma das classes contextuais `.alert-success`, `.alert-info`, `.alert-warning` ou `.alert-danger`.

Exemplo:

```
1. <div class="alert alert-success">
2.     <strong>Sucesso!</strong> Indica uma mensagem de sucesso.
3. </div>
```

### Alert com JavaScript

Para controlar um Alert com JavaScript, o seguinte método pode ser utilizado:

```
1. $(''.close').alert();
```

### Opções - Plugin Alert

Não há opções para o *plugin* Alert.

### Métodos - Plugin Alert

Os seguintes métodos estão disponíveis para o *plugin* Alert:

Método	Descrição
<code>.alert("close")</code>	Fecha a mensagem de alerta.

#### Eventos - Plugin Alert

Os seguintes eventos estão disponíveis para o *plugin* Alert:

Evento	Descrição
<code>close.bs.alert</code>	O evento é disparado antes do fechamento do alerta.
<code>closed.bs.alert</code>	O evento é disparado após o fechamento do alerta.

Exemplo:

```
1. <script>
2.   $(document).ready(function(){
3.     $(".close").click(function(){
4.       $("#id-Alert").alert("close");
5.     });
6.     $("#id-Alert").on('closed.bs.alert', function(){
7.       // instruções;
8.     });
9.   });
10. </script>
```

Abaixo é apresentado um código completo que utiliza JavaScript para manipular um Alert:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width, initial-scale=1">
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.    <script src="js/jquery-3.2.1.min.js"></script>
11.    <script src="js/bootstrap.min.js"></script>
12.
13.  </head>
14.
15.  <body>
16.
17.    <div class="container">
18.
19.      <h2>Bootstrap - Alert com JavaScript</h2> <br>
20.
21.      <div id="id-Alert" class="alert alert-success">
22.        <strong>Sucesso!</strong> Indica uma mensagem de sucesso.
23.        <a href="#" class="close">x</a>
24.      </div>
25.
26.
27.    </div>
28.
29.    <script>
30.      $(document).ready(function(){
31.        $(".close").click(function(){
32.          $("#id-Alert").alert("close");
33.        });
34.
35.        $("#id-Alert").on('closed.bs.alert', function(){
36.          alert("Evento disparado após o fechamento do alerta.");
37.        });
38.      });
39.    </script>
40.
```

```
41.     </body>
42.
43. </html>
```

## Button

Button (botão) é um elemento que pode disparar uma ação ou carregar uma página quando acionado.

Exemplo:

```
1. <button type="button" onclick="alert('Botão acionado!')">Clique aqui!</button>
```

Button com JavaScript

Para controlar um Button com JavaScript, o seguinte método pode ser utilizado:

```
1. $(''.btn').button();
```

Opções - Plugin Button

Não há opções para o *plugin* Button.

Métodos - Plugin Button

Os seguintes métodos estão disponíveis para o *plugin* Button:

Método	Descrição
button("toggle")	Faz que o botão apareça pressionado.
.button("loading")	Desabilita o botão e altera o seu texto para "loading...".

<code>.button("reset")</code>	Altera o texto do botão para o texto padrão, caso tenha sido alterado.
<code>.button("string")</code>	Especifica um novo texto para o botão.

Exemplo:

```
1. <script>
2.   $(document).ready(function(){
3.       $(".btn").click(function(){
4.           $(this).button('loading');
5.       });
6.   });
7. </script>
```

#### Eventos - Plugin Button

Não há eventos para o *plugin* Button.

Abaixo é apresentado um código completo que utiliza JavaScript para manipular um Button:



```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width, initial-scale=1">
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.    <script src="js/jquery-3.2.1.min.js"></script>
11.    <script src="js/bootstrap.min.js"></script>
12.
13.  </head>
14.
15.  <body>
16.
17.    <div class="container">
18.
19.      <h2>Bootstrap - Button com JavaScript</h2> <br>
20.
21.      <button type="button" class="btn btn-primary"
22.        id="idButton">
23.        Clique aqui!
24.      </button>
25.      <br><br>
26.      <button type="button" class="btn btn-default"
27.        id="idReset">
28.        Resetar texto
29.      </button>
30.
31.    </div>
32.
33.    <script>
34.      $(document).ready(function(){
35.        $("#idButton").click(function(){
36.          $(this).button('loading');
37.        });
38.
39.        $("#idReset").click(function(){
40.          $("#idButton").button("reset");
```

```
41.     });  
42.  
43.     });  
44. </script>  
45.  
46. </body>  
47.  
48. </html>
```

## Resumo do Tópico

Neste tópico foram apresentadas técnicas para utilização dos *plugins* Tab, Tooltip, Popover, Alert e Button com JavaScript.

## ATIVIDADE FINAL

Qual é o resultado gerado pela instrução jQuery abaixo?

```
$('.nav-tabs li:eq(2) a').tab('show');
```

- A. A terceira aba de um plugin Tab é acionada.
- B. A segunda aba de um plugin Tab é acionada.
- C. A primeira aba de um plugin Tab é acionada.
- D. A quarta aba de um plugin Tab é acionada.

Qual é o comportamento do *plugin* Popover quando é utilizado o valor *default* da opção *trigger*?

- A. O *plugin* Popover exibe uma mensagem de alerta quando o usuário clica sob um elemento.
- B. O *plugin* Popover exibe uma pequena caixa *popup* quando o usuário posiciona o cursor sob um elemento.
- C. O *plugin* Popover exibe uma pequena caixa *popup* quando o usuário clica sob um elemento.
- D. O *plugin* Popover exibe uma mensagem de alerta quando o usuário posiciona o cursor sob um elemento.

O que ocorre quando o método `.button("loading")` é acionado com JavaScript?

- A. O método alterar a cor e o texto do botão.
- B. O método aguarda um tempo (loading) para acionar o botão.
- C. O método aciona o botão e altera o seu texto para "loading...".
- D. O método desabilita o botão e altera o seu texto para "loading...".

## REFERÊNCIA

SILVA, Maurício Samy. Bootstrap 3.3.5: Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos. 1.<sup>a</sup> ed. São Paulo: Novatec, 2015c.

TUTORIALS POINT. Bootstrap Overview, Bootstrap Tutorial. Disponível em: <http://www.tutorialspoint.com/bootstrap/>

([http://www.tutorialspoint.com/bootstrap/bootstrap\\_tutorial.pdf](http://www.tutorialspoint.com/bootstrap/bootstrap_tutorial.pdf)). Acesso em: 17 de dezembro de 2017.

W3SCHOOLS.COM. Bootstrap Tutorial. Disponível em: <https://www.w3schools.com/bootstrap/default.asp> (<https://www.w3schools.com/bootstrap/default.asp>). Acesso em 20 de dezembro de 2017.

