



# Trabalhando com Eventos

## Revisão

Vimos como o jQuery garante a compatibilidade de nosso código entre diversos navegadores, inclusive como impacta na quantidade de código que escrevemos.

O problema é que nosso código utilizado no exemplo do capítulo anterior era executado apenas uma vez imediatamente após o carregamento da página. Queremos agora que ele seja executado a partir da ação do usuário como um clique de botão ou quando ele deixar um campo. São muitas as possibilidades.

## Hello World

Digite seu nome:

Alura

copiar

Quando pensamos na interatividade do usuário com nossas páginas, pensamos em **eventos** e, com certeza, o jQuery também vai nos ajudar, garantindo a compatibilidade do nosso código.

# Lidando com Eventos



Ainda não nos preocuparemos com a captura do que foi digitado pelo usuário, mas queremos que o nome “Alura” seja exibido apenas quando o usuário clicar no botão.

## Nossa primeira função de eventos: click()

Vamos voltar para a nossa página:

```
1. <!doctype html>
2. <html lang="pt">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>Introdução ao jQuery</title>
6.     <link rel="stylesheet" href="css/bootstrap.css">
7.     <link rel="stylesheet" href="css/estilos.css">
8. </head>
9.     <body class="container">
10.         <h1 class="page-header">Hello World</h1>
11.         <label>Digite seu nome:</label>
12.         <input id="campo-nome"/>
13.         <input id="botao-copiar" type="button" value="copiar"/>
14.
15.         <h2 id="texto-copiado"></h2>
16.
17.         <script src="js/jquery.js"></script>
18.         <script>
19.             //aqui queremos selecionar o botão
20.             $( '#texto-copiado' ).text( 'Alura' );
21.         </script>
22.     </body>
23. </html>
```

A primeira coisa e uma das mais rotineiras com jQuery é selecionar o elemento que queremos manipular, neste caso o botão:

```
1. $( '#botao-copiar' )
```

Agora que já temos acesso ao botão, vamos solicitar ao jQuery através de sua função `click()` que execute nosso código. Não podemos simplesmente movê-lo para dentro dela. A função `click()` assim como muitas outras do jQuery recebe uma **bloco de função** que contém o código que queremos executar. Vamos passar um bloco à função `click()`. Agora, podemos mover nosso código para dentro do bloco de função:

```
1. ...
2. <script src="js/jquery.js"></script>
3. <script>
4.     $( '#botao-copiar' ).click(function() {
5.         $( '#texto-copiado' ).text( 'Alura' );
6.     });
```

7. `</script>`
8. ...

Executando nossa página mais uma vez, a primeira diferença é que o texto “Alura” não aparece assim que a página é carregada, mas apenas quando clicamos no botão.

Agora que entendemos este mecanismo básico, podemos concluir nosso código, fazendo com que o texto digitado pelo usuário seja copiado e exibido na tela.

Vamos criar uma variável chamada ‘nome’ que armazenará o texto digitado pelo usuário.

O problema é que ela ainda não tem valor algum. Vamos pedir ao jQuery acessar o elemento de entrada e que obtenha o valor digitado.

Repare que aqui usamos a função `val()` e não `text()`. A primeira é utilizada por elementos de entrada de dados e o segundo por elementos que exibem informações. Agora que capturamos o retorno da função `val()`, já podemos testar a alteração.

```
1. $('#botao-copiar').click(function() {  
2.     var nome = $('#texto-copiado').val();  
3.     $('#texto-copiado').text(nome);  
4. });
```

Repare que podemos alterar o texto quantas vezes quisermos que, ao clicarmos no botão, o valor presente no campo de entrada será copiado para o elemento `h2` da página.

O jQuery tem funções para outros eventos, como o `blur()`. Disparado toda vez que o usuário deixar um elemento. `mouseover()`, quando o mouse estiver sobre o elemento. Há também o `dblclick()`, quando o elemento for clicado duas vezes em seguida. Há também o `change()`, quando o valor de mudar.

O que precisamos saber é que estas funções são atalhos para a função `on()`. Vamos entender como ela funciona.

Vamos substituir a função `click()` pela função `on()`. Repare que agora, não sabemos a qual evento queremos atribuir nosso código. Para isso, precisamos passar como primeiro parâmetro para a função `on()` o nome do evento, em nosso exemplo `click`. Pronto, se quisermos outro evento, como aquele disparado quando o usuário deixa o campo, usamos “`blur`”. Podemos escolher qualquer evento suportado pelo jQuery. Vamos manter o “`click`”, pois é ele que nos atende neste exemplo.

Abrindo mais uma vez a página, vimos que o comportamento continua o mesmo, a única diferença é que utilizamos a função `on()` no lugar das funções específicas do jQuery que correspondem cada uma a um tipo de evento.





[Termos e condições](#)

[Contato](#)

[Forum](#)

[Sobre](#)

[Sugira um curso](#)

[Quero ser um autor](#)